Bio Sensing Software Platform

Data Transfer Application Example using BLE Rapid Transfer Custom Profile

Introduction

This application note describes a rapid transfer profile (Renesas Rapid Transfer Profile, hereinafter abbreviated as "RRTP") developed based on a custom profile supplied with Bluetooth® low energy protocol stack (Hereinafter abbreviated as "BLE Software") for RL78/G1D.

The sample program includes demo software to check the operation of the rapid transfer profile on Central and Peripheral devices.

In addition, this introduces the rapid data transfer operation demo using RRTP.

In operation demo, it uses the Blood Pressure Monitoring Evaluation Kit (hereinafter abbreviated as "BPMEK")as the Peripheral device (Server), transmits measurement data acquired by Sigma-Delta AD converter on the RL78/H1D via the RL78/G1D on the BPMEK using RRTP.

Using the RL78/G1D Evaluation Board as the Central device (Client) on the opposing device, it converts the received data into text data and transfers the data to PC. You can graph received measurement data to check the waveform.

Target Device

RL78/G1D

Target Board

Bluetooth Central and Peripheral devices:

- RL78/G1D Evaluation Board (RTK0EN0001D01001BZ)
- https://www.renesas.com/us/en/solutions/key-technology/connectivity/bluetooth-smart/evaluation-board.html

Bluetooth Peripheral devices (for Rapid Data Transfer Operation Demo):

- Blood Pressure Monitoring Evaluation Kit (RTK0EH0003S02001BR)
- https://www.renesas.com/us/en/products/software-tools/boards-and-kits/starter-kits/bpm-evaluation-kits-rl78h1d.html

Related documents

- Bluetooth low energy Protocol Stack Application Development Guide (R01AN2768EJ)
- Bluetooth low energy Protocol Stack Sensor Application (R01AN4159EJ)
- Bluetooth low energy Protocol Stack User's Manual (R01UW0095EJ)
- Renesas Flash Programmer V3.05 Flash memory programming software User's Manual (R20UT4307EJ0130)
- Renesas Solution Starter Kit PC GUI Tool Operation Manual for BPMEK (R01AN4396EJ)
- Renesas Solution Starter Kit Blood Pressure Monitoring Evaluation Kit for RL78/H1D User's Manual (R20UT4128EJ)

Contents

1. Overview

The RL78/G1D Evaluation Board is used for BLE devices to be operated.

Prepare two units, one for Central device and one for Peripheral device, and execute the sample program with the configuration shown in 'Figure 1-1 Overview of Sample Program Operating Environment'.

Using two RL78/G1D evaluation boards, one is used as a Peripheral device to create dummy data and data transmission using RRTP, and the other as a Central device to receive data using RRTP.

The sample program of the Peripheral device is automatically put into Advertising state after startup, and starts transmitting data when the connection with the Central device is confirmed. When the sample program of the Central device is started, the registered BD Address is searched from the Peripheral device in the Advertising state, and the Central device is connected to the Peripheral device matching the BD Address. When connection is confirmed, data reception starts.

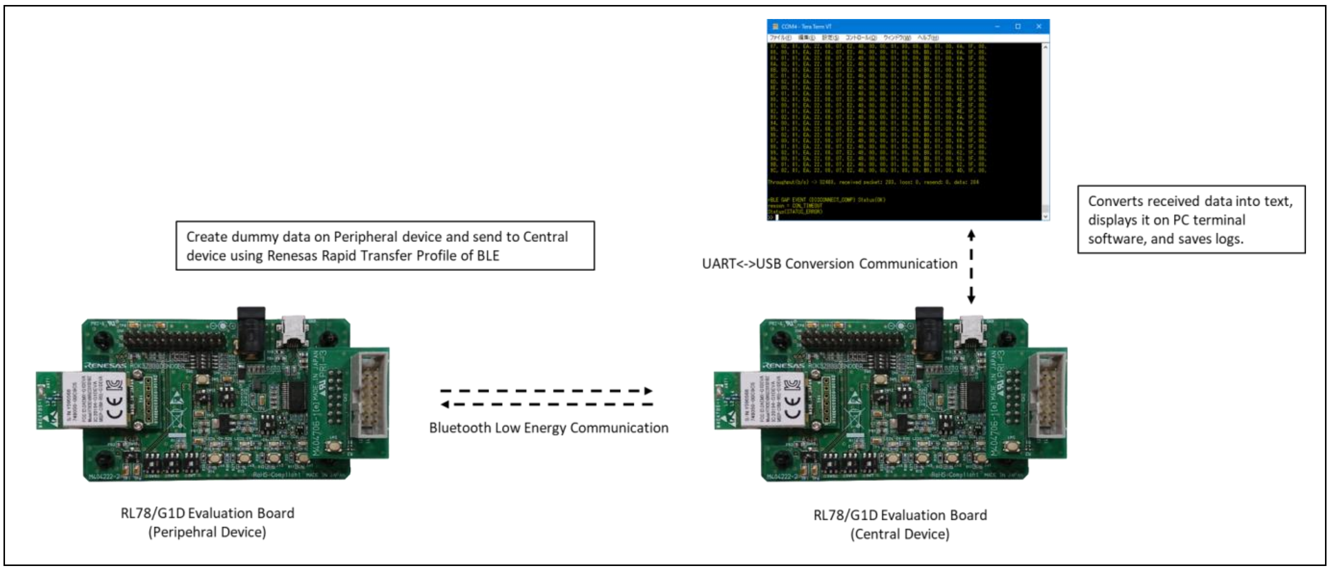

COM4 - Tera Term VT
Converts received data into text, displays it on PC terminal software, and saves logs.
Create dummy data on Peripheral device and send to Central device using Renesas Rapid Transfer Profile of BLE
UART<->USB Conversion Communication
Bluetooth Low Energy Communication
RL78/G1D Evaluation Board (Peripehral Device)
RL78/G1D Evaluation Board (Central Device)

Figure 1-1 Overview of Sample Program Operating Environment

1.1 Function of Sample Program

- Peripheral Device
  Automatic control of advertisement
  Creation of dummy data of 20 bytes per packet and transmission processing
  Send up to 4 Notification packets per Connection Interval

- Central Device
  Automated search and connection of BD Address (stored in data flash) of connected Peripheral device
  Connection Interval value setting (stored in data flash)
  When terminal software is connected, received data is displayed as hexadecimal text data

1.2 Term/Abbreviation

The following show terms and abbreviations to use in this document.

Table 1-1 Term/Abbreviation

Term/Abbreviation	Description
BLE	Stands for "Bluetooth low energy"
BLE Device	Stands for "Bluetooth low energy device"
BPMEK	Stands for "Blood Pressure Monitoring Evaluation Kit for RL78/H1D (RTK0EH0003S02001BR)"
Notification	Stands for Characteristic Value Notification and indicates the procedure for sending (notifying) data to the client at the server timing.
Opposing Device	Indicates a BLE Central device on the data receiving side.
Peripheral Device	Indicates a device that uses RL78/G1D in Peripheral setting.
RFP	Stands for "Renesas Flash Programmer"
Renesas Rapid Transfer Profile	Rapid transfer profile shown in this application note is shown.
RRTP	Stands for "Renesas Rapid Transfer Profile"
RRTS	Stands for "Renesas Rapid Transfer Service"

1.3 Notes on Usage/Restrictions

(1) Depending on conditions such as radio waves between Peripheral device and Central device, received data may be lost in packet units (1 packet = 20 bytes of user data). In addition, packet reception order may be changed by retransmission processing between BLE devices.

(2) When connecting between Peripheral device and Central device, the connection may fail.

If the Peripheral device remains in the Advertising state when the connection fails, perform the connection processing again on the Central device.
If the Peripheral device has progressed to a state after Advertising when the connection has failed, restart both the Central device and the Peripheral device.

(3) Maximum throughput of user data from Peripheral (server) device to Central (client) device is 85.3 kbps.

The conditions for obtaining the maximum throughput are as follows:
— Connection Interval: 7.5 milliseconds
— Send 4 Notification packets during single Connection Interval period

In the case of the maximum throughput setting, packet loss and received data replacement are likely to occur.
If you set the number of transmission packets during the single Connection Interval period to 3 packets or less (throughput theoretical value 64 kbps or less), the frequency of packet loss decreases.

(4) It is recommended to set the main system clock frequency to 32MHz.

1.4 Development Environment

The following shows the development environment.

Table 1-2 Development Environment

Items	Description
Integrated Development Environment (IDE)	Renesas Electronics CS + for CC V7.00.00
C Compiler	Renesas Electronics CC-RL V1.07.00

1.5 File Structure

The following shows the file structure.

Since there are multiple identical file names (rBLE_Emb_CCRL.mot file) in the ROM_Files folder, the file update date is described for identification purpose only for these files.

Table 1-3 File Structure

```
r01an4569ej0101-bsspf-communication                                                    File creation date
├─r01an4569ej0101-bsspf-communication.pdf
├─ROM_Files
│  ├─BPMEK_RL78_G1D
│  │  └─rBLE_Emb_CCRL.mot                                                              : 2019/02/27 11:14
│  ├─BPMEK_RL78_H1D
│  │  └─BPM_solution_H1D.mot
│  ├─RL78_G1D_central
│  │  └─rBLE_Emb_CCRL.mot                                                              : 2019/02/22 13:54
│  └─RL78_G1D_peripheral
│      └─rBLE_Emb_CCRL.mot                                                             : 2019/02/26 16:58
├─tools
│  ├─IIR_filter_coefficient_OSR1024_reference.csv
│  ├─RRTP sample.log
│  └─RRTP_graph_sample_rev101.xlsm
├─unique_code
│  └─sample_central.ruc
├─workspace_RL78_G1D_central
│  └─Project_Source
│      ├─bleip
│      │  └─src
│      │      ├─common
│      │      │  └─co_bt.h
│      │      └─rwble
│      │          ├─rwble.h
│      │          └─rwble_config.h
│      ├─rBLE
│      │  └─src
│      │      ├─include
│      │      │  ├─rble.h
│      │      │  ├─rble_api.h
│      │      │  ├─rble_api_custom.h
│      │      │  ├─rble_api_fwup.h
│      │      │  ├─rble_api_rrtp.h
│      │      │  ├─rble_app.h
│      │      │  ├─rble_rwke.h
│      │      │  ├─rble_trans.h
│      │      │  └─rscip_api.h
│      │      ├─sample_app
│      │      │  ├─beacon_app.c
│      │      │  ├─beacon_app.h
│      │      │  ├─Console.c
│      │      │  ├─console.h
│      │      │  ├─menu_sel.c
│      │      │  ├─menu_sel.h
│      │      │  ├─rble_app_rrtpc.c
│      │      │  ├─rble_app_rrtpc.h
│      │      │  ├─rble_fw_up_receiver_app.c
│      │      │  ├─rble_sample_app.c
│      │      │  ├─rble_sample_app_anp.c
│      │      │  ├─rble_sample_app_blp.c
│      │      │  ├─rble_sample_app_cpp.c
│      │      │  ├─rble_sample_app_cscp.c
│   │   │   ├──rble_sample_app_custom.c
│   │   │   ├──rble_sample_app_fmp.c
│   │   │   ├──rble_sample_app_fwup.c
│   │   │   ├──rble_sample_app_gap_sm_gatt.c
│   │   │   ├──rble_sample_app_glp.c
│   │   │   ├──rble_sample_app_hogp.c
│   │   │   ├──rble_sample_app_hrp.c
│   │   │   ├──rble_sample_app_htp.c
│   │   │   ├──rble_sample_app_lnp.c
│   │   │   ├──rble_sample_app_pasp.c
│   │   │   ├──rble_sample_app_pxp.c
│   │   │   ├──rble_sample_app_rscp.c
│   │   │   ├──rble_sample_app_scpp.c
│   │   │   ├──rble_sample_app_tip.c
│   │   │   └──rble_sample_app_vendor.c
│   │   ├──sample_profile
│   │   │   ├──fwup
│   │   │   │   └──fwupr.c
│   │   │   ├──rrtp
│   │   │   │   ├──rtpc.c
│   │   │   │   └──rtps.c
│   │   │   └──scp
│   │   │       ├──scpc.c
│   │   │       └──scps.c
│   │   └──sample_simple
│   │       ├──console.c
│   │       ├──console.h
│   │       ├──rble_sample_app_peripheral.c
│   │       ├──rble_sample_app_peripheral.h
│   │       └──sam
│   │           ├──sam.h
│   │           ├──sams.c
│   │           └──sams.h
│   └──renesas
│       ├──config
│       │   └──split
│       │       └──emb
│       │           ├──r_lk.dr
│       │           ├──r_lk_fw_emb.dr
│       │           ├──r_lk_fw_modem.dr
│       │           ├──r_lk_modem.dr
│       │           ├──r_lk_modem_R5F11AGH.dr
│       │           └──r_lk_R5F11AGH.dr
│       ├──lib
│       │   ├──BLE_CONTROLLER_lib.a
│       │   ├──BLE_CONTROLLER_LIB.lib
│       │   ├──BLE_CONTROLLER_LIB_CCRL.lib
│       │   ├──BLE_HOST_lib.a
│       │   ├──BLE_HOST_lib.lib
│       │   ├──BLE_HOST_lib_CCRL.lib
│       │   ├──BLE_PROFILE_ANP_lib.a
│       │   ├──BLE_PROFILE_ANP_LIB.lib
│       │   ├──BLE_PROFILE_ANP_LIB_CCRL.lib
│       │   ├──BLE_PROFILE_BLP_lib.a
│       │   ├──BLE_PROFILE_BLP_LIB.lib
│       │   ├──BLE_PROFILE_BLP_LIB_CCRL.lib
│       │   ├──BLE_PROFILE_CPP_lib.a
│       │   ├──BLE_PROFILE_CPP_LIB.lib
│       │   ├──BLE_PROFILE_CPP_LIB_CCRL.lib
│       │   ├──BLE_PROFILE_CSP_lib.a
│       │   ├──BLE_PROFILE_CSP_LIB.lib
│       │   ├──BLE_PROFILE_CSP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_FMP_lib.a
│         │   ├─BLE_PROFILE_FMP_LIB.lib
│         │   ├─BLE_PROFILE_FMP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_GLP_lib.a
│         │   ├─BLE_PROFILE_GLP_LIB.lib
│         │   ├─BLE_PROFILE_GLP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_HGP_lib.a
│         │   ├─BLE_PROFILE_HGP_LIB.lib
│         │   ├─BLE_PROFILE_HGP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_HRP_lib.a
│         │   ├─BLE_PROFILE_HRP_LIB.lib
│         │   ├─BLE_PROFILE_HRP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_HTP_lib.a
│         │   ├─BLE_PROFILE_HTP_LIB.lib
│         │   ├─BLE_PROFILE_HTP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_LNP_lib.a
│         │   ├─BLE_PROFILE_LNP_LIB.lib
│         │   ├─BLE_PROFILE_LNP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_PAP_lib.a
│         │   ├─BLE_PROFILE_PAP_LIB.lib
│         │   ├─BLE_PROFILE_PAP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_PXP_lib.a
│         │   ├─BLE_PROFILE_PXP_LIB.lib
│         │   ├─BLE_PROFILE_PXP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_RSP_lib.a
│         │   ├─BLE_PROFILE_RSP_LIB.lib
│         │   ├─BLE_PROFILE_RSP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_SCP_lib.a
│         │   ├─BLE_PROFILE_SCP_LIB.lib
│         │   ├─BLE_PROFILE_SCP_LIB_CCRL.lib
│         │   ├─BLE_PROFILE_TIP_lib.a
│         │   ├─BLE_PROFILE_TIP_LIB.lib
│         │   ├─BLE_PROFILE_TIP_LIB_CCRL.lib
│         │   ├─BLE_PROFILES_COMMON_lib.a
│         │   ├─BLE_PROFILES_COMMON_LIB.lib
│         │   ├─BLE_PROFILES_COMMON_LIB_CCRL.lib
│         │   ├─BLE_rBLE_lib.a
│         │   ├─BLE_rBLE_lib.lib
│         │   └─BLE_rBLE_lib_CCRL.lib
│         ├─src
│         │   ├─arch
│         │   │   └─rl78
│         │   │       ├─arch.h
│         │   │       ├─arch_main.c
│         │   │       ├─config.h
│         │   │       ├─db_handle.h
│         │   │       ├─fw_update_count0.c
│         │   │       ├─fw_update_count1.c
│         │   │       ├─hw_config.h
│         │   │       ├─ke_conf.c
│         │   │       ├─ke_conf_simple.c
│         │   │       ├─ll
│         │   │       │   └─ll.h
│         │   │       ├─main.c
│         │   │       ├─prf_config.c
│         │   │       ├─prf_config.h
│         │   │       ├─prf_config_host.c
│         │   │       ├─prf_sel.h
│         │   │       ├─rble_core_config.c
│         │   │       ├─rble_core_config.h
│         │   │       ├─rble_modem_config.c
│         │   │       ├─rble_modem_config.h
│         │   │      ├─rwble_mem.c
│         │   │      ├─rwble_mem.h
│         │   │      └─rwke_api.h
│         │   ├─compiler
│         │   │   ├─ccrl
│         │   │   │   ├─cstart.asm
│         │   │   │   ├─hdwinit.asm
│         │   │   │   └─stkinit.asm
│         │   │   ├─compiler.h
│         │   │   └─iodefine.h
│         │   ├─driver
│         │   │   ├─codeflash
│         │   │   │   ├─cc_rl
│         │   │   │   │   ├─fsl.h
│         │   │   │   │   ├─fsl.lib
│         │   │   │   │   └─fsl_types.h
│         │   │   │   ├─codeflash.c
│         │   │   │   ├─codeflash.h
│         │   │   │   ├─cs
│         │   │   │   │   ├─fsl.h
│         │   │   │   │   ├─fsl.lib
│         │   │   │   │   └─fsl_types.h
│         │   │   │   └─iar_v2
│         │   │   │     ├─fsl.a
│         │   │   │     ├─fsl.h
│         │   │   │     └─fsl_types.h
│         │   │   ├─csi
│         │   │   │   ├─csi.c
│         │   │   │   └─csi.h
│         │   │   ├─dataflash
│         │   │   │   ├─cc_rl
│         │   │   │   │   ├─eel.h
│         │   │   │   │   ├─eel.lib
│         │   │   │   │   ├─eel_types.h
│         │   │   │   │   ├─fdl.h
│         │   │   │   │   ├─fdl.lib
│         │   │   │   │   └─fdl_types.h
│         │   │   │   ├─cs
│         │   │   │   │   ├─eel.h
│         │   │   │   │   ├─eel.lib
│         │   │   │   │   ├─eel_types.h
│         │   │   │   │   ├─fdl.h
│         │   │   │   │   ├─fdl.lib
│         │   │   │   │   └─fdl_types.h
│         │   │   │   ├─dataflash.c
│         │   │   │   ├─dataflash.h
│         │   │   │   ├─eel_descriptor_t02.c
│         │   │   │   ├─eel_descriptor_t02.h
│         │   │   │   ├─fdl_descriptor_t02.c
│         │   │   │   ├─fdl_descriptor_t02.h
│         │   │   │   └─iar_v2
│         │   │   │     ├─eel.a
│         │   │   │     ├─eel.h
│         │   │   │     ├─eel_types.h
│         │   │   │     ├─fdl.a
│         │   │   │     ├─fdl.h
│         │   │   │     └─fdl_types.h
│         │   │   ├─DTM2Wire
│         │   │   │   ├─DTM2Wire.c
│         │   │   │   └─DTM2Wire.h
│         │   │   ├─iic
│         │   │   │   ├─iic_slave.c
│         │   │   │   └─iic_slave.h
│         │   │   ├─led
│         │   │   │   ├─led.c
│         │   │   │   └─led.h
│         │   │   ├─led_onoff
│         │   │   │   ├─led_onoff.c
│         │   │   │   └─led_onoff.h
│         │   │   ├─peak
│         │   │   │   ├─peak.h
│         │   │   │   └─peak_isr.c
│         │   │   ├─pktmon
│         │   │   │   └─pktmon.h
│         │   │   ├─plf
│         │   │   │   ├─plf.c
│         │   │   │   └─plf.h
│         │   │   ├─port
│         │   │   │   └─port.h
│         │   │   ├─push_state
│         │   │   │   ├─push_state.c
│         │   │   │   └─push_state.h
│         │   │   ├─push_sw
│         │   │   │   ├─push_sw.c
│         │   │   │   └─push_sw.h
│         │   │   ├─rf
│         │   │   │   └─rf.h
│         │   │   ├─serial
│         │   │   │   └─serial.h
│         │   │   ├─uart
│         │   │   │   ├─uart.c
│         │   │   │   └─uart.h
│         │   │   └─wakeup
│         │   │       ├─wakeup.c
│         │   │       └─wakeup.h
│         │   └─types.h
│         └─tools
│             ├─project
│             │   ├─CS_CCRL
│             │   │   ├─BLE_Embedded
│             │   │   │   ├─BLE_Embedded.mtpj
│             │   │   │   ├─BLE_Embedded.rcpe
│             │   │   │   ├─DefaultBuild
│             │   │   │   └─rBLE_Emb
│             │   │   │       ├─DefaultBuild
│             │   │   │       │   ├─rBLE_Emb_CCRL.hex
│             │   │   │       │   ├─rBLE_Emb_CCRL.lmf
│             │   │   │       │   ├─rBLE_Emb_CCRL.map
│             │   │   │       │   └─rBLE_Emb_CCRL.sni
│             │   │   │       ├─rBLE_Emb.mtsp
│             │   │   │       ├─sect_emb.hsi
│             │   │   │       └─sect_emb_fwup.hsi
│             │   │   └─BLE_Modem
│             │   │       ├─BLE_Modem.mtpj
│             │   │       └─rBLE_Mdm
│             │   │           ├─rBLE_Mdm.mtsp
│             │   │           ├─sect_mdm.hsi
│             │   │           └─sect_mdm_fwup.hsi
│             │   ├─CubeSuite
│             │   │   ├─BLE_Embedded
│             │   │   │   ├─BLE_Emb
│             │   │   │   │   └─BLE_Emb.mtsp
│             │   │   │   └─BLE_Embedded.mtpj
│             │   │   └─BLE_Modem
│           │   │       ├─BLE_Modem.mtpj
│           │   │       └─rBLE_emb
│           │   │           └─rBLE_emb.mtsp
│           │   ├─e2studio
│           │   │   ├─BLE_Embedded
│           │   │   │   └─rBLE_Emb
│           │   │   │       ├─.cproject
│           │   │   │       ├─.DefaultBuildlinker
│           │   │   │       ├─.info
│           │   │   │       ├─.project
│           │   │   │       ├─rBLE_Emb_CCRL.x.launch
│           │   │   │       ├─sect_emb.esi
│           │   │   │       └─sect_emb_fwup.esi
│           │   │   └─BLE_Modem
│           │   │       └─rBLE_Mdm
│           │   │           ├─.cproject
│           │   │           ├─.DefaultBuildlinker
│           │   │           ├─.info
│           │   │           ├─.project
│           │   │           ├─rBLE_Mdm_CCRL.x.launch
│           │   │           ├─sect_mdm.esi
│           │   │           └─sect_mdm_fwup.esi
│           │   └─iar_v2
│           │       ├─BLE_Embedded
│           │       │   ├─BLE_Emb
│           │       │   │   └─BLE_Emb.ewp
│           │       │   └─BLE_Embedded.eww
│           │       ├─BLE_Modem
│           │       │   ├─BLE_Emb
│           │       │   │   └─BLE_Emb.ewp
│           │       │   └─BLE_Modem.eww
│           │       └─config
│           │           ├─lnkr5f11agj.icf
│           │           ├─lnkr5f11agj_fw.icf
│           │           └─lnkr5f11agj_fw_mdm.icf
│           ├─project_devices
│           │   ├─CS_CCRL
│           │   │   ├─BLE_Embedded
│           │   │   │   ├─BLE_Embedded_R5F11AGG.mtpj
│           │   │   │   ├─BLE_Embedded_R5F11AGH.mtpj
│           │   │   │   ├─rBLE_Emb_R5F11AGG
│           │   │   │   │   └─rBLE_Emb_R5F11AGG.mtsp
│           │   │   │   └─rBLE_Emb_R5F11AGH
│           │   │   │       └─rBLE_Emb_R5F11AGH.mtsp
│           │   │   └─BLE_Modem
│           │   │       ├─BLE_Modem_R5F11AGG.mtpj
│           │   │       ├─BLE_Modem_R5F11AGH.mtpj
│           │   │       ├─rBLE_Mdm_R5F11AGG
│           │   │       │   └─rBLE_Mdm_R5F11AGG.mtsp
│           │   │       └─rBLE_Mdm_R5F11AGH
│           │   │           └─rBLE_Mdm_R5F11AGH.mtsp
│           │   ├─CubeSuite
│           │   │   ├─BLE_Embedded
│           │   │   │   ├─BLE_Emb_R5F11AGH
│           │   │   │   │   └─BLE_Emb_R5F11AGH.mtsp
│           │   │   │   └─BLE_Embedded_R5F11AGH.mtpj
│           │   │   └─BLE_Modem
│           │   │       ├─BLE_Modem_R5F11AGH.mtpj
│           │   │       └─rBLE_emb_R5F11AGH
│           │   │           └─rBLE_emb_R5F11AGH.mtsp
│           │   ├─e2studio
│           │   │   ├─BLE_Embedded
│           │   │   │   ├─rBLE_Emb_R5F11AGG
│           │   │   │   │   ├─.cproject
│           │   │   │   │   ├─.DefaultBuildlinker
│           │   │   │   │   ├─.info
│           │   │   │   │   ├─.project
│           │   │   │   │   ├─rBLE_Emb_R5F11AGG_CCRL.x.launch
│           │   │   │   │   └─sect_emb.esi
│           │   │   │   └─rBLE_Emb_R5F11AGH
│           │   │   │       ├─.cproject
│           │   │   │       ├─.DefaultBuildlinker
│           │   │   │       ├─.info
│           │   │   │       ├─.project
│           │   │   │       ├─rBLE_Emb_R5F11AGH_CCRL.x.launch
│           │   │   │       └─sect_emb.esi
│           │   │   └─BLE_Modem
│           │   │       ├─rBLE_Mdm_R5F11AGG
│           │   │       │   ├─.cproject
│           │   │       │   ├─.DefaultBuildlinker
│           │   │       │   ├─.info
│           │   │       │   ├─.project
│           │   │       │   ├─rBLE_Mdm_R5F11AGG_CCRL.x.launch
│           │   │       │   └─sect_mdm.esi
│           │   │       └─rBLE_Mdm_R5F11AGH
│           │   │           ├─.cproject
│           │   │           ├─.DefaultBuildlinker
│           │   │           ├─.info
│           │   │           ├─.project
│           │   │           ├─rBLE_Mdm_R5F11AGH_CCRL.x.launch
│           │   │           └─sect_mdm.esi
│           │   └─iar_v2
│           │       ├─BLE_Embedded
│           │       │   ├─BLE_Emb_R5F11AGG
│           │       │   │   └─BLE_Emb_R5F11AGG.ewp
│           │       │   ├─BLE_Emb_R5F11AGH
│           │       │   │   └─BLE_Emb_R5F11AGH.ewp
│           │       │   ├─BLE_Embedded_R5F11AGG.eww
│           │       │   └─BLE_Embedded_R5F11AGH.eww
│           │       ├─BLE_Modem
│           │       │   ├─BLE_Emb_R5F11AGG
│           │       │   │   └─BLE_Emb_R5F11AGG.ewp
│           │       │   ├─BLE_Emb_R5F11AGH
│           │       │   │   └─BLE_Emb_R5F11AGH.ewp
│           │       │   ├─BLE_Modem_R5F11AGG.eww
│           │       │   └─BLE_Modem_R5F11AGH.eww
│           │       └─config
│           │           ├─lnkr5f11agg.icf
│           │           └─lnkr5f11agh.icf
│           └─project_simple
│               ├─CS_CCRL
│               │   └─BLE_Embedded
│               │       ├─BLE_Embedded.mtpj
│               │       └─rBLE_Emb
│               │           ├─rBLE_Emb.mtsp
│               │           └─sect_emb.hsi
│               ├─CubeSuite
│               │   └─BLE_Embedded
│               │       ├─BLE_Emb
│               │       │   └─BLE_Emb.mtsp
│               │       └─BLE_Embedded.mtpj
│               ├─e2studio
│               │   └─BLE_Embedded
│               │       └─rBLE_Emb
│                 │          ├─.cproject
│                 │          ├─.DefaultBuildlinker
│                 │          ├─.info
│                 │          ├─.project
│                 │          ├─rBLE_Emb_CCRL.x.launch
│                 │          └─sect_emb.esi
│                 └─iar_v2
│                     ├─BLE_Embedded
│                     │    ├─BLE_Emb
│                     │    │    └─BLE_Emb.ewp
│                     │    └─BLE_Embedded.eww
│                     └─config
│                          └─lnkr5f11agj.icf
└─workspace_RL78_G1D_peripheral
    └─Project_Source
        ├─bleip
        │    └─src
        │        ├─common
        │        │    └─co_bt.h
        │        └─rwble
        │            ├─rwble.h
        │            └─rwble_config.h
        ├─rBLE
        │    └─src
        │        ├─include
        │        │    ├─rble.h
        │        │    ├─rble_api.h
        │        │    ├─rble_api_custom.h
        │        │    ├─rble_api_fwup.h
        │        │    ├─rble_api_rrtp.h
        │        │    ├─rble_app.h
        │        │    ├─rble_external.h
        │        │    ├─rble_rwke.h
        │        │    ├─rble_trans.h
        │        │    └─rscip_api.h
        │        ├─sample_app
        │        │    ├─beacon_app.c
        │        │    ├─beacon_app.h
        │        │    ├─Console.c
        │        │    ├─console.h
        │        │    ├─menu_sel.c
        │        │    ├─menu_sel.h
        │        │    ├─r_rrtp.c
        │        │    ├─r_rrtp.h
        │        │    ├─rble_app_common.c
        │        │    ├─rble_app_common.h
        │        │    ├─rble_app_rrtp.c
        │        │    ├─rble_app_rrtp.h
        │        │    ├─rble_fw_up_receiver_app.c
        │        │    ├─rble_sample_app.c
        │        │    ├─rble_sample_app_anp.c
        │        │    ├─rble_sample_app_blp.c
        │        │    ├─rble_sample_app_cpp.c
        │        │    ├─rble_sample_app_cscp.c
        │        │    ├─rble_sample_app_custom.c
        │        │    ├─rble_sample_app_fmp.c
        │        │    ├─rble_sample_app_fwup.c
        │        │    ├─rble_sample_app_gap_sm_gatt.c
        │        │    ├─rble_sample_app_glp.c
        │        │    ├─rble_sample_app_hogp.c
        │        │    ├─rble_sample_app_hrp.c
        │        │    ├─rble_sample_app_htp.c
        │       │   ├──rble_sample_app_lnp.c
        │       │   ├──rble_sample_app_pasp.c
        │       │   ├──rble_sample_app_pxp.c
        │       │   ├──rble_sample_app_rscp.c
        │       │   ├──rble_sample_app_scpp.c
        │       │   ├──rble_sample_app_tip.c
        │       │   └──rble_sample_app_vendor.c
        │       ├──sample_profile
        │       │   ├──fwup
        │       │   │   └──fwupr.c
        │       │   ├──rrtp
        │       │   │   ├──rtpc.c
        │       │   │   └──rtps.c
        │       │   └──scp
        │       │       ├──scpc.c
        │       │       └──scps.c
        │       └──sample_simple
        │           ├──console.c
        │           ├──console.h
        │           ├──rble_sample_app_peripheral.c
        │           ├──rble_sample_app_peripheral.h
        │           └──sam
        │               ├──sam.h
        │               ├──sams.c
        │               └──sams.h
        └──renesas
            ├──config
            │   └──split
            │       └──emb
            │           ├──r_lk.dr
            │           ├──r_lk_fw_emb.dr
            │           ├──r_lk_fw_modem.dr
            │           ├──r_lk_modem.dr
            │           ├──r_lk_modem_R5F11AGH.dr
            │           └──r_lk_R5F11AGH.dr
            ├──lib
            │   ├──BLE_CONTROLLER_lib.a
            │   ├──BLE_CONTROLLER_LIB.lib
            │   ├──BLE_CONTROLLER_LIB_CCRL.lib
            │   ├──BLE_HOST_lib.a
            │   ├──BLE_HOST_lib.lib
            │   ├──BLE_HOST_lib_CCRL.lib
            │   ├──BLE_PROFILE_ANP_lib.a
            │   ├──BLE_PROFILE_ANP_LIB.lib
            │   ├──BLE_PROFILE_ANP_LIB_CCRL.lib
            │   ├──BLE_PROFILE_BLP_lib.a
            │   ├──BLE_PROFILE_BLP_LIB.lib
            │   ├──BLE_PROFILE_BLP_LIB_CCRL.lib
            │   ├──BLE_PROFILE_CPP_lib.a
            │   ├──BLE_PROFILE_CPP_LIB.lib
            │   ├──BLE_PROFILE_CPP_LIB_CCRL.lib
            │   ├──BLE_PROFILE_CSP_lib.a
            │   ├──BLE_PROFILE_CSP_LIB.lib
            │   ├──BLE_PROFILE_CSP_LIB_CCRL.lib
            │   ├──BLE_PROFILE_FMP_lib.a
            │   ├──BLE_PROFILE_FMP_LIB.lib
            │   ├──BLE_PROFILE_FMP_LIB_CCRL.lib
            │   ├──BLE_PROFILE_GLP_lib.a
            │   ├──BLE_PROFILE_GLP_LIB.lib
            │   ├──BLE_PROFILE_GLP_LIB_CCRL.lib
            │   ├──BLE_PROFILE_HGP_lib.a
            │   ├──BLE_PROFILE_HGP_LIB.lib
            │  ├─BLE_PROFILE_HGP_LIB_CCRL.lib
            │  ├─BLE_PROFILE_HRP_lib.a
            │  ├─BLE_PROFILE_HRP_LIB.lib
            │  ├─BLE_PROFILE_HRP_LIB_CCRL.lib
            │  ├─BLE_PROFILE_HTP_lib.a
            │  ├─BLE_PROFILE_HTP_LIB.lib
            │  ├─BLE_PROFILE_HTP_LIB_CCRL.lib
            │  ├─BLE_PROFILE_LNP_lib.a
            │  ├─BLE_PROFILE_LNP_LIB.lib
            │  ├─BLE_PROFILE_LNP_LIB_CCRL.lib
            │  ├─BLE_PROFILE_PAP_lib.a
            │  ├─BLE_PROFILE_PAP_LIB.lib
            │  ├─BLE_PROFILE_PAP_LIB_CCRL.lib
            │  ├─BLE_PROFILE_PXP_lib.a
            │  ├─BLE_PROFILE_PXP_LIB.lib
            │  ├─BLE_PROFILE_PXP_LIB_CCRL.lib
            │  ├─BLE_PROFILE_RSP_lib.a
            │  ├─BLE_PROFILE_RSP_LIB.lib
            │  ├─BLE_PROFILE_RSP_LIB_CCRL.lib
            │  ├─BLE_PROFILE_SCP_lib.a
            │  ├─BLE_PROFILE_SCP_LIB.lib
            │  ├─BLE_PROFILE_SCP_LIB_CCRL.lib
            │  ├─BLE_PROFILE_TIP_lib.a
            │  ├─BLE_PROFILE_TIP_LIB.lib
            │  ├─BLE_PROFILE_TIP_LIB_CCRL.lib
            │  ├─BLE_PROFILES_COMMON_lib.a
            │  ├─BLE_PROFILES_COMMON_LIB.lib
            │  ├─BLE_PROFILES_COMMON_LIB_CCRL.lib
            │  ├─BLE_rBLE_lib.a
            │  ├─BLE_rBLE_lib.lib
            │  └─BLE_rBLE_lib_CCRL.lib
            ├─src
            │  ├─arch
            │  │  └─rl78
            │  │      ├─arch.h
            │  │      ├─arch_main.c
            │  │      ├─config.h
            │  │      ├─db_handle.h
            │  │      ├─fw_update_count0.c
            │  │      ├─fw_update_count1.c
            │  │      ├─hw_config.h
            │  │      ├─ke_conf.c
            │  │      ├─ke_conf_simple.c
            │  │      ├─ll
            │  │      │  └─ll.h
            │  │      ├─main.c
            │  │      ├─prf_config.c
            │  │      ├─prf_config.h
            │  │      ├─prf_config_host.c
            │  │      ├─prf_sel.h
            │  │      ├─rble_core_config.c
            │  │      ├─rble_core_config.h
            │  │      ├─rble_modem_config.c
            │  │      ├─rble_modem_config.h
            │  │      ├─rwble_mem.c
            │  │      ├─rwble_mem.h
            │  │      └─rwke_api.h
            │  ├─compiler
            │  │  ├─ccrl
            │  │  │  ├─cstart.asm
            │  │  │  ├─hdwinit.asm
            │  │  │  └─stkinit.asm
            │  │  ├─compiler.h
            │  │  └─iodefine.h
            │  ├─driver
            │  │  ├─codeflash
            │  │  │  ├─cc_rl
            │  │  │  │  ├─fsl.h
            │  │  │  │  ├─fsl.lib
            │  │  │  │  └─fsl_types.h
            │  │  │  ├─codeflash.c
            │  │  │  ├─codeflash.h
            │  │  │  ├─cs
            │  │  │  │  ├─fsl.h
            │  │  │  │  ├─fsl.lib
            │  │  │  │  └─fsl_types.h
            │  │  │  └─iar_v2
            │  │  │      ├─fsl.a
            │  │  │      ├─fsl.h
            │  │  │      └─fsl_types.h
            │  │  ├─csi
            │  │  │  ├─csi.c
            │  │  │  └─csi.h
            │  │  ├─dataflash
            │  │  │  ├─cc_rl
            │  │  │  │  ├─eel.h
            │  │  │  │  ├─eel.lib
            │  │  │  │  ├─eel_types.h
            │  │  │  │  ├─fdl.h
            │  │  │  │  ├─fdl.lib
            │  │  │  │  └─fdl_types.h
            │  │  │  ├─cs
            │  │  │  │  ├─eel.h
            │  │  │  │  ├─eel.lib
            │  │  │  │  ├─eel_types.h
            │  │  │  │  ├─fdl.h
            │  │  │  │  ├─fdl.lib
            │  │  │  │  └─fdl_types.h
            │  │  │  ├─dataflash.c
            │  │  │  ├─dataflash.h
            │  │  │  ├─eel_descriptor_t02.c
            │  │  │  ├─eel_descriptor_t02.h
            │  │  │  ├─fdl_descriptor_t02.c
            │  │  │  ├─fdl_descriptor_t02.h
            │  │  │  └─iar_v2
            │  │  │      ├─eel.a
            │  │  │      ├─eel.h
            │  │  │      ├─eel_types.h
            │  │  │      ├─fdl.a
            │  │  │      ├─fdl.h
            │  │  │      └─fdl_types.h
            │  │  ├─DTM2Wire
            │  │  │  ├─DTM2Wire.c
            │  │  │  └─DTM2Wire.h
            │  │  ├─iic
            │  │  │  ├─iic_slave.c
            │  │  │  └─iic_slave.h
            │  │  ├─led
            │  │  │  ├─led.c
            │  │  │  └─led.h
            │  │  ├─led_onoff
            │  │  │  ├─led_onoff.c
            │  │  │  └─led_onoff.h
            │  │  ├─peak
            │   │   │   ├─peak.h
            │   │   │   └─peak_isr.c
            │   │   ├─pktmon
            │   │   │   └─pktmon.h
            │   │   ├─plf
            │   │   │   ├─plf.c
            │   │   │   └─plf.h
            │   │   ├─port
            │   │   │   └─port.h
            │   │   ├─push_state
            │   │   │   ├─push_state.c
            │   │   │   └─push_state.h
            │   │   ├─push_sw
            │   │   │   ├─push_sw.c
            │   │   │   └─push_sw.h
            │   │   ├─rf
            │   │   │   └─rf.h
            │   │   ├─serial
            │   │   │   └─serial.h
            │   │   ├─uart
            │   │   │   ├─uart.c
            │   │   │   └─uart.h
            │   │   └─wakeup
            │   │       ├─wakeup.c
            │   │       └─wakeup.h
            │   └─types.h
            └─tools
                ├─project
                │   ├─CS_CCRL
                │   │   ├─BLE_Embedded
                │   │   │   ├─BLE_Embedded.mtpj
                │   │   │   ├─BLE_Embedded.rcpe
                │   │   │   └─rBLE_Emb
                │   │   │       ├─rBLE_Emb.mtsp
                │   │   │       ├─sect_emb.hsi
                │   │   │       └─sect_emb_fwup.hsi
                │   │   └─BLE_Modem
                │   │       ├─BLE_Modem.mtpj
                │   │       └─rBLE_Mdm
                │   │           ├─rBLE_Mdm.mtsp
                │   │           ├─sect_mdm.hsi
                │   │           └─sect_mdm_fwup.hsi
                │   ├─CubeSuite
                │   │   ├─BLE_Embedded
                │   │   │   ├─BLE_Emb
                │   │   │   │   └─BLE_Emb.mtsp
                │   │   │   └─BLE_Embedded.mtpj
                │   │   └─BLE_Modem
                │   │       ├─BLE_Modem.mtpj
                │   │       └─rBLE_emb
                │   │           └─rBLE_emb.mtsp
                │   ├─e2studio
                │   │   ├─BLE_Embedded
                │   │   │   └─rBLE_Emb
                │   │   │       ├─.cproject
                │   │   │       ├─.DefaultBuildlinker
                │   │   │       ├─.info
                │   │   │       ├─.project
                │   │   │       ├─rBLE_Emb_CCRL.x.launch
                │   │   │       ├─sect_emb.esi
                │   │   │       └─sect_emb_fwup.esi
                │   │   └─BLE_Modem
                │  │      └─rBLE_Mdm
                │  │         ├─.cproject
                │  │         ├─.DefaultBuildlinker
                │  │         ├─.info
                │  │         ├─.project
                │  │         ├─rBLE_Mdm_CCRL.x.launch
                │  │         ├─sect_mdm.esi
                │  │         └─sect_mdm_fwup.esi
                │  └─iar_v2
                │     ├─BLE_Embedded
                │     │  ├─BLE_Emb
                │     │  │  └─BLE_Emb.ewp
                │     │  └─BLE_Embedded.eww
                │     ├─BLE_Modem
                │     │  ├─BLE_Emb
                │     │  │  └─BLE_Emb.ewp
                │     │  └─BLE_Modem.eww
                │     └─config
                │        ├─lnkr5f11agj.icf
                │        ├─lnkr5f11agj_fw.icf
                │        └─lnkr5f11agj_fw_mdm.icf
                ├─project_devices
                │  ├─CS_CCRL
                │  │  ├─BLE_Embedded
                │  │  │  ├─BLE_Embedded_R5F11AGG.mtpj
                │  │  │  ├─BLE_Embedded_R5F11AGH.mtpj
                │  │  │  ├─rBLE_Emb_R5F11AGG
                │  │  │  │  └─rBLE_Emb_R5F11AGG.mtsp
                │  │  │  └─rBLE_Emb_R5F11AGH
                │  │  │     └─rBLE_Emb_R5F11AGH.mtsp
                │  │  └─BLE_Modem
                │  │     ├─BLE_Modem_R5F11AGG.mtpj
                │  │     ├─BLE_Modem_R5F11AGH.mtpj
                │  │     ├─rBLE_Mdm_R5F11AGG
                │  │     │  └─rBLE_Mdm_R5F11AGG.mtsp
                │  │     └─rBLE_Mdm_R5F11AGH
                │  │        └─rBLE_Mdm_R5F11AGH.mtsp
                │  ├─CubeSuite
                │  │  ├─BLE_Embedded
                │  │  │  ├─BLE_Emb_R5F11AGH
                │  │  │  │  └─BLE_Emb_R5F11AGH.mtsp
                │  │  │  └─BLE_Embedded_R5F11AGH.mtpj
                │  │  └─BLE_Modem
                │  │     ├─BLE_Modem_R5F11AGH.mtpj
                │  │     └─rBLE_emb_R5F11AGH
                │  │        └─rBLE_emb_R5F11AGH.mtsp
                │  ├─e2studio
                │  │  ├─BLE_Embedded
                │  │  │  ├─rBLE_Emb_R5F11AGG
                │  │  │  │  ├─.cproject
                │  │  │  │  ├─.DefaultBuildlinker
                │  │  │  │  ├─.info
                │  │  │  │  ├─.project
                │  │  │  │  ├─rBLE_Emb_R5F11AGG_CCRL.x.launch
                │  │  │  │  └─sect_emb.esi
                │  │  │  └─rBLE_Emb_R5F11AGH
                │  │  │     ├─.cproject
                │  │  │     ├─.DefaultBuildlinker
                │  │  │     ├─.info
                │  │  │     ├─.project
                │  │  │     ├─rBLE_Emb_R5F11AGH_CCRL.x.launch
                │  │  │     └─sect_emb.esi
                │   │   └─BLE_Modem
                │   │       ├─rBLE_Mdm_R5F11AGG
                │   │       │   ├─.cproject
                │   │       │   ├─.DefaultBuildlinker
                │   │       │   ├─.info
                │   │       │   ├─.project
                │   │       │   ├─rBLE_Mdm_R5F11AGG_CCRL.x.launch
                │   │       │   └─sect_mdm.esi
                │   │       └─rBLE_Mdm_R5F11AGH
                │   │           ├─.cproject
                │   │           ├─.DefaultBuildlinker
                │   │           ├─.info
                │   │           ├─.project
                │   │           ├─rBLE_Mdm_R5F11AGH_CCRL.x.launch
                │   │           └─sect_mdm.esi
                │   └─iar_v2
                │       ├─BLE_Embedded
                │       │   ├─BLE_Emb_R5F11AGG
                │       │   │   └─BLE_Emb_R5F11AGG.ewp
                │       │   ├─BLE_Emb_R5F11AGH
                │       │   │   └─BLE_Emb_R5F11AGH.ewp
                │       │   ├─BLE_Embedded_R5F11AGG.eww
                │       │   └─BLE_Embedded_R5F11AGH.eww
                │       ├─BLE_Modem
                │       │   ├─BLE_Emb_R5F11AGG
                │       │   │   └─BLE_Emb_R5F11AGG.ewp
                │       │   ├─BLE_Emb_R5F11AGH
                │       │   │   └─BLE_Emb_R5F11AGH.ewp
                │       │   ├─BLE_Modem_R5F11AGG.eww
                │       │   └─BLE_Modem_R5F11AGH.eww
                │       └─config
                │           ├─lnkr5f11agg.icf
                │           └─lnkr5f11agh.icf
                └─project_simple
                    ├─CS_CCRL
                    │   └─BLE_Embedded
                    │       ├─BLE_Embedded.mtpj
                    │       └─rBLE_Emb
                    │           ├─rBLE_Emb.mtsp
                    │           └─sect_emb.hsi
                    ├─CubeSuite
                    │   └─BLE_Embedded
                    │       ├─BLE_Emb
                    │       │   └─BLE_Emb.mtsp
                    │       └─BLE_Embedded.mtpj
                    ├─e2studio
                    │   └─BLE_Embedded
                    │       └─rBLE_Emb
                    │           ├─.cproject
                    │           ├─.DefaultBuildlinker
                    │           ├─.info
                    │           ├─.project
                    │           ├─rBLE_Emb_CCRL.x.launch
                    │           └─sect_emb.esi
                    └─iar_v2
                        ├─BLE_Embedded
                        │   ├─BLE_Emb
                        │   │   └─BLE_Emb.ewp
                        │   └─BLE_Embedded.eww
                        └─config
                            └─lnkr5f11agj.icf
```

2. RRTP (Renesas Rapid Transfer Profile)

2.1 Overview of RRTP

RRTP is a rapid transfer profile developed based on a custom profile attached to BLE software for RL78/G1D.

This RRTP increases the amount of data per unit time by the following method.

- Central device connects the Connection Interval value with a small value (minimum value according to standard: 7.5 milliseconds)
- During communication, the Peripheral (server) device transmits up to 4 Notification1 packets per Connection Interval by MD (More Data) bit control included in the header of the data channel

Note 1: It indicates the procedure of Characteristic Value Notification.

With the above control, the following throughput can be achieved for 20 bytes of user data.

Table 2-1 Throughput of User Data by RRTP (Design Value)

Number of Packets in Single Connection Interval	Throughput of User Data (Design Value)
4	(20 bytes × 4 packets) / 7.5 milliseconds = 10,666 bytes/second = 85.33 kbps
3	(20 bytes × 3 packets) / 7.5 milliseconds = 8,000 bytes/second = 64 kbps
2	(20 bytes × 2 packets) / 7.5 milliseconds = 5,333 bytes/second = 42.67 kbps
1	(20 bytes × 1 packets) / 7.5 milliseconds = 2,666 bytes/second = 21.33 kbps

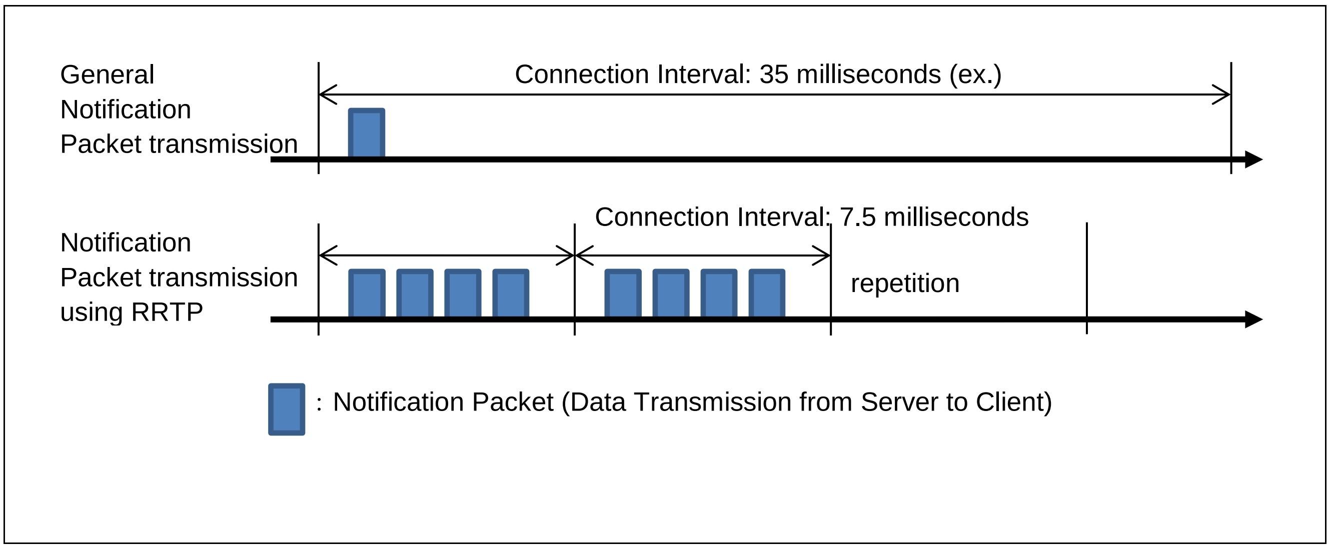

General Notification Packet transmission
Connection Interval: 35 milliseconds (ex.)
Notification Packet transmission using RRTP
Connection Interval: 7.5 milliseconds
repetition
: Notification Packet (Data Transmission from Server to Client)

Figure 2-1 Notification by RRTP Notification Packet (Data Transmission from Server)

2.1.1 Connection Interval Setting

When connection is completed and communication is in progress, the data channel sends and receives data at regular intervals according to the timing of the Central device. This timing is called Connection Event. Also, the interval is called Connection Interval. The minimum value on the Bluetooth low energy standard is 7.5 milliseconds.

In RRTP, Connection Interval is set short to improve throughput.

2.1.2 Multiple Characteristic Value Notification Packets Transmission using MD Bit Control

There is an MD bit in the header of Data Channel PDU (Protocol Data Unit).

The MD is a bit indicating that there is data to be sent to the connection destination. If MD = 0, it indicates that there is no subsequent data, and if MD = 1, it indicates that there is subsequent data. Therefore, if MD = 1, communication continues by extending the Connection Event.

If multiple user data are delivered to BLE software during single Connection Interval, the BLE software controls MD bit and sends multiple Data Channel PDU during single Connection Interval.

LSB		MSB
Header (16 bits)	Payload [Note 1] (27 bytes)	MIC (32 bits)

Note 1: The first 7 bytes of Payload are used for L2CAP and ATT headers, etc. The Payload available to the user is 20 bytes.

Figure 2-2 Data Channel PDU Structure (for Bluetooth Ver. 4.1)

Header						
LLID (2 bits)	NESN (1 bit)	SN (1 bit)	**MD** (1 bit)	RFU (3 bits)	Length (5 bits)	RFU (3 bits)

Figure 2-3 PDU Header Structure (for Bluetooth Ver. 4.1)

In addition, there are the following procedures for sending data from the server to the client:

- Characteristic Value Read
  The client periodically requests the server to read and the server sends data as the response.

- Characteristic Value Notification (hereafter referred to as "Notification")
  This is a BLE-specific procedure in which the client receives a notification when the measured values etc. are updated. Send data to the client at the server timing.

- Characteristic Value Indication
  Sends data from server to client. However, the client needs a response to the server.

In RRTP, Notification during communication minimizes communication between server and client. As a result, Peripheral (server) devices do not need RF reception processing and perform only RF transmission processing.

2.2 Specification of RRTP

A unique GATT-based profile is defined for RRTP control.

2.2.1 Profile Specification

The following shows the specification of RRTP of the sample program.

(1) Role

- RL78/G1D of the Peripheral device to which the sensor is connected is an RRTP server.
  The server has the Renesas Rapid Transfer Service (RRTS).

- A Central device that controls the sensor by connecting to the RRTP server is an RRTP client.
  The client accesses the RRTS server.
  In this application note, an RL78/G1D evaluation board is the client.

(2) Service and Characteristic

- RRTS is configured with Characteristic to control data transmission
- The server notifies clients of Characteristic Value using Notification

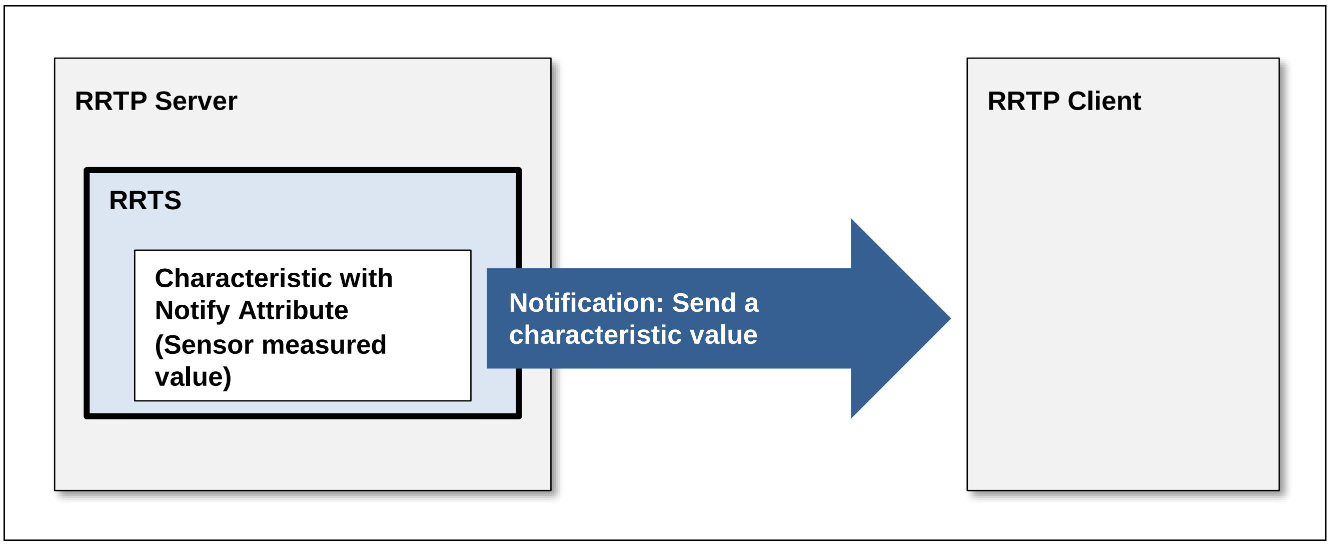

RRTP Server
RRTS
Characteristic with Notify Attribute (Sensor measured value)
Notification: Send a characteristic value
RRTP Client

Figure 2-4 RRTP

2.2.2 Service Specification

The following shows the specification of RRTS of the sample program.

Table 2-2 Specification of RRTS

Attribute Handle	**Attribute Type**	**Attribute Value**
Renesas Rapid Transfer Service		
0x000C	Primary Service Declaration (0x2800)	UUID: E01B0001-BA28-466A-88FA-337721DC020B
Rapid Transfer Data Value Characteristic (Notify Property)		
0x000D	Characteristic Declaration (0x2803)	Properties: Notify (0x10) Value Handle: 0x000E UUID: E01B0002-BA28-466A-88FA-337721DC020B
0x000E	Characteristic Value (0xFFFF)	Transfer data (20-byte) [Note 1]
0x000F	Characteristic Declaration (0x2803)	Properties: Notify (0x10) Value Handle: 0x0010 UUID: E01B0002-BA28-466A-88FA-337721DC020B
0x0010	Characteristic Value (0xFFFF)	Transfer data (20-byte) [Note 1]
0x0011	Characteristic Declaration (0x2803)	Properties: Notify (0x10) Value Handle: 0x0012 UUID: E01B0002-BA28-466A-88FA-337721DC020B
0x0012	Characteristic Value (0xFFFF)	Transfer data (20-byte) [Note 1]
0x0013	Characteristic Declaration (0x2803)	Properties: Notify (0x10) Value Handle: 0x0014 UUID: E01B0002-BA28-466A-88FA-337721DC020B
0x0014	Characteristic Value (0xFFFF)	Transfer data (20-byte) [Note 1]
0x0015	Client Characteristic Configuration (0x2902)	Properties: Read, Write (0x0A) Notification Configuration (2-byte) [Note 2]

Note 1: It indicates user data to be transferred using RRTP.
Mote 2: It is 0x0001 (= RBLE_RRTP_START_NTF) and indicates the start of Notification.

2.3 Based Software

Both Central and Peripheral devices use the same BLE software as a base.

Download the BLE software from below.

- Bluetooth low energy Protocol Stack (Ver.1.21) RTM5F11A00NBLE0F10RZ-V0121.zip
- https://www.renesas.com/jp/ja/products/software-tools/software-os-middleware-driver/protocol-stack/ble-protocol-stack.html#downloads

2.3.1 Project Layer

The project to use is shown below.

```
RTM5F11A00NBLE0F10RZ-V0121 (ZIP File)
      └SoftwarePackage
           └BLE_Software_Ver_1_21 (ZIP File)
                └Renesas
                     └BLE_Software_Ver_1_21
                          └RL78_G1D
                               └Project_Source（WORKSPACE Folder）
```

Figure 2-5 Execution Environment Project Layer (v1.21)

2.4 Peripheral Device

The Peripheral device automatically starts advertising after power on. Also, when the connection with the Central device is confirmed, transmission of 20 bytes of data per packet starts.

2.4.1 List of Changed files

The following shows the differences from the base software of the BLE software of the Peripheral device.

Table 2-3 List of Changed files

Folder	File name	treatment	Remarks
rBLE/src/include	rble_api_rrtp.h	Added	Definitions of constants and structures used by RRTP
rBLE/src/include	rble_external.h	Added	Definition of structure used at initialization
rBLE/src/sample_app	r_rrtp.c	Added	Functions related to send queue
rBLE/src/sample_app	r_rrtp.h	Added	Header file associated with r_rrtp.c
rBLE/src/sample_app	rble_app_common.c	Added	Common functions such as rble initialization
rBLE/src/sample_app	rble_app_common.h	Added	Header file for rble_app_common.c
rBLE/src/sample_app	rble_app_rrtp.c	Added	RRTP API and callback function
rBLE/src/sample_app	rble_app_rrtp.h	Added	Header file for rble_app_rrtp.c
rBLE/src/sample_profile/rrtp	rtps.c	Added	Service processing functions of RRTP client
renesas/src/arch/rl78	arch_main.c	Modified	
renesas/src/arch/rl78	db_handle.h	Modified	
renesas/src/arch/rl78	main.c	Modified	
renesas/src/arch/rl78	prf_config.c	Modified	
renesas/src/arch/rl78	prf_config.h	Modified	
renesas/src/arch/rl78	prf_sel.h	Modified	

2.4.2 Change Difference Details

The following shows the difference information from the BLE software. The line number mainly indicates the line position after the change.

The "-" at the beginning of the line indicates the deletion of the line and the "+" at the beginning of the line indicates the addition of the line.

(1) arch_main.c

(a) Added to line 59 (lines 59 to 62 of the sample code)

```
+#include "rble_external.h"
+#include "rble_app_common.h"
+#include "rble_app_rrtp.h"
+
```

(b) Added to line 68 (lines 72 to 88 of the sample code)

Adds definitions and initializes variables for automatic connection.

```
-
+#include "r_rrtp.h"
+
+typedef enum
+{
```

```
+        E_RF_STATE_UNINITIALIZED = 0,
+        E_RF_STATE_REQ_INITIALIZE,
+        E_RF_STATE_WAIT_INITIALIZE,
+        E_RF_STATE_REQ_RF_ENABLE,
+        E_RF_STATE_WAIT_RF_ENABLE,
+        E_RF_STATE_REQ_ADVERTISE,
+        E_RF_STATE_WAIT_ADVERTISE,
+        E_RF_STATE_WAIT_CONNECTION,
+        E_RF_STATE_WAIT_SERVER_ENABLE,
+        E_RF_STATE_CONNECTED,
+} e_rf_state_t;
+
+e_rf_state_t g_rf_state = E_RF_STATE_UNINITIALIZED;
```

(c) Added to line 77 (lines 97 to 99 of the sample code)

Adds prototype declarations for functions for automatic connection and data transfer.

```
+void rble_rf_control(void);
+void rrtp_data_send(void);
+
```

(d) Added to line 317 (lines 340 to 341 of the sample code)

Performs automatic connection variable initialization and BLE software initialization.

```
+    g_rble_rf_status = RBLE_RF_UNINITIALIZED;
+    RBLE_App_Init();
```

(e) Added to line 388 (lines 413 to 420 of the sample code)

Adds automatic connection sequence processing call and processing for data transfer to the main routine.

```
+        rble_rf_control();
+
+        if (E_RF_STATE_CONNECTED == g_rf_state)
+        {
+            rrtp_data_send ();
+            RBLE_App_RapidTransferHandler();
+        }
+
```

(f) Added to line 451 (lines 484 of the sample code)

Changes the return value of the judgment function to disable sleep.

```
-    return true;
+    return false;
```

(g) Added to line 483 (lines 516 to 665 of the sample code)

Adds functions for automatic connection and data transfer.

```
+void rrtp_data_send(void)
+{
+    uint8_t data_array[D_RAPID_TRANSFER_BUFFER_SIZE];
+    uint8_t data_len = D_RAPID_TRANSFER_BUFFER_SIZE;
+
+    static uint8_t serial_num = 0;
+
+    data_array[0] = serial_num;
+
+    if (TRUE == R_RRTP_DataEnqueue(data_array, data_len))
+    {
+        serial_num++;
+    }
+    else
+    {
```

```
+           /* Do nothing */
+           __nop();
+       }
+}
+
+void rble_rf_control(void)
+{
+    uint8_t             temp;
+    st_init_api_info_t init_api_info;
+    const uint8_t       gs_device_name[] = "RSSK-RRTP-TEST";
+
+        switch(g_rf_state)
+        {
+            case E_RF_STATE_UNINITIALIZED:
+                {
+                    g_rf_state++;
+                }
+                break;
+            case E_RF_STATE_REQ_INITIALIZE:
+                {
+                    for (temp = 0; temp < RF_DEVICE_NAME_SZ; temp++)
+                    {
+                        if (temp < sizeof(gs_device_name))
+                        {
+                            init_api_info.device_name[temp] =
gs_device_name[temp];
+                        }
+                        else
+                        {
+                            init_api_info.device_name[temp] = 0x00;
+                        }
+                    }
+
+                    init_api_info.service_list.list_num     = 0x05;
+                    init_api_info.service_list.service_list = 0x0010;
+                    init_api_info.iocap                     =
RBLE_IO_CAP_NO_INPUT_NO_OUTPUT;
+                    init_api_info.auth                      =
RBLE_AUTH_REQ_NO_MITM_BOND;
+                    init_api_info.op.context                = 0x01;
+
+                    RBLE_App_InitializeAPIRequested(&init_api_info);
+
+                    g_rf_state++;
+                }
+                break;
+            case E_RF_STATE_WAIT_INITIALIZE:
+                {
+                    if (RBLE_RF_INITIALIZED == g_rble_rf_status)
+                    {
+                        g_rf_state++;
+                    }
+                    else
+                    {
+                        /* Do nothing */
+                        __nop();
+                    }
+                }
+                break;
+            case E_RF_STATE_REQ_RF_ENABLE:
+                {
+                    RBLE_App_EnableRfRequested(FALSE);
+                    g_rf_state++;
+                }
+                break;
+            case E_RF_STATE_WAIT_RF_ENABLE:
+                {
+                    if (RBLE_RF_ENABLED == g_rble_rf_status)
+                    {
+                        g_rf_state++;
+                    }
+                    else
+                    {
+                        /* Do nothing */
+                        __nop();
+                    }
+                }
+                break;
+            case E_RF_STATE_REQ_ADVERTISE:
+                {
+                    RBLE_App_StartAdvertiseRequested();
+                    g_rf_state++;
+                }
+                break;
+            case E_RF_STATE_WAIT_ADVERTISE:
+                {
+                    if (RBLE_RF_ADVERTISING == g_rble_rf_status)
+                    {
+                        g_rf_state++;
+                    }
+                    else
+                    {
+                        /* Do nothing */
+                        __nop();
+                    }
+                }
+                break;
+            case E_RF_STATE_WAIT_CONNECTION:
+                {
+                    if (RBLE_RF_CONNECTED == g_rble_rf_status)
+                    {
+                        g_rf_state++;
+                    }
+                    else
+                    {
+                        /* Do nothing */
+                        __nop();
+                    }
+                }
+                break;
+            case E_RF_STATE_WAIT_SERVER_ENABLE:
+                {
+                    if (RBLE_RF_SERVER_ENABLED == g_rble_rf_status)
+                    {
+                        g_rf_state++;
+                        g_rrtp_moredata = 3U;
+                     }
+                     else
+                     {
+                         /* Do nothing */
+                         __nop();
+                     }
+                 }
+                 break;
+             case E_RF_STATE_CONNECTED:
+                 {
+                     /* Do nothing */
+                     __nop();
+                 }
+                 break;
+             default:
+                 {
+                     /* Do nothing */
+                     __nop();
+                 }
+                 break;
+         }
+}
```

(2) db_handle.h

(a) Added to line 413 (lines 413 to 429 of the sample code)

Adds handle definition of RRTP.

```
+    #if (PRF_SEL_RTPS)
+    /* Renesas Rapid Transfer Service */
+    RRTS_HDL_SVC,
+    RRTS_HDL_NOTIFY_CHAR1,
+    RRTS_HDL_NOTIFY_VAL1,
+    RRTS_HDL_NOTIFY_CHAR2,
+    RRTS_HDL_NOTIFY_VAL2,
+    RRTS_HDL_NOTIFY_CHAR3,
+    RRTS_HDL_NOTIFY_VAL3,
+    RRTS_HDL_NOTIFY_CHAR4,
+    RRTS_HDL_NOTIFY_VAL4,
+    RRTS_HDL_NOTIFY_CFG,
+    RRTS_HDL_IND_CHAR,
+    RRTS_HDL_IND_VAL,
+    RRTS_HDL_IND_CFG,
+    #endif /* #if (PRF_SEL_RTPS) */
+
```

(3) main.c

(a) Added to line 101 (lines 101 to 102 of the sample code)

Refers setting of variable for more data control.

```
+extern uint16_t more_data_count;
+
```

(b) Added to line 476 (lines 478 of the sample code)

Resets the More Data control variable with the peak interrupt.

```
+    more_data_count = 0;
```

(4) prf_config.c

(a) Added to line 1238 (lines 1238 to 1294 of the sample code)

Defines service data for RRTP.

```
+#if (PRF_SEL_RTPS)
+/**********************************
+ * Renesas Rapid Transfer Service *
+ **********************************/
+/* Service (rtps) */
+static const uint8_t rtps_svc[RBLE_GATT_128BIT_UUID_OCTET] =
RBLE_SVC_RAPID_TRANSFER;
+
+/* Renesas Rapid Transfer Service Notify characteristic 1 */
+static const struct atts_char128_desc rtps_notify_char1 =
{ RBLE_GATT_CHAR_PROP_NTF,
+
{(uint8_t)(RRTS_HDL_NOTIFY_VAL1 & 0xff), (uint8_t)((RRTS_HDL_NOTIFY_VAL1 >> 8)
& 0xff)},
+
RBLE_CHAR_RRTP_NOTIFY };
+uint8_t rtps_notify_char_val1[RBLE_ATTM_MAX_VALUE] = {0};
+struct atts_elmt_128 rtps_notify_char_val_elmt1 = { RBLE_CHAR_RRTP_NOTIFY,
+
RBLE_GATT_128BIT_UUID_OCTET,
+
&rtps_notify_char_val1[0] };
+
+/* Renesas Rapid Transfer Service Notify characteristic 2 */
+static const struct atts_char128_desc rtps_notify_char2 =
{ RBLE_GATT_CHAR_PROP_NTF,
+
{(uint8_t)(RRTS_HDL_NOTIFY_VAL2 & 0xff), (uint8_t)((RRTS_HDL_NOTIFY_VAL2 >> 8)
& 0xff)},
+
RBLE_CHAR_RRTP_NOTIFY };
+uint8_t rtps_notify_char_val2[RBLE_ATTM_MAX_VALUE] = {0};
+struct atts_elmt_128 rtps_notify_char_val_elmt2 = { RBLE_CHAR_RRTP_NOTIFY,
+
RBLE_GATT_128BIT_UUID_OCTET,
+
&rtps_notify_char_val2[0] };
+
+/* Renesas Rapid Transfer Service Notify characteristic 3 */
+static const struct atts_char128_desc rtps_notify_char3 =
{ RBLE_GATT_CHAR_PROP_NTF,
+
{(uint8_t)(RRTS_HDL_NOTIFY_VAL3 & 0xff), (uint8_t)((RRTS_HDL_NOTIFY_VAL3 >> 8)
& 0xff)},
+
RBLE_CHAR_RRTP_NOTIFY };
+uint8_t rtps_notify_char_val3[RBLE_ATTM_MAX_VALUE] = {0};
+struct atts_elmt_128 rtps_notify_char_val_elmt3 = { RBLE_CHAR_RRTP_NOTIFY,
+
RBLE_GATT_128BIT_UUID_OCTET,
+
&rtps_notify_char_val3[0] };
+
+/* Renesas Rapid Transfer Service Notify characteristic 4 */
```

```
+static const struct atts_char128_desc rtps_notify_char4 =
{ RBLE_GATT_CHAR_PROP_NTF,
+
{(uint8_t)(RRTS_HDL_NOTIFY_VAL4 & 0xff), (uint8_t)((RRTS_HDL_NOTIFY_VAL4 >> 8)
& 0xff)},
+
RBLE_CHAR_RRTP_NOTIFY };
+uint8_t rtps_notify_char_val4[RBLE_ATTM_MAX_VALUE] = {0};
+struct atts_elmt_128 rtps_notify_char_val_elmt4 = { RBLE_CHAR_RRTP_NOTIFY,
+
RBLE_GATT_128BIT_UUID_OCTET,
+
&rtps_notify_char_val4[0] };
+
+uint16_t rtps_notify_en = 0x0000u;
+
+/* Renesas Rapid Transfer Service Indicate characteristic */
+static const struct atts_char128_desc rtps_ind_char =
{ RBLE_GATT_CHAR_PROP_IND,
+
{(uint8_t)(RRTS_HDL_IND_VAL & 0xff), (uint8_t)((RRTS_HDL_IND_VAL >> 8) &
0xff)},
+
RBLE_CHAR_RRTP_INDICATE };
+uint8_t rtps_ind_char_val[RBLE_ATTM_MAX_VALUE] = {0};
+struct atts_elmt_128 rtps_ind_char_val_elmt = { RBLE_CHAR_RRTP_INDICATE,
+                                                RBLE_GATT_128BIT_UUID_OCTET,
+                                                &rtps_ind_char_val[0] };
+
+uint16_t rtps_ind_en = 0x0000u;
+#endif /* (PRF_SEL_RTPS) */
```

(b) Added to line 2135 (lines 2193 to 2236 of the sample code)

Registers service data of RRTP in database.

```
+#if (PRF_SEL_RTPS)
+    /**********************************
+     * Renesas Rapid Transfer Service *
+     **********************************/
+    { RBLE_DECL_PRIMARY_SERVICE,
+        sizeof(rtps_svc), sizeof(rtps_svc), TASK_ATTID(TASK_RBLE,
RRTS_IDX_SVC), RBLE_GATT_PERM_RD, (void *)&rtps_svc },
+    /* Renesas Rapid Transfer Notify1 Char */
+    { RBLE_DECL_CHARACTERISTIC,
+        sizeof(rtps_notify_char1), sizeof(rtps_notify_char1),
TASK_ATTID(TASK_RBLE, RRTS_IDX_NOTIFY_CHAR1), RBLE_GATT_PERM_RD, (void
*)&rtps_notify_char1 },
+    /* Renesas Rapid Transfer Notify1 Value */
+    { DB_TYPE_128BIT_UUID,
+        sizeof(rtps_notify_char_val1), sizeof(rtps_notify_char_val1),
TASK_ATTID(TASK_RBLE, RRTS_IDX_NOTIFY_VAL1),
(RBLE_GATT_PERM_NI|RBLE_GATT_PERM_NOTIFY_COMP_EN), (void
*)&rtps_notify_char_val_elmt1 },
+    /* Renesas Rapid Transfer Notify2 Char */
+    { RBLE_DECL_CHARACTERISTIC,
+        sizeof(rtps_notify_char2), sizeof(rtps_notify_char2),
TASK_ATTID(TASK_RBLE, RRTS_IDX_NOTIFY_CHAR2), RBLE_GATT_PERM_RD, (void
*)&rtps_notify_char2 },
+    /* Renesas Rapid Transfer Notify2 Value */
+    { DB_TYPE_128BIT_UUID,
+        sizeof(rtps_notify_char_val2), sizeof(rtps_notify_char_val2),
TASK_ATTID(TASK_RBLE, RRTS_IDX_NOTIFY_VAL2),
(RBLE_GATT_PERM_NI|RBLE_GATT_PERM_NOTIFY_COMP_EN), (void
*)&rtps_notify_char_val_elmt2 },
+    /* Renesas Rapid Transfer Notify3 Char */
+    { RBLE_DECL_CHARACTERISTIC,
+        sizeof(rtps_notify_char3), sizeof(rtps_notify_char3),
TASK_ATTID(TASK_RBLE, RRTS_IDX_NOTIFY_CHAR3), RBLE_GATT_PERM_RD, (void
*)&rtps_notify_char3 },
+    /* Renesas Rapid Transfer Notify3 Value */
+    { DB_TYPE_128BIT_UUID,
+        sizeof(rtps_notify_char_val3), sizeof(rtps_notify_char_val3),
TASK_ATTID(TASK_RBLE, RRTS_IDX_NOTIFY_VAL3),
(RBLE_GATT_PERM_NI|RBLE_GATT_PERM_NOTIFY_COMP_EN), (void
*)&rtps_notify_char_val_elmt3 },
+    /* Renesas Rapid Transfer Notify4 Char */
+    { RBLE_DECL_CHARACTERISTIC,
+        sizeof(rtps_notify_char4), sizeof(rtps_notify_char4),
TASK_ATTID(TASK_RBLE, RRTS_IDX_NOTIFY_CHAR4), RBLE_GATT_PERM_RD, (void
*)&rtps_notify_char4 },
+    /* Renesas Rapid Transfer Notify4 Value */
+    { DB_TYPE_128BIT_UUID,
+        sizeof(rtps_notify_char_val4), sizeof(rtps_notify_char_val4),
TASK_ATTID(TASK_RBLE, RRTS_IDX_NOTIFY_VAL4),
(RBLE_GATT_PERM_NI|RBLE_GATT_PERM_NOTIFY_COMP_EN), (void
*)&rtps_notify_char_val_elmt4 },
+    /* Renesas Rapid Transfer Notify Cfg Value */
+    { RBLE_DESC_CLIENT_CHAR_CONF,
+        sizeof(rtps_notify_en), sizeof(rtps_notify_en), TASK_ATTID(TASK_RBLE,
RRTS_IDX_NOTIFY_CFG), (RBLE_GATT_PERM_RD|RBLE_GATT_PERM_WR), (void
*)&rtps_notify_en },
+    /* Renesas Rapid Transfer Indicate Char */
+    { RBLE_DECL_CHARACTERISTIC,
+        sizeof(rtps_ind_char), sizeof(rtps_ind_char), TASK_ATTID(TASK_RBLE,
RRTS_IDX_IND_CHAR), RBLE_GATT_PERM_RD, (void *)&rtps_ind_char },
+    /* Renesas Rapid Transfer Indicate Value */
+    { DB_TYPE_128BIT_UUID,
+        sizeof(rtps_ind_char_val), sizeof(rtps_ind_char_val),
TASK_ATTID(TASK_RBLE, RRTS_IDX_IND_VAL), RBLE_GATT_PERM_NI, (void
*)&rtps_ind_char_val_elmt },
+    /* Renesas Rapid Transfer Indicate Cfg Value */
+    { RBLE_DESC_CLIENT_CHAR_CONF,
+        sizeof(rtps_ind_en), sizeof(rtps_ind_en), TASK_ATTID(TASK_RBLE,
RRTS_IDX_IND_CFG), (RBLE_GATT_PERM_RD|RBLE_GATT_PERM_WR), (void
*)&rtps_ind_en },
+#endif /* (PRF_SEL_RTPS) */
+
```

(5) prf_config.h

(a) Added to line 33 (lines 33 of the sample code)

Adds header file to include.

```
+#include "rble_api_rrtp.h"
```

(b) Added to line 502 (lines 503 to 518 of the sample code)

Adds index definition of RRTP.

```
- SAMS_IDX_LED_CONTROL_VAL
```

```
 +    SAMS_IDX_LED_CONTROL_VAL,
 +
 +    /* Renesas Rapid Transfer Service */
 +    RRTS_IDX_SVC,
 +    RRTS_IDX_NOTIFY_CHAR1,
 +    RRTS_IDX_NOTIFY_VAL1,
 +    RRTS_IDX_NOTIFY_CHAR2,
 +    RRTS_IDX_NOTIFY_VAL2,
 +    RRTS_IDX_NOTIFY_CHAR3,
 +    RRTS_IDX_NOTIFY_VAL3,
 +    RRTS_IDX_NOTIFY_CHAR4,
 +    RRTS_IDX_NOTIFY_VAL4,
 +    RRTS_IDX_NOTIFY_CFG,
 +    RRTS_IDX_IND_CHAR,
 +    RRTS_IDX_IND_VAL,
 +    RRTS_IDX_IND_CFG,
```

(6) prf_sel.h

(a) Added to line 38 (lines 38 to 41 of the sample code)

Disables unnecessary profiles.

```
 -#define PRF_SEL_PXPM    1   /* Proximity Profile Monitor role */
 -#define PRF_SEL_PXPR    1   /* Proximity Profile Reporter role */
 -#define PRF_SEL_FMPL    1   /* Find Me Profile Locator role */
 -#define PRF_SEL_FMPT    1   /* Find Me Profile Target role */
 +#define PRF_SEL_PXPM    0   /* Proximity Profile Monitor role */
 +#define PRF_SEL_PXPR    0   /* Proximity Profile Reporter role */
 +#define PRF_SEL_FMPL    0   /* Find Me Profile Locator role */
 +#define PRF_SEL_FMPT    0   /* Find Me Profile Target role */
```

(b) Added to line 61 (lines 61 to 62 of the sample code)

Disables unnecessary profiles.

```
 -#define PRF_SEL_ANPC    1   /* Alert Notification Profile Client role */
 -#define PRF_SEL_ANPS    1   /* Alert Notification Profile Server role */
 +#define PRF_SEL_ANPC    0   /* Alert Notification Profile Client role */
 +#define PRF_SEL_ANPS    0   /* Alert Notification Profile Server role */
```

(c) Added to line 102 (lines 103 to 106 of the sample code)

Activates RRTP (Server roll).

```
 +/* Renesas original custom profile selection */
 +#define PRF_SEL_RTPC    0   /* Renesas Rapid Transfer Profile Client role */
 +#define PRF_SEL_RTPS    1   /* Renesas Rapid Transfer Profile Server role */
 +
```

2.4.3 Project Files

There are multiple environment projects in BLE software. The Peripheral device in this application note use projects in the following layer.

For the layer of WRORKSPACE Folder, refer to “2.3.1 Project Layer”.

```
Project_Source (WORKSPACE Folder)
      └renesas
           └tools
                └project
                     └CS_CCRL
                          └BLE_Embedded (CS+ CC-RL Project: BLE_Embedded.mtpj)
                               └rBLE_Emb
                                    └DefaultBuild (Output File: ROM)
```

Figure 2-6 Project to Use

2.4.4 Build Settings

Changes the subproject build tool properties.

(1) Macro definition

The following lists the definition macros for Compile Option tab. Do not change or delete the definition macros listed below.

Table 2-4 Macro Definition to Change

Default Macro Name	Macro Name after Change	Contents of Change
CFG_FULLEMB	CFG_FULLEMB	
CFG_CON=4	CFG_CON=4	[Note 1]
CFG_EXMEM_NOT_PRESENT	CFG_EXMEM_NOT_PRESENT	
CFG_BLECORE_10	noCFG_BLECORE_10	Changed (Disabled)
CFG_PROFEMB	noCFG_PROFEMB	Changed (Disabled)
CFG_SECURITY_ON	noCFG_SECURITY_ON	Changed (Disabled)
CFG_RBLE	noCFG_RBLE	Changed (Disabled)
CFG_USE_EEL	CFG_USE_EEL	
CFG_FW_NAK	noCFG_FW_NAK	Changed (Disabled)
CONFIG_EMBEDDED	CONFIG_EMBEDDED	
_USE_CCRL_RL78	_USE_CCRL_RL78	
CFG_SAMPLE	noCFG_SAMPLE	Changed (Disabled)
noUSE_SAMPLE_PROFILE	noUSE_SAMPLE_PROFILE	
noCFG_USE_PEAK	CFG_USE_PEAK	Changed (Enabled)
noUSE_FW_UPDATE_PROFILE	noUSE_FW_UPDATE_PROFILE	
CLK_HOCO_8MHZ	CLK_HOCO_32MHZ	Changed (Changed Clock)
CLK_SUB_XT1	CLK_SUB_XT1	
noCFG_PKTMON	noCFG_PKTMON	

Note 1: When operating as Slave only in "6.1.1 Maximum Number of Simultaneous Connections" of 'Bluetooth Low Energy Protocol Stack User's Manual' (https://www.renesas.com/us/en/doc/products/mpumcu/doc/rl78/r01uw0095ej0122-g1dum.pdf), it may be set to "1".
But when communicating using More Data, set "CFG_CON" to a value equal to or greater than the number of More Data.

(2) User Option Byte

Changes the operating clock. Be sure to synchronize with the clock setting in 'Table 2-4 Macro Definition to Change'.

Table 2-5 User Option Byte to Change

User Option Byte	Settings before Change	Settings after Change
Address 0x000C2	EFFFAA (8MHz)	EFFFE8 (32MHz)

2.5 Central Device

The Central device searches the BD Address of the Peripheral device and automatically connects it, and then displays the data sent from the Peripheral device and the throughput value per second through the terminal software screen.

2.5.1 List of Changed files

The following shows the differences from the base software of the BLE software of the Central device.

Table 2-6 List of Changed files

Folder	File name	treatment	Remarks
r_BLE/src/include	rble_api_rrtp.h	Added	Definitions of constants and structures used by RRTP
r_BLE/src/sample_app	rble_app_rrtpc.c	Added	RRTP client API and callback function
r_BLE/src/sample_app	rble_app_rrtpc.h	Added	Header file for rble_app_rrtpc.c
r_BLE/src/sample_profile	rtpc.c	Added	Service processing functions of RRTP client
bleip/src/common	co.bt.h	Modified	
r_BLE/src/include	rble_app.h	Modified	
r_BLE/src/sample_app	Console.c	Modified	
r_BLE/src/sample_app	Console.h	Modified	
r_BLE/src/sample_app	menu_sel.c	Modified	
r_BLE/src/sample_app	rble_sample_app.c	Modified	
r_BLE/src/sample_app	rble_sample_app_gap_sm_gatt.c	Modified	
renesass/src/arch/rl78	arch_main.c	Modified	
renesass/src/arch/rl78	config.h	Modified	
renesass/src/arch/rl78	main.c	Modified	
renesass/src/arch/rl78	prf_sel.h	Modified	
renesass/src/driver/dataflash	dataflash.c	Modified	
renesass/src/driver/dataflash	dataflash.h	Modified	
renesass/src/driver/uart	uart.c	Modified	

2.5.2 Change Difference Details

The following shows the difference information from the BLE software. The line number mainly indicates the line position after the change.

The "-" at the beginning of the line indicates the deletion of the line and the "+" at the beginning of the line indicates the addition of the line.

(1) co_bt.h

(a) Added to line 42 (lines 42 of the sample code)

Adds auto connection definition.

```
+ #define __AUTO_CONNECT_DEMO__
```

(b) Added to line 681 (lines 682 to 688 of the sample code)

Adds a type to store Connection Interval at automatic connection.

```
+ #ifdef __AUTO_CONNECT_DEMO__
+ /* connection interval variable */
+ struct con_intval
+ {
+  uint8_t      val;
+ };
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(2) rble_app.h

(a) Added to line 18 (lines 18 to 19 of the sample code)

Adds branch macro definition for connection processing.

```
+ #define __THROUGHPUT_TEST__
+
```

(b) Added to line 22 (lines 24 to 32 of the sample code)

Adds header files to include for connection processing.

```
+ #ifdef __THROUGHPUT_TEST__
+ #if !defined(_USE_RWBLE_SOURCE)
+ #include "arch.h"
+ #include "rwke_api.h"
+ #else /* !defined(_USE_RWBLE_SOURCE) */
+ #include  "ke_task.h"
+ #endif
+ #endif
+
```

(c) Added to line 34 (lines 45 to 52 of the sample code)

Adds macro definition for connection processing.

```
+ #ifdef __THROUGHPUT_TEST__
+ /* Task Infomation */
+ #define DEMO_STATE_MAX    1   /* Max State Num */
+ #define DEMO_IDX_MAX      1   /* Max ID Num */
+
+ #define DEMO_DATA_SEND    1   /* Task API ID */
+ #endif
+
```

(d) Added to line 81 (lines 100 to 109 of the sample code)

Adds extern declaration for connection processing.

```
+
+ #ifdef __THROUGHPUT_TEST__
+ /* Status Handler */
+ extern const struct ke_state_handler Demo_state_handler[ DEMO_STATE_MAX ];
+ /* Default Handler */
+ extern const struct ke_state_handler Demo_default_handler;
+ /* Status */
+ extern ke_state_t Demo_State[ DEMO_IDX_MAX ];
+ #endif
+
```

(3) Console.c

(a) Added to line 18 (lines 19 to 19 of the sample code)

Adds branch macro definition for connection processing.

```
+ #define  __THROUGHPUT_TEST__
+
```

(b) Added to line 41 (lines 43 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(c) Added to line 45 (lines 47 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif
+ /*#endif*/
```

(d) Added to line 50 (lines 52 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(e) Added to line 54 (lines 56 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif
+ /*#endif*/
```

(f) Added to line 82 (lines 84 of the sample code)

Commenst out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(g) Added to line 84 (lines 86 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif
+ /*#endif*/
```

(h) Added to line 244 (lines 246 to 249 of the sample code)

Adds extern declaration for connection processing.

```
+ #ifdef __THROUGHPUT_TEST__
+ extern uint8_t     RBLE_Test_Start_Flg;
+ #endif /* __THROUGHPUT_TEST__ */
+
```

(i) Added to line 608 (lines 614 to 618 of the sample code)

Adds conditional branch processing at connection processing.

```
+
+ #ifdef __THROUGHPUT_TEST__
+ if ( false == RBLE_Test_Start_Flg ) {
```

```
+ #endif /* __THROUGHPUT_TEST__ */
+
```

(j) Added to line630 (lines 641 to 643 of the sample code)

Adds conditional branch end at connection processing.

```
+ #ifdef __THROUGHPUT_TEST__
+ }
+ #endif /* __THROUGHPUT_TEST__ */
```

(k) Added to line 674 (lines 688 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(l) Added to line 788 (lines 802 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif /* USE_CUSTOM_DEMO */
+ /*#endif*/ /* USE_CUSTOM_DEMO */
```

(4) Console.h

(a) Added to line 83 (lines 83 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(b) Added to line 89 (lines 89 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif
+ /*#endif*/
```

(c) Added to line 59 (lines 59 to 62 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(d) Added to line 105 (lines 105 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif
+ /*#endif*/
```

(5) menu_sel.c

(a) Added to line 206 (lines 206 of the sample code)

Adds branch macro processing not to process at automatic connection.

```
+ #ifndef __AUTO_CONNECT_DEMO__
```

(b) Added to line 209 (lines 210 of the sample code)

Adds branch macro end.

```
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(6) rble_sample_app.c

(a) Added to line 22 (lines 22 to 24 of the sample code)

Adds conditional branch processing at connection processing.

```
+ #define __THROUGHPUT_TEST__
+ #define __AUTO_CONNECT_DEMO__
```

```
+
```

(b) Added to line 37 (lines 40 to 44 of the sample code)

Comments out the inclusion of unnecessary headers and adds the required headers

```
- #include    "push_sw.h"
+ /* #include    "push_sw.h" */
+ #endif
+ #ifdef __AUTO_CONNECT_DEMO__
+ #include    "co_bt.h"
+ #include    "dataflash.h"
```

(c) Added to line 61 (lines 68 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(d) Added to line 64 (lines 71 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif
+ /*#endif*/
```

(e) Added to line 70 (lines 77 to 80 of the sample code)

Adds extern declaration of function used at automatic connection

```
+ #ifdef  __THROUGHPUT_TEST__
+ extern BOOL RBLE_Test_Select( void );
+ #endif  /* __THROUGHPUT_TEST__ */
+
```

(f) Added to line 97 (lines 108 to 119 of the sample code)

Adds extern declaration of variable used in connection processing.

```
+ #ifdef  __THROUGHPUT_TEST__
+ #define             DEVICE_SEARCH_MAX              10
+
+ extern RBLE_BD_ADDR Remote_Device;
+ extern RBLE_BD_ADDR        Device_Search_Result[ DEVICE_SEARCH_MAX ];
+ extern uint16_t            Device_Search_Cnt;
+ #endif
+
+ #ifdef __AUTO_CONNECT_DEMO__
+ extern struct bd_addr remote_bda_addr;     /* Remote Device Address */
+ extern struct con_intval df_intval;        /* connection interval */
+ #else /* __AUTO_CONNECT_DEMO__ */
```

(g) Added to line 165 (lines 188 to 190 of the sample code)

Adds connection processing settings.

```
+ #ifdef  __THROUGHPUT_TEST__
+     {  6,   RBLE_Test_Select,     NULL,
"6.Test Case Select\n", },
+ #endif  /* __THROUGHPUT_TEST__ */
```

(h) Added to line 173 (lines 199 of the sample code)

Add branch macro end of automatic connection.

```
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(i) Added to line 215 (lines 242 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(j) Added to line 218 (lines 245 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif
+ /*#endif*/
```

(k) Added to line 256 (lines 283 to 286 of the sample code)

Adds data setting process for connection processing.

```
+ #ifdef  __THROUGHPUT_TEST__
+         Device_Search_Result[ 0 ] = Remote_Device;
+         Device_Search_Cnt = 1;
+ #endif
```

(l) Added to line 265 (lines 296 of the sample code)

Comments out unnecessary function calls.

```
-         push_sw2_start( &sw_int );
+         /* push_sw2_start( &sw_int ); */
```

(m) Added to line 267 (lines 298 to 304 of the sample code)

Adds get function call and serial output processing from data flash for automatic connection.

```
+ #ifdef __AUTO_CONNECT_DEMO__
+         flash_get_remote_bda();
+  printf("Dataflash 0xF2800 set Remote BLE Addr
= %02X:%02X:%02X:%02X:%02X:%02X\n",
+     remote_bda_addr.addr[0],remote_bda_addr.addr[1],remote_bda_addr.addr[2],
+
 remote_bda_addr.addr[3],remote_bda_addr.addr[4],remote_bda_addr.addr[5]);
+         flash_get_intval();
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(n) Added to line 287 (lines 325 of the sample code)

Adds branch macro processing of automatic connection.

```
+ #ifndef __AUTO_CONNECT_DEMO__
```

(o) Added to line 288 (lines 327 of the sample code)

Adds branch macro end.

```
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(p) Added to line 400 (lines 440 of the sample code)

Adds branch macro processing of automatic connection.

```
+ #ifndef __AUTO_CONNECT_DEMO__
```

(q) Added to line 401 (lines 442 of the sample code)

Adds branch macro end.

```
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(7) rble_sample_app_gap_sm_gatt.c

(a) Added to line 22 (lines 22 to 24 of the sample code)

Adds macro definition for connection processing.

```
+ #define __THROUGHPUT_TEST__
+ #define __AUTO_CONNECT_DEMO__
+
```

(b) Added to line 35 (lines 38 to 41 of the sample code)

Add header files to include for automatic connection.

```
+ #ifdef __AUTO_CONNECT_DEMO__
+ #include    "co_bt.h"
+ #include "rble_app_rrtpc.h"
+ #endif
```

(c) Added to line 102 (lines 109 of the sample code)

Removes 'static' for reference by other sources.

```
- static BOOL RBLE_GAP_Device_Search_Test( void );
/* A GAP_Device_Search command is executed. */
+        BOOL RBLE_GAP_Device_Search_Test( void );
/* A GAP_Device_Search command is executed. */
```

(d) Added to line 146 (lines 153 to 155 of the sample code)

Adds extern declaration to use function of other sources in connection processing.

```
+ extern void send_data(void);
+ #ifdef  __THROUGHPUT_TEST__
+ extern BOOL RBLE_SCP_Client_Enable_Test(void);
```

(e) Added to line 147 (lines 157 to 163 of the sample code)

Adds extern declaration to use function of other sources in connection processing.

```
+ #endif
+
+ #ifdef  __THROUGHPUT_TEST__
+ extern void RBLEL_Throughput_Disp( void );
+ #endif  /* __THROUGHPUT_TEST__ */
+
+ extern BOOL RBLE_SCP_Client_Write_Char_Test( void );
/* A SCP_Client_Write_Char command is executed. */
```

(f) Added to line 354 (lines 374 to 405 of the sample code)

Add a declaration to use in connection processing.

```
+ #ifdef  __THROUGHPUT_TEST__
+ #define             DEVICE_SEARCH_MAX               20
+
+ extern RBLE_BD_ADDR        Device_Search_Result[];
+ extern uint16_t            Device_Search_Cnt;
+ typedef struct {
+     BOOL        Notify_en;
+     BOOL        Indicate_en;
+     BOOL        Timer_en;
+     uint16_t    Timer_interval;
+     uint8_t     Notify_len;
+     uint8_t     Indicate_len;
+ } RBLE_SCP_SAMPLE_INFO;
+ //extern RBLE_SCP_SAMPLE_INFO    scp_sample_info;
+ extern RBLE_RRTP_INFO rrtp_info;
+ extern uint8_t     RBLE_Test_Data;
+ extern int_t       RBLE_Test_Type;
+ extern uint8_t     RBLE_Test_Start_Flg;
+ extern uint8_t send_packet_start;
+ extern uint32_t packet_count;
+ #endif  /* __THROUGHPUT_TEST__ */
+
+ #ifdef __AUTO_CONNECT_DEMO__
+ uint8_t d_serch_1st_flg = 0;
+ uint8_t d_search_remote_device = 0;
+ extern struct bd_addr remote_bda_addr;     /* Remote Device Address */
+ extern struct con_intval df_intval;        /* connection interval */
+ #endif /* __AUTO_CONNECT_DEMO__ */
+ #ifdef __USE_REPEAT_CONNECTION__
+ extern uint8_t g_hdl_set_flg;
+ #endif /* __USE_REPEAT_CONNECTION__ */
```

```
+
```

(g) Added to line 434 (lines 438 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(h) Added to line 435 (lines 484 to 490 of the sample code)

Adds branch processing of connection processing.

```
+ #ifdef  __THROUGHPUT_TEST__
+             if ( 0 == RBLE_Test_Type ) {
+             RBLE_GAP_Broadcast_Enable_Test();
+             } else if ( (1 == RBLE_Test_Type) || (2 == RBLE_Test_Type) ) {
+                 RBLE_GAP_Create_Connection_Test();
+             }
+ #else
```

(i) Added to line 436 (lines 492 of the sample code)

Adds branch macro end.

```
+ #endif  /* __THROUGHPUT_TEST__ */
```

(j) Added to line 436 (lines 493 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif
+ /*#endif*/
```

(k) Added to line 437 (lines 494 to 499 of the sample code)

Adds processing to repeat search to find the Peripheral device at automatic connection.

```
+ #ifdef __AUTO_CONNECT_DEMO__
+             if(d_serch_1st_flg == 0){
+                 d_serch_1st_flg = 1;
+                 RBLE_GAP_Device_Search_Test();
+       }
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(l) Added to line 501 (lines 564 of the sample code)

Adds branch macro processing not to process by automatic connection.

```
+ #ifndef __AUTO_CONNECT_DEMO__
```

(m) Added to line 502 (lines 566 to 599 of the sample code)

Adds processing to repeat search to find Peripheral device at automatic connection.

```
+ #endif /* __AUTO_CONNECT_DEMO__ */
+ #ifdef  __THROUGHPUT_TEST__
+ {
+             uint16_t        i;
+
+             for ( i = 0;i < Device_Search_Cnt;i++ ) {
+ #ifdef __AUTO_CONNECT_DEMO__
+                 if((Device_Search_Result[ i ].addr[5] ==
remote_bda_addr.addr[0]) &&
+                    (Device_Search_Result[ i ].addr[4] ==
remote_bda_addr.addr[1]) &&
+                    (Device_Search_Result[ i ].addr[3] ==
remote_bda_addr.addr[2]) &&
+                    (Device_Search_Result[ i ].addr[2] ==
remote_bda_addr.addr[3]) &&
+                    (Device_Search_Result[ i ].addr[1] ==
remote_bda_addr.addr[4]) &&
+                     (Device_Search_Result[ i ].addr[0] ==
remote_bda_addr.addr[5])) {
+                         d_search_remote_device = 1;
+                         Remote_Device = Device_Search_Result[ i ];
+             RBLE_Test_Type = 1;
+  Console_SetTextAttribute( CONSOLE_COLOR );
+  printf( "                             " );
+  Console_SetTextAttribute( COMMAND_COLOR );
+             RBLE_GAP_Reset_Test();
+                  }
+ #else /* __AUTO_CONNECT_DEMO__ */
+                  printf( "%d/%d:", i + 1, Device_Search_Cnt );
+                  BdAddress_Disp( &Device_Search_Result[ i ] );
+ #endif /* __AUTO_CONNECT_DEMO__ */
+              }
+ #ifdef __AUTO_CONNECT_DEMO__
+              if(d_search_remote_device == 0){
+                  Device_Search_Cnt = 1;
+                  RBLE_GAP_Device_Search_Test();
+       }
+ #endif /* __AUTO_CONNECT_DEMO__ */
+ }
+ #endif
```

(n) Added to line 504 (lines 602 to 625 of the sample code)

Adds processing to retrieve search results by automatic connection.

```
+ #ifdef __AUTO_CONNECT_DEMO__
+             Adv_Rep_p = &event->param.dev_search_result.adv_resp;
+              Peer_Addr_Type = Adv_Rep_p->adv_addr_type;
+              /* The last device is saved. */
+              Remote_Device = Adv_Rep_p->adv_addr;
+
+              Add_White_List_Dev_info.dev_addr_type = Peer_Addr_Type;
+              Add_White_List_Dev_info.dev_addr = Remote_Device;
+ #ifdef  __THROUGHPUT_TEST__
+ {
+              uint16_t            i;
+
+              for ( i = 0;i < Device_Search_Cnt && Device_Search_Cnt <
DEVICE_SEARCH_MAX;i++ ) {
+                  if ( 0 == memcmp( &Device_Search_Result[ i ],
&Remote_Device, sizeof( RBLE_BD_ADDR ) ) ) {
+                      break;
+                  }
+              }
+              if ( i == Device_Search_Cnt && DEVICE_SEARCH_MAX !=
Device_Search_Cnt ) {
+                  Device_Search_Result[ i ] = Remote_Device;
+                  Device_Search_Cnt++;
+              }
+ }
+ #endif
+ #else /* __AUTO_CONNECT_DEMO__ */
```

(o) Added to line 513 (lines 635 to 650 of the sample code)

Adds processing to retrieve search results by automatic connection.

```
+ #ifdef  __THROUGHPUT_TEST__
+ {
```

```
+               uint16_t            i;
+
+               for ( i = 0;i < Device_Search_Cnt && Device_Search_Cnt <
DEVICE_SEARCH_MAX;i++ ) {
+                   if ( 0 == memcmp( &Device_Search_Result[ i ],
&Remote_Device, sizeof( RBLE_BD_ADDR ) ) ) {
+                       break;
+                   }
+               }
+               if ( i == Device_Search_Cnt && DEVICE_SEARCH_MAX !=
Device_Search_Cnt ) {
+                   Device_Search_Result[ i ] = Remote_Device;
+                   Device_Search_Cnt++;
+               }
+ }
+ #endif
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(p) Added to line 530 (lines 668 to 669 of the sample code)

Adds processing to serial-output Connection Interval value at connection time.

```
+       /* For Connection Interval */
+       printf("Connection Interval = %f msec, ",(float)(1.25 * Con_Info_p-
>con_interval));
```

(q) Added to line 537 (lines 677 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #ifdef USE_CUSTOM_DEMO
+ /*#ifdef USE_CUSTOM_DEMO*/
```

(r) Added to line 538 (lines 678 to 682 of the sample code)

Adds branch processing of connection processing.

```
+ #ifdef  __THROUGHPUT_TEST__
+               if ( (1 == RBLE_Test_Type) || (2 == RBLE_Test_Type) ) {
+                   RBLE_SCP_Client_Enable_Test();
+               }
+ #else
```

(s) Added to line 539 (lines 684 of the sample code)

Adds branch macro end.

```
+ #endif  /* __THROUGHPUT_TEST__ */
```

(t) Added to line 539 (lines 685 of the sample code)

Comments out branch macro for use in the RRTP sample.

```
- #endif
+ /*#endif*/
```

(u) Added to line 545 (lines 691 to 694 of the sample code)

Adds variable setting of connection processing.

```
+ #ifdef  __THROUGHPUT_TEST__
+               RBLE_Test_Start_Flg = false;
+               send_packet_start = false;
+ #endif  /* __THROUGHPUT_TEST__ */
```

(v) Added to line 547 (lines 697 to 703 of the sample code)

Adds disconnected termination processing and reconnect processing.

```
+ #ifdef __USE_REPEAT_CONNECTION__
+         RBLE_RTP_Client_Disable_Test();
+         d_serch_1st_flg = 0;
+         RBLE_Test_Type = -1;
+         g_hdl_set_flg = 0;
+         RBLE_GAP_Reset_Test();
+ #endif /* __USE_REPEAT_CONNECTION__ */
```

(w) Added to line 572 (lines 729 to 737 of the sample code)

Adds branch processing of connection processing.

```
+ #ifdef  __THROUGHPUT_TEST__
+              if ( (1 == RBLE_Test_Type) || (2 == RBLE_Test_Type) ) {
+                  uint16_t result = 0x0000;
+                  if ( 500 < event-
>param.chg_connect_param_req.conn_param.latency ) {
+                      result = 0x0001;
+                  }
+                  RBLE_GAP_Change_Connection_Param(event-
>param.chg_connect_param_req.conhdl, result, &event-
>param.chg_connect_param_req.conn_param, Role_Status);
+              }
+ #endif  /* __THROUGHPUT_TEST__ */
```

(x) Added to line 577 (lines 743 to 775 of the sample code)

Adds variable setting of connection processing.

```
+ #ifdef  __THROUGHPUT_TEST__
+              if ( 0 == RBLE_Test_Type ) {
+               printf("Server!\n");
+                  RBLE_Test_Data = 0;
+                  RBLE_Test_Start_Flg = true;
+                  if ( TRUE == scp_sample_info.Notify_en ) {
+                      RBLE_Test_Start_Flg = false;
+                      printf("Notify_en!\n");
+                      RBLE_Test_Start_Flg = true;
+
+                      if ( 0 == packet_count ) {
+                        //send_packet_start = true;
+                          Console_Send_Timer_msg( RBLEL_Throughput_Disp );
+                          Console_Set_Timer( 100 ); // 100 * 10ms => 1S
+                      }
+ #ifndef CFG_USE_PEAK
+                      RBLE_Test_Start_Flg = false;
+                      RBLE_Test_Start_Flg = true;
+ #else
+ #ifndef CLK_HOCO_32MHZ
+                      RBLE_Test_Start_Flg = false;
+                      RBLE_Test_Start_Flg = true;
+ #endif
+ #endif
+                  } else if ( TRUE == scp_sample_info.Indicate_en ) {
+                      printf("Indication_en!\n");
+                      send_data();
+                  }
+     else {
+         printf("Error! No Notify & Indication!");
+     }
+              }
+ #endif  /* __THROUGHPUT_TEST__ */
```

(y) Added to line 1058 (lines 1257 of the sample code)

Removes 'static' for reference by other sources.

```
- static BOOL RBLE_GAP_Device_Search_Test( void )
```

```
+ BOOL RBLE_GAP_Device_Search_Test( void )
```

(z) Added to line 1060 (lines 1259 of the sample code)

Adds branch macro processing not to process by automatic connection.

```
+ #ifndef __AUTO_CONNECT_DEMO__
```

(aa) Added to line 1061 (lines 1261 of the sample code)

Adds branch macro end.

```
-
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(bb) Added to line 1074 (lines 1274 of the sample code)

Adds branch macro processing not to process by automatic connection.

```
-
+ #ifndef __AUTO_CONNECT_DEMO__
```

(cc) Added to line 1076 (lines 1276 of the sample code)

Adds branch macro end.

```
-
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(dd) Added to line 1187 (lines 1387 to 1389 of the sample code)

接続処理の変数設定を追加します。

```
+ #ifdef  __THROUGHPUT_TEST__
+     param.peer_addr_type    = RBLE_ADDR_PUBLIC;                 /* Peer
address type */
+ #else
```

(ee) Added to line 1188 (lines 1391 of the sample code)

Adds branch macro end.

```
+ #endif
```

(ff) Added to line 1190 (lines 1394 to 1404 of the sample code)

Adds Connection Interval value setting of automatic connection processing.

```
+ #ifdef  __THROUGHPUT_TEST__
+     // For Connection Interval
+ #ifdef __AUTO_CONNECT_DEMO__
+     param.con_intv_min      = df_intval.val;
+     param.con_intv_max      = df_intval.val;
+ #else /* __AUTO_CONNECT_DEMO__ */
+     // 7.5msec (Minimum value for BLE)
+     param.con_intv_min      = 0x6;                              /* Minimum
of connection interval ( 7.5msec = 0x6 * 1.25msec ) Range:0x0006-0x0c80 */
+     param.con_intv_max      = 0x6;                              /* Maximum
of connection interval ( 7.5msec = 0x6 * 1.25msec ) Range:0x0006-0x0c80 */
+ #endif /* __AUTO_CONNECT_DEMO__ */
+ #else
```

(gg) Added to line 1192 (lines 1407 of the sample code)

Adds branch macro end.

```
+ #endif
```

(8) arch_main.c

(a) Added to line 31 (lines 31 to 32 of the sample code)

Adds macro definition of connection processing.

```
+ #define __THROUGHPUT_TEST__
+
```

(b) Added to line 43 (lines 45 of the sample code)

Comments out the inclusion of unnecessary header.

```
- #include "push_sw.h"
+ /* #include "push_sw.h" */
```

(c) Added to line 251 (lines 253 to 261 of the sample code)

Enables peak handling at connection processing.

```
+ #ifdef __THROUGHPUT_TEST__
+ #ifdef  CLK_HOCO_32MHZ
+ #ifdef CFG_USE_PEAK
+     peak_init( 4 );      /* 4msec before peak */
+ #else  /* noCFG_USE_PEAK */
+     peak_init( 0 );      /* not peak */
+ #endif /* CFG_USE_PEAK */
+ #endif /* CLK_HOCO_32MHZ */
+ #else  /* no__THROUGHPUT_TEST__ */
```

(d) Added to line 255 (lines 266 to 267 of the sample code)

Adds comment and adds branch macro end.

```
- #endif
+ #endif /* CFG_USE_PEAK */
+ #endif /* __THROUGHPUT_TEST__ */
```

(9) config.h

(a) Added to line 22 (lines 22 to 23 of the sample code)

Adds auto connection definition.

```
+ #define __AUTO_CONNECT_DEMO__
+
```

(b) Added to line 81 (lines 83 to 87 of the sample code)

Adds address of data flash area to be referred by automatic connection.

```
+ #ifdef __AUTO_CONNECT_DEMO__
+ #define df_bleinfo_top      (0xF2800) /* for R5F11AGJ(BLOCK262) */
+ #define p_remote_bda_ptr    ((__far const struct
bd_addr*    )(df_bleinfo_top+0))
+ #define p_con_intval_ptr    ((__far const struct
con_intval* )(df_bleinfo_top+6))
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(10) main.c

(a) Added to line 36 (lines 36 to 37 of the sample code)

Adds connection processing definition.

```
+ #define __THROUGHPUT_TEST__
+
```

(b) Added to line 59 (lines 61 to 65 of the sample code)

Adds header including for connection processing.

```
+ #define __THROUGHPUT_TEST__
+ #ifndef CONFIG_MODEM
+ #include "rble_app.h"
+ #endif
+ #endif
```

(c) Added to line 464 (lines 471 of the sample code)

Adds branch macro not to use at connection processing.

```
+ #ifndef __THROUGHPUT_TEST__
```

(d) Added to line 468 (lines 476 of the sample code)

Adds branch macro end.

```
+ #endif
```

(e) Added to line 473 (lines 482 to 491 of the sample code)

Adds extern declaration of variables used in connection processing and adds task process.

```
+ #ifdef __THROUGHPUT_TEST__
+ {
+ extern uint8_t      RBLE_Test_Start_Flg;
+ extern uint8_t      RBLE_Test_Notify_en;
+
+     if ( (true == RBLE_Test_Start_Flg) && (true == RBLE_Test_Notify_en) ) {
+         ke_msg_send_basic( DEMO_DATA_SEND, TASK_USR_0, TASK_RBLE );
+     }
+ }
+ #else
```

(f) Added to line 476 (lines 495 of the sample code)

Adds branch macro end.

```
+ #endif
```

(11) prf_sel.h

(a) Added to line 38 (lines 38 to 41 of the sample code)

Disables unnecessary profiles.

```
- #define PRF_SEL_PXPM    1   /* Proximity Profile Monitor role */
- #define PRF_SEL_PXPR    1   /* Proximity Profile Reporter role */
- #define PRF_SEL_FMPL    1   /* Find Me Profile Locator role */
- #define PRF_SEL_FMPT    1   /* Find Me Profile Target role */
+ #define PRF_SEL_PXPM    0   /* Proximity Profile Monitor role */
+ #define PRF_SEL_PXPR    0   /* Proximity Profile Reporter role */
+ #define PRF_SEL_FMPL    0   /* Find Me Profile Locator role */
+ #define PRF_SEL_FMPT    0   /* Find Me Profile Target role */
```

(b) Added to line 61 (lines 61 to 62 of the sample code)

Disables unnecessary profiles.

```
- #define PRF_SEL_ANPC    1   /* Alert Notification Profile Client role */
- #define PRF_SEL_ANPS    1   /* Alert Notification Profile Server role */
+ #define PRF_SEL_ANPC    0   /* Alert Notification Profile Client role */
+ #define PRF_SEL_ANPS    0   /* Alert Notification Profile Server role */
```

(c) Added to line 102 (lines 102 to 105 of the sample code)

Adds RRTP and enables it.

```
+
+ /* Renesas original custom profile selection */
+ #define PRF_SEL_RTPC    1   /* Renesas Rapid Transfer Profile Client role */
+ #define PRF_SEL_RTPS    0   /* Renesas Rapid Transfer Profile Server role */
```

(12) dataflash.c

(a) Added to line 831 (lines 831 to 872 of the sample code)

Adds processing to read BD Address and Connection Interval value from data flash.

```
+
+ #ifdef __AUTO_CONNECT_DEMO__
+ #define D_SAMPLE_CONNECTION_INTVAL_MAX (40)
+ #define D_SAMPLE_CONNECTION_INTVAL_MIN (6)
+ struct bd_addr remote_bda_addr;
+ struct con_intval df_intval;
+ /**
+
****************************************************************************
+  * @brief   get remote device address from DataFlash
+  *
+  * @param[out] remote_bda_addr      remote device address
+  *
+
****************************************************************************
+  */
+ _DFL_CODE void flash_get_remote_bda(void)
+ {
+     uint8_t ii;
+
+     /* try to get device address from DataFlash */
+     for(ii = 0; ii < BD_ADDR_LEN; ii++)
+     {
+         remote_bda_addr.addr[ii] = p_remote_bda_ptr->addr[ii];
+     }
+ }
+
+ /**
+
****************************************************************************
+  * @brief   get connection interval from DataFlash
+  *
+  * @param[out] df_intval     connection interval
+  *
+
****************************************************************************
+  */
+ _DFL_CODE void flash_get_intval(void)
+ {
+     df_intval.val = p_con_intval_ptr->val;
+     if((df_intval.val < D_SAMPLE_CONNECTION_INTVAL_MIN) || (df_intval.val >
D_SAMPLE_CONNECTION_INTVAL_MAX))
+     {
+         df_intval.val = D_SAMPLE_CONNECTION_INTVAL_MIN;
+     }
+ }
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(13) dataflash.h

(a) Added to line 225 (lines 225 to 247 of the sample code)

Adds prototype declaration of the function added in dataflash.h.

```
+ #ifdef __AUTO_CONNECT_DEMO__
+ /**
+
****************************************************************************
+  * @brief   get remote device address from DataFlash
+  *
+  * @param[out] bda      remote device address
+  *
+
****************************************************************************
```

```
+   */
+ void flash_get_remote_bda(void);
+
+ /**
+
***************************************************************************
+   * @brief   get connection interval from DataFlash
+   *
+   * @param[out] df_intval     connection interval
+   *
+
***************************************************************************
+   */
+ void flash_get_intval(void);
+
+ #endif /* __AUTO_CONNECT_DEMO__ */
```

(14) uart.c

(a) Added to line 131 (lines 131 of the sample code)

Adds 32 MHz operation definition.

```
+ #define UART_VAL_SPS_32MHZ  0x00U
```

(b) Added to line 390 (lines 391 of the sample code)

Change the effective branch processing.

```
- #if (1)
+ #if (0)
```

(c) Added to line 408 (lines 409 to 416 of the sample code)

Changes the communication speed to 1Mbps.

```
-         write_sfr(SPS0L, (uint8_t)((read_sfr(SPS0L) | UART_VAL_SPS_2MHZ)));
+ /*         write_sfr(SPS0L, (uint8_t)((read_sfr(SPS0L) |
UART_VAL_SPS_2MHZ)));*/
          /* baudrate 250000bps(when MCK = 2MHz) */
-         write_sfrp(UART_TXD_SDR, (uint16_t)0x0600U);
+ /*         write_sfrp(UART_TXD_SDR, (uint16_t)0x0600U);*/
-         write_sfrp(UART_RXD_SDR, (uint16_t)0x0600U);
+ /*         write_sfrp(UART_RXD_SDR, (uint16_t)0x0600U);*/
+          /* MCK = fclk/n = 32MHz */
+          write_sfr(SPS0L, (uint8_t)((read_sfr(SPS0L) | UART_VAL_SPS_32MHZ)));
+          write_sfrp(UART_TXD_SDR, (uint16_t)0x1E00U); /* 1/32(1,000,000bps)
*/
+          write_sfrp(UART_RXD_SDR, (uint16_t)0x1E00U); /* 1/32(1,000,000bps)
*/
```

2.5.3 Project Files

There are multiple environment projects in BLE software. The Central device in this application note use projects in the following layer.

For the layer of WRORKSPACE Folder, refer to “2.3.1 Project Layer”.

```
Project_Source (WORKSPACE Folder)
        └renesas
                └tools
                        └project
                                └CS_CCRL
                                        └BLE_Embedded (CS+ CC-RL Project: BLE_Embedded.mtpj)
                                                └rBLE_Emb
                                                        └DefaultBuild (Output File: ROM)
```

Figure 2-7 Project to Use

2.5.4 Build Settings

Changes the subproject build tool properties.

(1) Macro definition

The following lists the definition macros for Compile Option tab. Do not change or delete the definition macros listed below.

Table 2-7 Macro Definition to Change

Default Macro Name	Macro Name after Change	Contents of Change
CFG_FULLEMB	CFG_FULLEMB	
CFG_CON=4	CFG_CON=4	[Note 1]
CFG_EXMEM_NOT_PRESENT	CFG_EXMEM_NOT_PRESENT	
CFG_BLECORE_10	noCFG_BLECORE_10	
CFG_PROFEMB	noCFG_PROFEMB	
CFG_SECURITY_ON	noCFG_SECURITY_ON	
CFG_RBLE	noCFG_RBLE	
CFG_USE_EEL	CFG_USE_EEL	
CFG_FW_NAK	noCFG_FW_NAK	
CONFIG_EMBEDDED	CONFIG_EMBEDDED	
_USE_CCRL_RL78	_USE_CCRL_RL78	
CFG_SAMPLE	CFG_SAMPLE	
noUSE_SAMPLE_PROFILE	noUSE_SAMPLE_PROFILE	
noCFG_USE_PEAK	CFG_USE_PEAK	Changed (Enabled)
noUSE_FW_UPDATE_PROFILE	noUSE_FW_UPDATE_PROFILE	
CLK_HOCO_8MHZ	CLK_HOCO_32MHZ	Changed (Changed Clock)
CLK_SUB_XT1	CLK_SUB_XT1	
noCFG_PKTMON	noCFG_PKTMON	

Note 1: When communicating using More Data, set "CFG_CON" to a value equal to or greater than the number of More Data.

(2) User Option Byte

Table 2-8 User Option Byte to Change

User Option Byte	Settings before Change	Settings after Change
Address 0x000C2	EFFFAA (8MHz)	EFFFE8 (32MHz)

2.6 Simple Demo for Operation Check

With the configuration shown in “Figure 1-1 Overview of Sample Program Operating Environment”, you can run a simple demo to check the operation.

2.6.1 Simple Demo System Configuration for Operation Check

Using RL78/G1D evaluation board as a peripheral device, creates packets to be transmitted from the generated random data.

The measurement data is received using L78/G1D evaluation board as the Central device of the opposing device. Received data and communication status information such as reception throughput can be displayed on PC terminal software via built-in UART to USB conversion device.

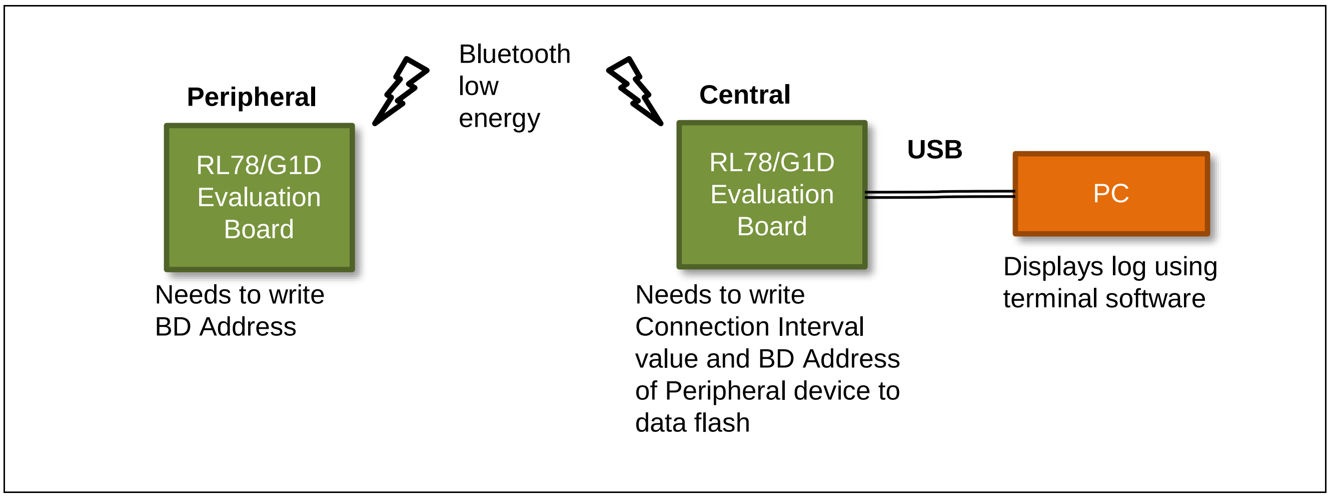

Peripheral
RL78/G1D Evaluation Board
Needs to write BD Address
Bluetooth low energy
Central
RL78/G1D Evaluation Board
Needs to write Connection Interval value and BD Address of Peripheral device to data flash
USB
PC
Displays log using terminal software

Figure 2-8 Simple Demo System Configuration for Operation Check

The format of the 20-byte user data (packet) transmitted in this demo is shown below.

Data content is random and has no meaning.

Table 2-9 Contents of Transmission Data

Data	Size (byte)	Descriptions
SN	**1**	Serial number: 0x00 to 0xFF [Note 1] Incremented each time a packet is sent. After 0xFF, it returns to 0x00.
DATA	**19**	The content is random.

Note 1: Depending on the conditions such as radio waves between Peripheral device and Central device, packet loss and exchange of reception order may occur.

2.6.2 Preparation of Peripheral Device (RL78/G1D Evaluation Board)

The contents of preparation for the simple demo are shown below.

(1) Hardware Configuration

Use an RL78/G1D evaluation board alone.

As shown in Table 2-10, set the slide switches SW7, SW8 and SW11 on the RL78/G1D evaluation board to operate with USB power supply from the PC. For details on slide switch settings, refer to the RL78/G1D evaluation board user's manual.

Table 2-10 Setting of Slide Switches SW7, SW8 and SW11 on RL78/G1D Evaluation Board

Slide Switch	Setting
SW7	"2-3 connection" side (all Pin No.3 side)
SW8	"2-3 connection" side (all Pin No.3 side)
SW11	"2-3 connection" side (all Pin No.3 side)

(2) Firmware Write

Open BLE_Embedded.mtpj (refer to "2.4.3 Project Files") in the development environment CS +.

After launching the firmware project, select the menu "Debug"-> "Rebuild & Debug Tool" and download the firmware to the evaluation board.

The firmware is stored in the following folder in the MOT file format. Therefore, it is possible to write this file to RL78/G1D code flash using Renesas Flash Programmer without using a development environment.

Table 2-11 Firmware Storage Location

```
├─ROM_Files
│  └─RL78_G1D_peripheral
│  │  └─rBLE_Emb_CCRL.mot
```

Note: Do not delete block #255 when rewriting the code flash of RL78/G1D. If BD Address has been written, BD Address will be lost if deleted.

(3) BD Address Write

Since a RL78/G1D evaluation board is equipped with a single RL78/G1D, BD Address is not stored. Pay attention to the endianness and write BD Address to RL78/G1D. After writing, it is recommended to scan using SmartPhone etc. and check if the set BD Address is displayed.

For details on BD Address. refer to ‘Bluetooth Low Energy Protocol Stack User’s Manual (R01UW0095EJ)’.

2.6.3 Preparation of Central Device (RL78/G1D Evaluation Board)

The contents of preparation for the simple demo are shown below.

(1) Hardware Configuration

Connect an RL78/G1D evaluation board and a PC that displays the communication results using USB.

As shown in Table 2-12, set the slide switches SW7, SW8 and SW11 on the RL78/G1D evaluation board to operate with USB power supply from the PC. For details on slide switch settings, refer to the RL78/G1D evaluation board user's manual.

Table 2-12 Setting of Slide Switches SW7, SW8 and SW11 on RL78/G1D Evaluation Board

Slide Switch	Setting
SW7	"2-3 connection" side (all Pin No.3 side)
SW8	"2-3 connection" side (all Pin No.3 side)
SW11	"2-3 connection" side (all Pin No.3 side)

(2) Preparation of PC Terminal Software

Use Tera Term as terminal software for log output. The following shows the settings for "Terminal…" and "Serial port…" in the "Setup" menu.

Table 2-13 Setting for "Serial port…" in "Setup" Menu

"Tera Term: Serial port setup" Screen	Setting Value
Speed	1,000,000 (Enter value directly)
Data	8 bit
Parity	none
Stop bits	1 bit
Flow control	none
Transit delay (msec/char)	1
Transit delay (msec/line)	100

Table 2-14 Setting for " Terminal…" in "Setup" Menu

"Tera Term: terminal setup" Screen	Setting Value
New-line Receive	LF
New-line Transmit	CR

You can put a timestamp on the output log using the function of Tera Term on the serial console. To output a timestamp, enable the "Timestamp" checkbox in the "Options" item in the "Setup" menu -> "Additional setting…" -> "Log" tab. It has been confirmed in Tera Term Version 4.86.

Table 2-15 Setting for "Additional setting" in "Setup" Menu

"Tera Term: Additional setting" Screen "Log" Tab	Setting Value
Log option: Timestamp	✓ (Valid)

The following shows how to set log storage.

Table 2-16 Setting for "Log…" in "File" Menu

"Tera Term: Log" Screen	Setting Value
Filename	Log output filename

(3) Writing to Code Flash

Rewrite RL78/G1D code flash using Renesas Flash Programmer (RFP). The firmware for demo only in MOT file format is provided. It has been stored in the following folder.

Table 2-17 Firmware Storage Location

```
├─ROM_Files
│  ├─RL78_G1D_central
│  │  └─rBLE_Emb_CCRL.mot
```

Note: Do not delete a block #255 when rewriting code flash of RL78/G1D. If the BD Address is written, it will be lost by deleting.

(4) Writing to Data Flash

Use RFP to write specific information for demo. Use unique code of RFP to write the specific information for demo. The following shows the data structure of specific information for demo and how to create the unique code.

The following shows how to write only the data flash when the code flash has been written.

(a) Data Configuration of Specific Information for Demo

Data flash block #6 (address: 0xF2800) needs to store the BD Address of the Peripheral device and Connection Interval settings.

The BD address of the Peripheral device is used to identify the connection target when the Central device searches for the Peripheral device.

Table 2-18 Data Flash Block #6 of Central Device

Offset Address	Size (Bytes)	Description
0x00 to 0x05	**6**	BD address of the Peripheral device for Connection Target
0x06	**1**	Connection Interval value (0x06 to 0x28) [Note 1] Connection Interval time (milliseconds) is 1.25 times the setting value.

Note 1: The setting value is valid from 0x06 to 0x28 (7.5 to 50 milliseconds). If it is out of the setting range, it operates in 7.5 milliseconds. Using the RRTP, it recommends setting 7.5 milliseconds.

(b) Preparation of Unique Code

Edit the unique code file (sample_central.ruc) with a text editor, and set the BD Address of the Peripheral device to be connected and Connection Interval value (recommended value: 0x06 (7.5 milliseconds)) to connect to.

In the case of the RL78/G1D evaluation board, the “749050-XXXXXX” (XXXXXX is a number) described on the label on the RL78/G1D module (RTK0EN0002C01001BZ) can be used as the BD Address.

```
//Sample unique code file
// Peripheral BD Address(6-byte), connection interval(1-byte)
format hex
address 0xf2800
size 7
index data
000001 749050800c7606
```

BD Address of Peripheral device to be connected

Connection Interval Value

Figure 2-9 Description Example of Unique Code

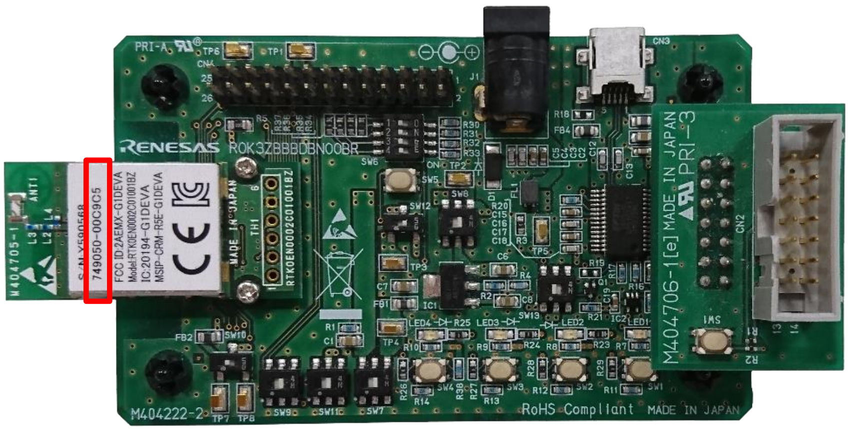
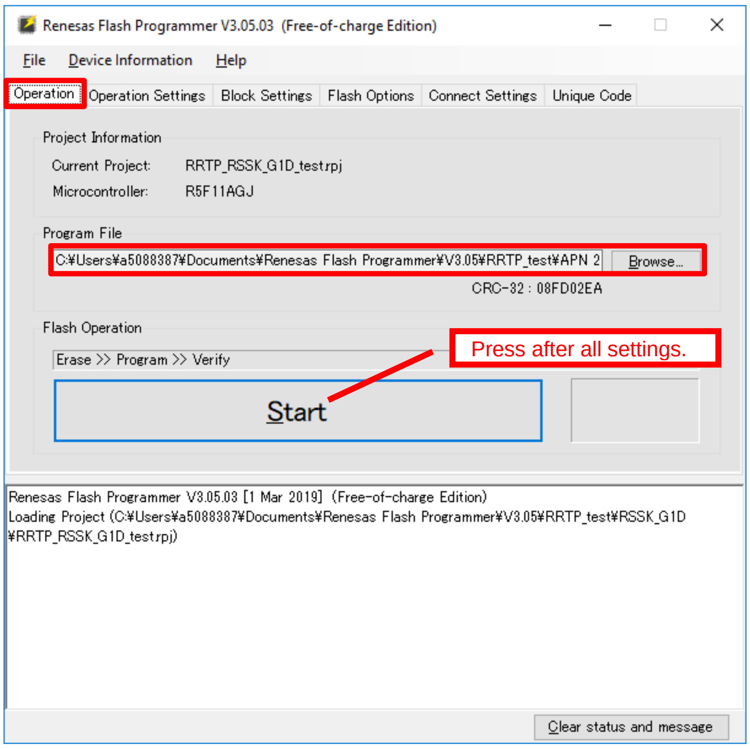


Figure 2-10 BD Address of RL78/G1D Module (RTK0EN0002C01001BZ)

(c) Writing of Unique Code

1. Start up the RFP, and select the firmware dedicated MOT file (rBLE_Emb_CCRL.mot in RL78_G1D_peripheral folder) for demo of the Central device in the "Operation" tab.

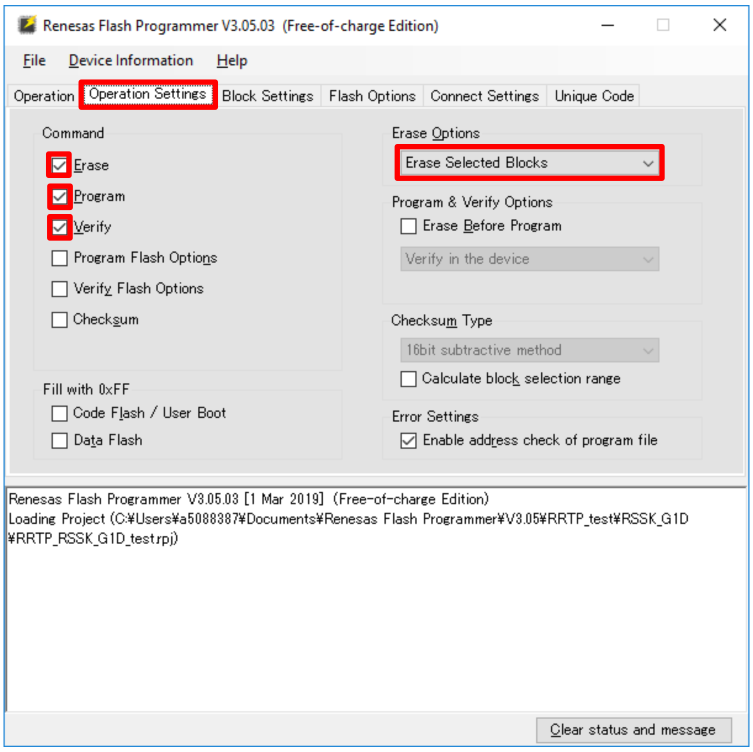


Current Project: RRTP_RSSK_G1D_test.rpj
Microcontroller: R5F11AGJ
C:¥Users¥a5088387¥Documents¥Renesas Flash Programmer¥V3.05¥RRTP_test¥APN 2
CRC-32 : 08FD02EA

```
Renesas Flash Programmer V3.05.03 [1 Mar 2019] (Free-of-charge Edition)
Loading Project (C:¥Users¥a5088387¥Documents¥Renesas Flash Programmer¥V3.05¥RRTP_test¥RSSK_G1D
¥RRTP_RSSK_G1D_test.rpj)
```

Figure 2-11 RFP "Operation" Tab

2. On the "Operation settings" tab, check that "Erase", "Write" and "Verify" are checked in "Command". Also, change "Erase Options" to "Erase Selected Blocks".



Command
- [✓] Erase
- [✓] Program
- [✓] Verify
- [ ] Program Flash Options
- [ ] Verify Flash Options
- [ ] Checksum

Fill with 0xFF
- [ ] Code Flash / User Boot
- [ ] Data Flash

Erase Options: Erase Selected Blocks

Program & Verify Options
- [ ] Erase Before Program
- Verify in the device

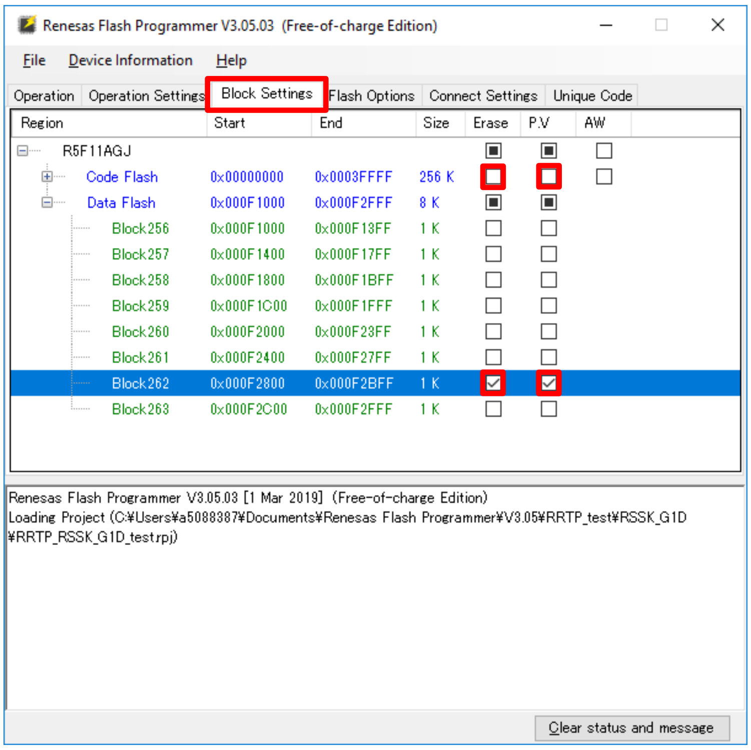
Checksum Type
- 16bit subtractive method
- [ ] Calculate block selection range

Error Settings
- [✓] Enable address check of program file

```
Renesas Flash Programmer V3.05.03 [1 Mar 2019] (Free-of-charge Edition)
Loading Project (C:¥Users¥a5088387¥Documents¥Renesas Flash Programmer¥V3.05¥RRTP_test¥RSSK_G1D
¥RRTP_RSSK_G1D_test.rpj)
```

Figure 2-12 RFP "Operation Setting" Tab

3. On the "Block Setting" tab, check only the check box of data flash #6 (Address 0x000F2800) in order to exclude code flash from operation.



Renesas Flash Programmer V3.05.03 (Free-of-charge Edition)

File Device Information Help

Operation | Operation Settings | Block Settings | Flash Options | Connect Settings | Unique Code

Region	Start	End	Size	Erase	P.V	AW
R5F11AGJ				▣	▣	☐
Code Flash	0x00000000	0x0003FFFF	256 K	☐	☐	☐
Data Flash	0x000F1000	0x000F2FFF	8 K	▣	▣	
Block256	0x000F1000	0x000F13FF	1 K	☐	☐	
Block257	0x000F1400	0x000F17FF	1 K	☐	☐	
Block258	0x000F1800	0x000F1BFF	1 K	☐	☐	
Block259	0x000F1C00	0x000F1FFF	1 K	☐	☐	
Block260	0x000F2000	0x000F23FF	1 K	☐	☐	
Block261	0x000F2400	0x000F27FF	1 K	☐	☐	
Block262	0x000F2800	0x000F2BFF	1 K	✓	✓	
Block263	0x000F2C00	0x000F2FFF	1 K	☐	☐	

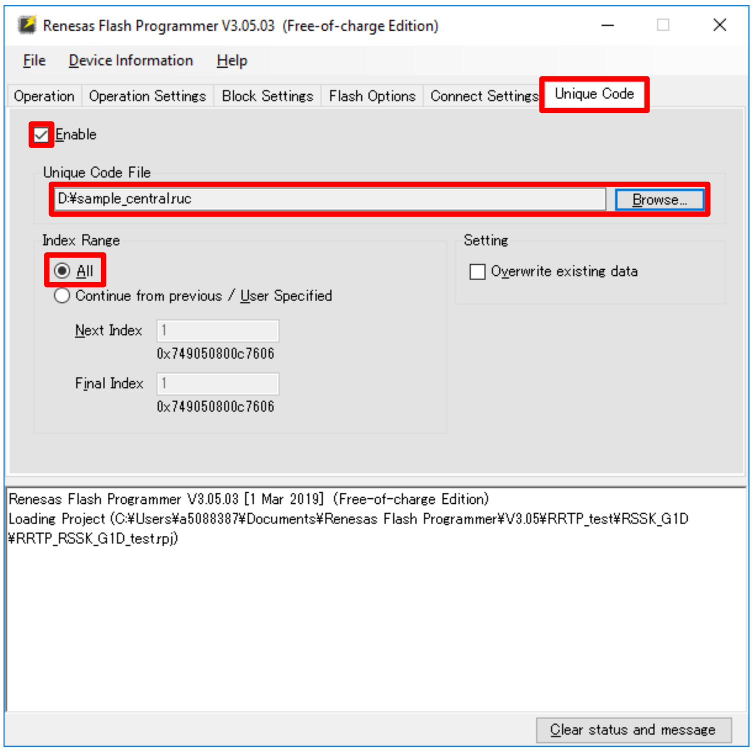
```
Renesas Flash Programmer V3.05.03 [1 Mar 2019] (Free-of-charge Edition)
Loading Project (C:¥Users¥a5088387¥Documents¥Renesas Flash Programmer¥V3.05¥RRTP_test¥RSSK_G1D
¥RRTP_RSSK_G1D_test.rpj)
```

Clear status and message

Figure 2-13 RFP "Block Setting" Tab

4. On the "Unique Code" tab, check "Enable", set "Index Range" to "All", and select the unique code file (sample_central.ruc) that you created.

The "Flash Options" tab and the "Connection Settings" tab are the default settings and do not need to be changed.



Renesas Flash Programmer V3.05.03 (Free-of-charge Edition)

File Device Information Help

Operation | Operation Settings | Block Settings | Flash Options | Connect Settings | Unique Code

☑ Enable

Unique Code File: D:¥sample_central.ruc Browse...

Index Range
- [✓ All]
- Continue from previous / User Specified

Next Index: 1
0x749050800c7606

Final Index: 1
0x749050800c7606

Setting
- ☐ Overwrite existing data

```
Renesas Flash Programmer V3.05.03 [1 Mar 2019] (Free-of-charge Edition)
Loading Project (C:¥Users¥a5088387¥Documents¥Renesas Flash Programmer¥V3.05¥RRTP_test¥RSSK_G1D
¥RRTP_RSSK_G1D_test.rpj)
```

Clear status and message

Figure 2-14 RFP "Unique Code" Tab

5. On the ”Operation” tab, press “Start” to execute the writing (see Figure 2-11). Close the window and exits after normal completion.

2.6.4 Demo Software

(1) Peripheral Device

The following shows the state transition of the demo software on the Peripheral device.

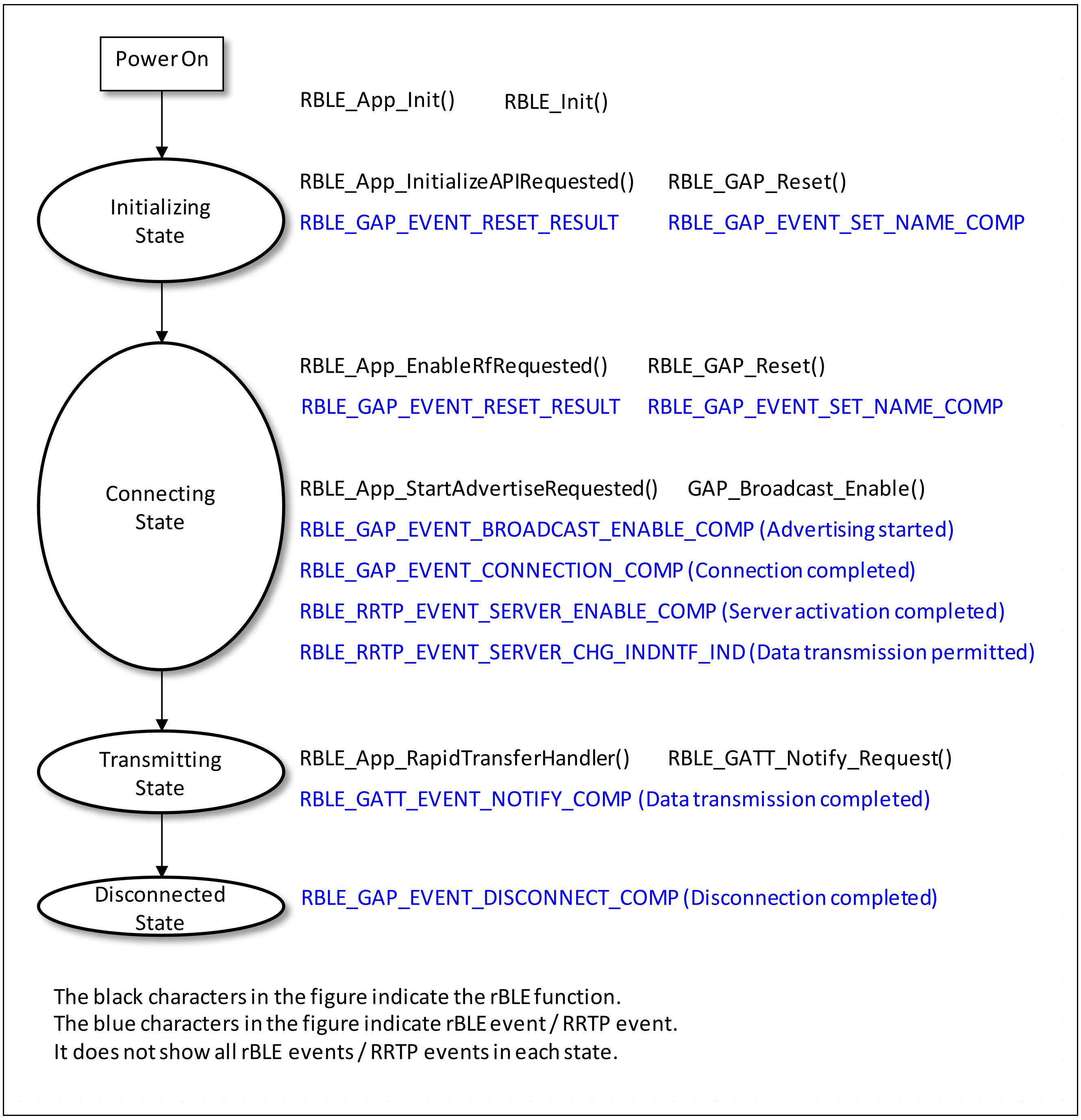

Power On
RBLE_App_Init() RBLE_Init()
Initializing State
RBLE_App_InitializeAPIRequested() RBLE_GAP_Reset()
RBLE_GAP_EVENT_RESET_RESULT RBLE_GAP_EVENT_SET_NAME_COMP
Connecting State
RBLE_App_EnableRfRequested() RBLE_GAP_Reset()
RBLE_GAP_EVENT_RESET_RESULT RBLE_GAP_EVENT_SET_NAME_COMP
RBLE_App_StartAdvertiseRequested() GAP_Broadcast_Enable()
RBLE_GAP_EVENT_BROADCAST_ENABLE_COMP (Advertising started)
RBLE_GAP_EVENT_CONNECTION_COMP (Connection completed)
RBLE_RRTP_EVENT_SERVER_ENABLE_COMP (Server activation completed)
RBLE_RRTP_EVENT_SERVER_CHG_INDNTF_IND (Data transmission permitted)
Transmitting State
RBLE_App_RapidTransferHandler() RBLE_GATT_Notify_Request()
RBLE_GATT_EVENT_NOTIFY_COMP (Data transmission completed)
Disconnected State
RBLE_GAP_EVENT_DISCONNECT_COMP (Disconnection completed)
The black characters in the figure indicate the rBLE function.
The blue characters in the figure indicate rBLE event / RRTP event.
It does not show all rBLE events / RRTP events in each state.

Figure 2-15 State Transition of Demo Software on Peripheral Device

(2) Central Device

The following shows the state transition of the demo software on the Central device.

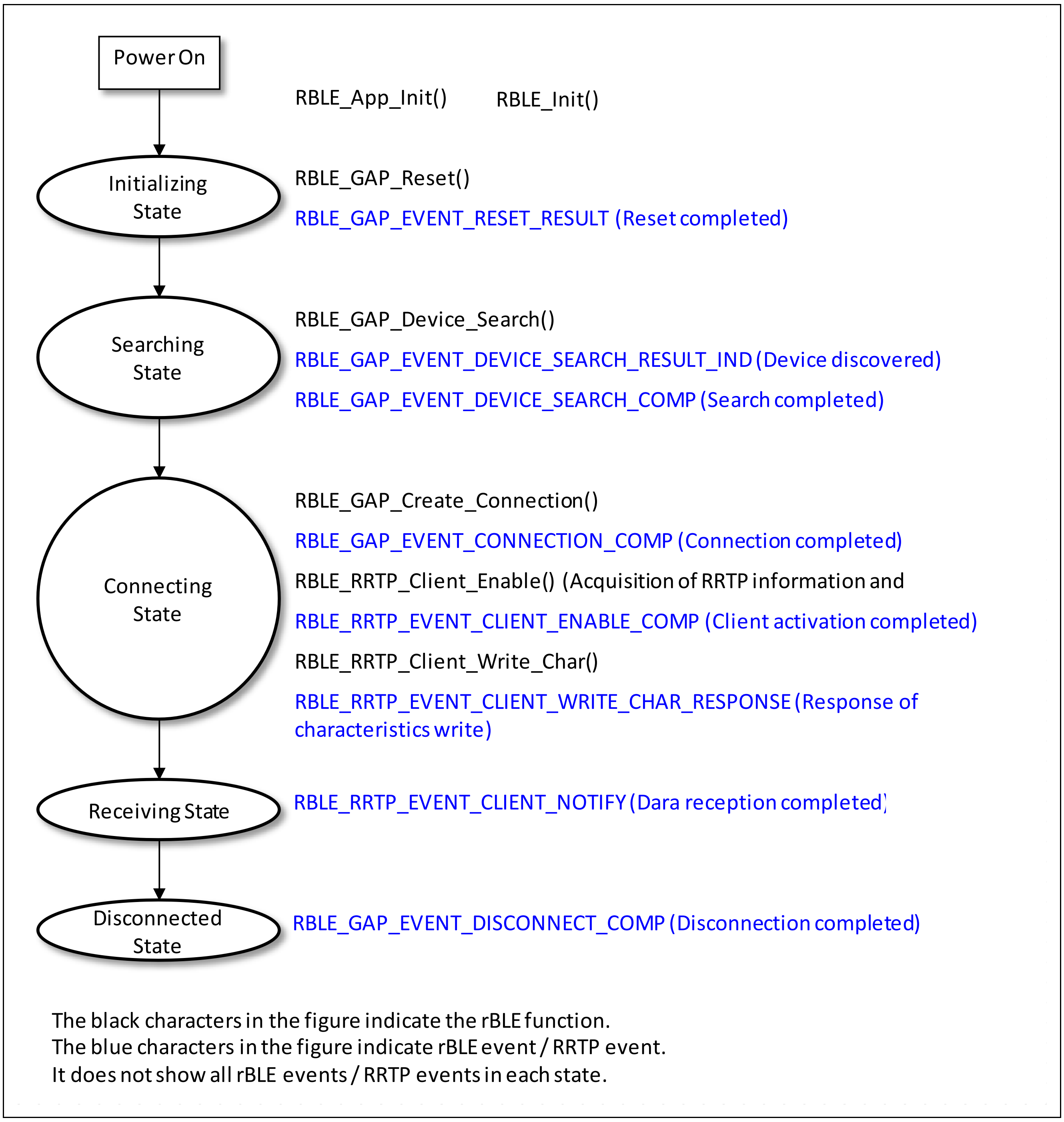

Power On
RBLE_App_Init()
RBLE_Init()
Initializing State
RBLE_GAP_Reset()
RBLE_GAP_EVENT_RESET_RESULT (Reset completed)
Searching State
RBLE_GAP_Device_Search()
RBLE_GAP_EVENT_DEVICE_SEARCH_RESULT_IND (Device discovered)
RBLE_GAP_EVENT_DEVICE_SEARCH_COMP (Search completed)
Connecting State
RBLE_GAP_Create_Connection()
RBLE_GAP_EVENT_CONNECTION_COMP (Connection completed)
RBLE_RRTP_Client_Enable() (Acquisition of RRTP information and
RBLE_RRTP_EVENT_CLIENT_ENABLE_COMP (Client activation completed)
RBLE_RRTP_Client_Write_Char()
RBLE_RRTP_EVENT_CLIENT_WRITE_CHAR_RESPONSE (Response of characteristics write)
Receiving State
RBLE_RRTP_EVENT_CLIENT_NOTIFY (Dara reception completed)
Disconnected State
RBLE_GAP_EVENT_DISCONNECT_COMP (Disconnection completed)
The black characters in the figure indicate the rBLE function.
The blue characters in the figure indicate rBLE event / RRTP event.
It does not show all rBLE events / RRTP events in each state.

Figure 2-16 State Transition of Demo Software on Central Device

2.6.5 Demonstration Procedure

Operate in the following order.

Operate in the following order.

1. Set Up Demo

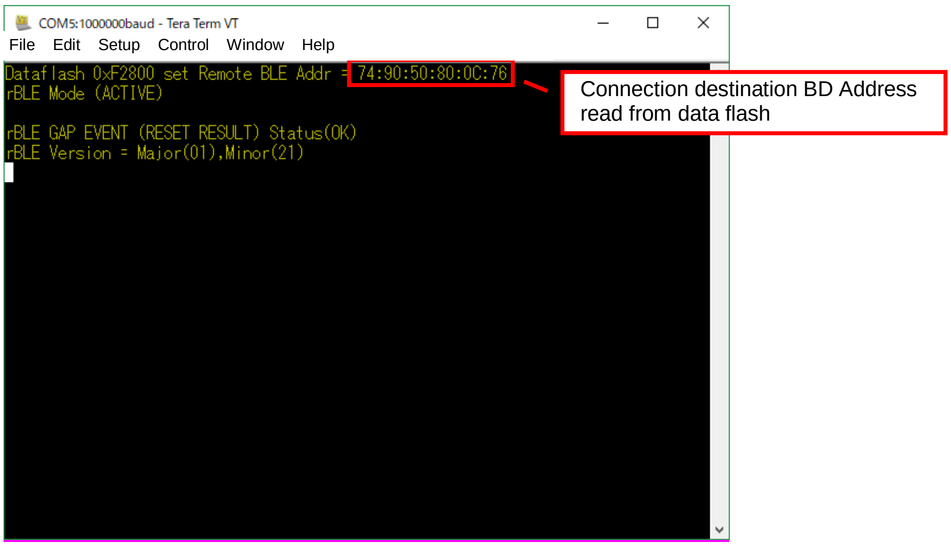
   Refer to “2.6.2 Preparation of Peripheral Device (RL78/G1D Evaluation Board)” and “2.6.3 Preparation of Central Device (RL78/G1D Evaluation Board)”
   Do not supply power to the Peripheral devices (RL78/G1D evaluation board). This is to set up the Central devices in advance.

2. Start terminal software on a PC connected to the Central device (RL78/G1D evaluation board)

   Connect the RL78/G1D evaluation board to a PC, start up the terminal software on the PC, select "Serial", and set the "Port" to the COM number to which the RL78/G1D evaluation board is connected.
   For settings of terminal software, refer to “2.6.3(2) Preparation of PC Terminal Software“.
   Set to get log from serial port. This is because the Central device starts connection procedure after reset release and outputs a log.

3. Reset the Central device (RL78/G1D evaluation board)

   Press the reset switch (SW5) on the Central device (RL78/G1D evaluation board) to reset.
   After releasing the reset, the log display on the terminal software starts. Wait for the Peripheral device to be in the Advertising state.


COM5:1000000baud - Tera Term VT
File Edit Setup Control Window Help
Connection destination BD Address read from data flash

```
Dataflash 0xF2800 set Remote BLE Addr = 74:90:50:80:0C:76
rBLE Mode (ACTIVE)

rBLE GAP EVENT (RESET RESULT) Status(OK)
rBLE Version = Major(01),Minor(21)
```

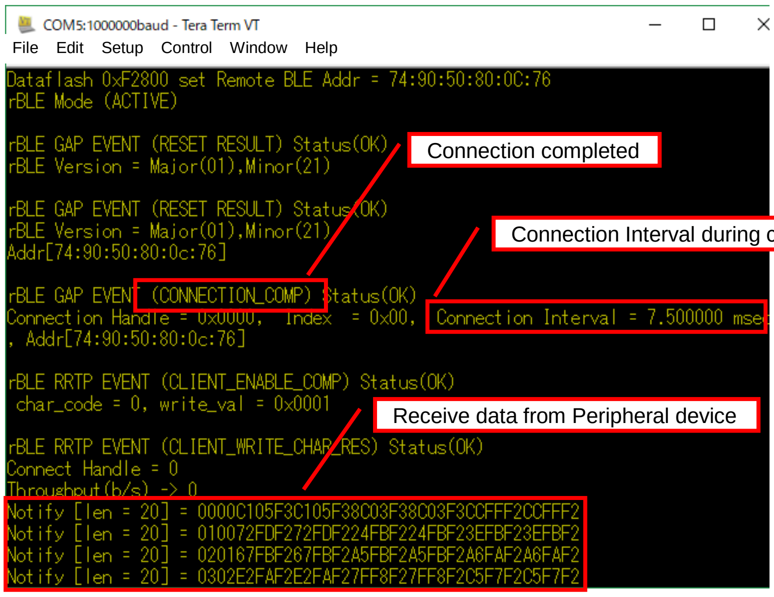
Figure 2-17 Log Output Waiting for Advertisement of Peripheral Device

4. Power on the Peripheral device (RL78/G1D evaluation board)

   Advertisement is started automatically after power on. After connection is made, the Peripheral device starts transmitting data.

5. The Central device (RL78/G1D evaluation board) connects with the Peripheral device (RL78/G1D evaluation board)

   Once connected, the terminal software connected to the Central device will display the received data and information such as the received throughput.


COM5:1000000baud - Tera Term VT
File Edit Setup Control Window Help
Connection completed
Connection Interval during connection
Receive data from Peripheral device

```
Dataflash 0xF2800 set Remote BLE Addr = 74:90:50:80:0C:76
rBLE Mode (ACTIVE)

rBLE GAP EVENT (RESET RESULT) Status(OK)
rBLE Version = Major(01),Minor(21)

rBLE GAP EVENT (RESET RESULT) Status(OK)
rBLE Version = Major(01),Minor(21)
Addr[74:90:50:80:0c:76]

rBLE GAP EVENT (CONNECTION_COMP) Status(OK)
Connection Handle = 0x0000,  Index  = 0x00, Connection Interval = 7.500000 mse
, Addr[74:90:50:80:0c:76]

rBLE RRTP EVENT (CLIENT_ENABLE_COMP) Status(OK)
 char_code = 0, write_val = 0x0001

rBLE RRTP EVENT (CLIENT_WRITE_CHAR_RES) Status(OK)
Connect Handle = 0
Throughput(b/s) -> 0
Notify [len = 20] = 0000C105F3C105F38C03F38C03F3CCFFF2CCFFF2
Notify [len = 20] = 010072FDF272FDF224FBF224FBF23EFBF23EFBF2
Notify [len = 20] = 020167FBF267FBF2A5FBF2A5FBF2A6FAF2A6FAF2
Notify [len = 20] = 0302E2FAF2E2FAF27FF8F27FF8F2C5F7F2C5F7F2
```

Figure 2-18 Log Output of Automatic Connection Completion and Data Reception

6. Operation to stop the demo

   Press Reset SW (SW5) on the Peripheral device (RL78/G1D evaluation board). It will be disconnected. It remains the same after reset release.

7. Close the log file

   Select "Show Log dialog…" from the terminal software "File" menu and press the "Close" button.

8. Operation to restart the demo

   Follow the steps below.

   I. Start up the terminal software, select "Log…" from the "File" menu and set the log output file name. If you do not change the file name, the file is overwritten.
   II. Press and hold the reset SW (SW5) for both the Central device (RL78/G1D evaluation board) and the Peripheral device (RL78/G1D evaluation board).
   III. Release the reset only for the Central device (RL78/G1D evaluation board). The log display is started on the terminal software. Wait for the Peripheral device to be in the Advertising state.
   IV. Only reset the Peripheral device (RL78/G1D evaluation board). it starts advertisement automatically. After the Peripheral device is connected, it starts transmitting data.

2.6.6 Reference Data

The following shows the evaluation results of the simple demo. The evaluation results below are for reference only, as they depend on radio wave conditions.

From the evaluation results, we recommend the following.

- A system clock frequency of 16 MHz or more is required. The system clock frequency of 32 MHz is recommended for user application loads.
- If there are 4 packets in single Connection Interval, many missing data packets may occur. It is recommended to use it when the number of packets in single Connection Interval is 3 or less (sampling rate of user data: 64 kbps).
- The transmit data order and the receive data order may be switched. It is recommended to assign a serial number to part of user data and reconstruct data based on the serial number after data is received.

Table 2-19 Transmission Evaluation Result (Reference)

Number of Packets in Single Connection Interval	**RL78/G1D System Clock Frequency [MHz]**	**Central Device (Client) Received Throughput of User Data [kbps]**	**Number of Transmission Data Packets**	**Number of Missing Data Packets [Note 1]**	**Number of Retrans-mission Data Packets [Note 2]**
4	32	70.94	133,896	27,328	0
	16	69.51	132,510	28,093	8,738
	8	49.79 [Note 3]	93,041	0	141
3	32	64.02	122,036	1	0
	16	64.01	122,822	0	0
	8	49.84 [Note 3]	94,376	0	156

Note 1: Indicates the number of packets that could not be received on the receiving device. Does not include the number of resent data packets.

Note 2: Indicates the number of packets in which the reception order has been switched by the retransmission procedure etc. between BLE. In this case, packets loss of serial number occurs on the receiving device, but packets of the missing serial number is received later.

Note 3: Even though there are no missing packets, the receive data rate (the rate at which data can be sent to BLE software) is limited because the receive throughput has not reached the theoretical value.

3. About High-Speed Data Transfer Operation Demo

This explains demo which transmits the measurement data from the sensor device and displays the reception data on the PC via the opposing device.

3.1 High-Speed Data Transfer Operation Demo System

3.1.1 System Configuration

A Blood Pressure Monitoring Evaluation Kit (BPMEK) is used as a Peripheral device, and transmission packets are created from data acquired by 24-bit ΔΣ AD converter on RL78/H1D.

The measurement data is received using the RL78/G1D evaluation board as a Central device of the opposing device. Received data and communication status information such as reception throughput can be displayed on PC terminal software via built-in UART to USB conversion device.

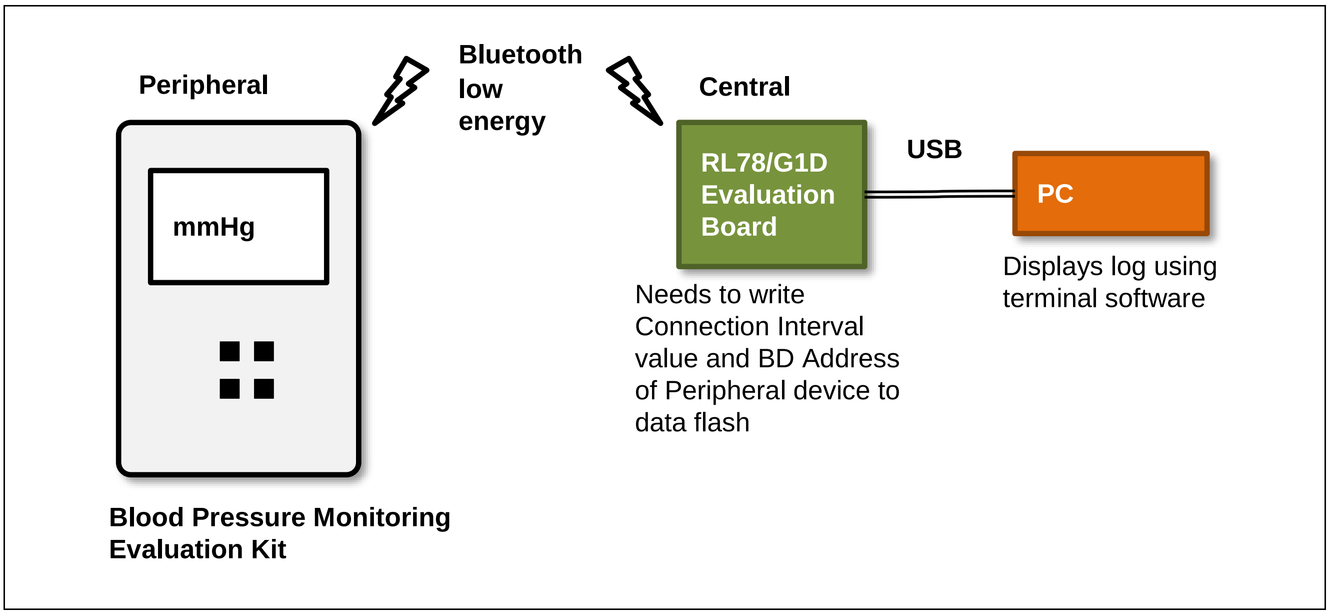

Peripheral
Bluetooth low energy
Central
mmHg
RL78/G1D Evaluation Board
USB
PC
Displays log using terminal software
Needs to write Connection Interval value and BD Address of Peripheral device to data flash
Blood Pressure Monitoring Evaluation Kit

Figure 3-1 High-Speed Data Transfer Operation Demo System Configuration

3.1.2 Transmission Data (Packet) Format

The followings are the contents of the transmission data and the transmission data (packet).

Three bytes of measurement data acquired using the ΔΣ AD converter of the BPMEK and three bytes of data to which the digital filter (IIR filter) incorporated in it is applied are considered as one set. Store up to 3 sets of measurement data in one packet.

In addition, serial numbers are stored in each packet, so that the occurrence of communication errors such as packet loss can be checked on the receiving device.

In the demo using the BPMEK, the sampling rate of measurement data is set to about 1 K sample/s, and data communication of about 52 kbps is performed.

Table 3-1 Contents of Transmission Data

Data	Size (Bytes)	Description
SN	1	Serial number: 0x00 to 0xFF It is incremented each time a packet is sent. After 0xFF, it returns to 0x00.
DN	1	Number of Data: 0x01 to 0x03 It Indicates the number of measurement data sets included in one packet.
DATA_AD	3 [Note 1]	Measurement data (24-bit AD conversion result)
DATA_DF	3 [Note 1]	Data after digital filtering of measured data

Note 1: Each data is stored in little endian.

Transmission Order ->

SN	DN	DATA_AD <1>	DATA_DF <1>	DATA_AD <2> [Note 1]	DATA_DF <2> [Note 1]	DATA_AD <3> [Note 2]	DATA_DF <3> [Note 2]

Note 1: If DN is 0x01, 0x000000 is stored.
Note 2: If DN is 0x01 or 0x02, 0x000000 is stored.

Figure 3-2 Transmission Data (Packet) Format

3.2 Preparation of High-Speed Data Transfer Operation Demo

3.2.1 Preparation of Peripheral Device (Blood Pressure Monitoring Evaluation Kit)

The following shows the preparation contents for the high-speed data transfer demo.

(1) Hardware Configuration

Refer to user's manual of the BPMEK.

(2) Firmware Write

The firmware for high-speed data transfer demo only in MOT file format is provided. It has been stored in the following folder. Rewrite the code flash of both RL78/H1D and RL78/G1D using RFP. For writing method, refer to user's manual of the BPMEK.

Note: Do not delete block #255 and #256 when rewriting the code flash of RL78/G1D. For details, Refer to user's manual of the BPMEK.

Table 3-2 Firmware Storage Location

```
├─ROM_Files
│  ├─BPMEK_RL78_G1D
│  │  └─rBLE_Emb_CCRL.mot                    : For RL78/G1D of Blood Pressure Monitoring Evaluation Kit
│  ├─BPMEK_RL78_H1D
│  │  └─BPM_solution_H1D.mot                 : For RL78/H1D of Blood Pressure Monitoring Evaluation Kit
```

(3) IIR Filter Setting of Blood Pressure Monitoring Evaluation Kit

Set IIR Filter of the BPMEK using GUI tool for the BPMEK. For details on setting, refer to 'PC GUI Tool Operation Manual for BPMEK' (R01AN4396EJ). After setting, do not turn off the BPMEK.

The example of IIR filter coefficients for high-speed data transfer operation demo is stored in the following folder. Because the sampling rate of measurement data is changed for high-speed data transfer operation demo, do not use it for the evaluation application using BPMEK.

Table 3-3 Storage Location of IIR Filter Coefficients

```
├─tools
│  ├─IIR_filter_coefficient_OSR1024_reference.csv
```

3.2.1 Preparation of Central Device (RL78/G1D Evaluation Board)

As firmware of Central devices is common, refer to "2.6.3(3) Writing to Code Flash".

In addition, it is necessary to write the BD Address and Connection Interval values for BPMEK to data flash. Edit the unique code file (sample_central.ruc) with a text editor. Set the BD Address of BPMEK of the connection target and Connection Interval value (recommended value: 0x06 (7.5 milliseconds)). For details on data, refer to "2.6.3(4) Writing to Data Flash".

3.3 High-Speed Data Transfer Operation Demo

3.3.1 Demonstration Procedure

Operate in the following order.

1. Set Up Demo

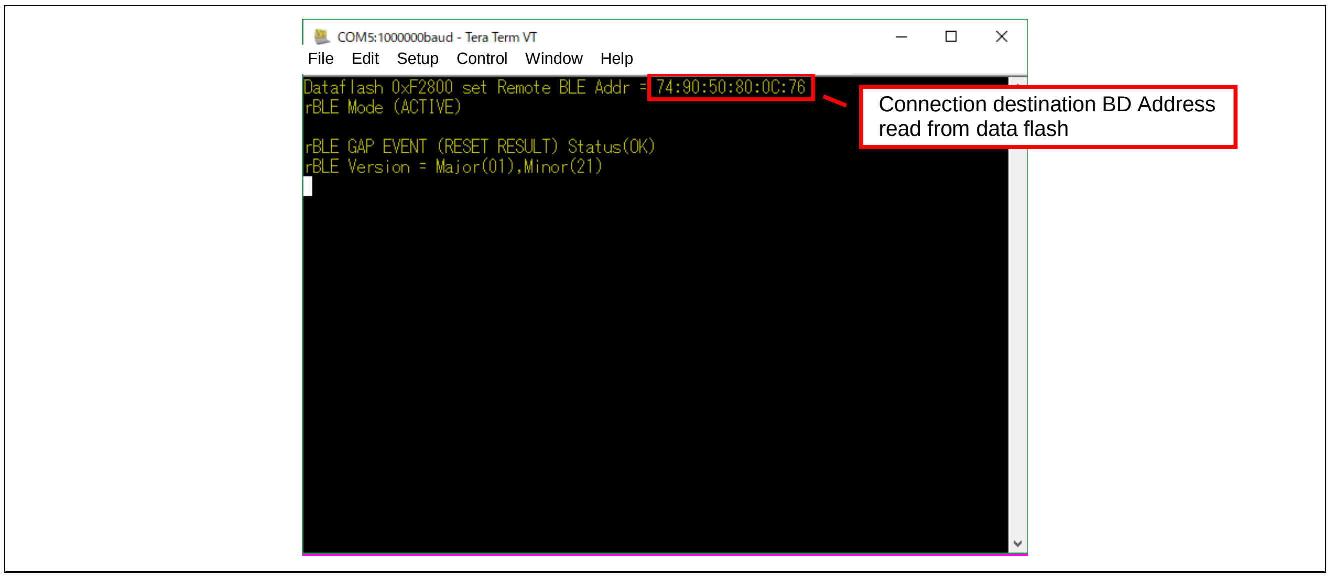
   Refer to “3.2 Preparation of High-Speed Data Transfer Operation Demo”.

2. Start up terminal software on a PC connected to the Central device (RL78/G1D evaluation board)

   Connect the RL78/G1D evaluation board to a PC, start up the terminal software on the PC, select "Serial", and set the "Port" to the COM number to which the RL78/G1D evaluation board is connected.
   For settings of terminal software, refer to “2.6.3(2) Preparation of PC Terminal Software“.
   Set to get log from serial port. This is because the Central device starts connection procedure after reset release and outputs a log.

3. Reset the Central device (RL78/G1D evaluation board)

   Press the reset switch (SW5) on the Central device (RL78/G1D evaluation board) to reset.
   After releasing the reset, the log display on the terminal software starts. Wait for the Peripheral device to be in the Advertising state.


COM5:1000000baud - Tera Term VT
File Edit Setup Control Window Help
Connection destination BD Address read from data flash

```
Dataflash 0xF2800 set Remote BLE Addr = 74:90:50:80:0C:76
rBLE Mode (ACTIVE)

rBLE GAP EVENT (RESET RESULT) Status(OK)
rBLE Version = Major(01),Minor(21)
```

Figure 3-3 Log Output Waiting for Advertisement of Peripheral Device

4. The Central device (RL78/G1D evaluation board) connects with the Peripheral device (Blood Pressure Monitoring Evaluation Kit)

The BPMEK is set to the Advertising status by the switch (SW) operation of the BPMEK shown below.

I. After powering on the BPMEK, press the lower left SW "Down key" once briefly. "RRTP" lights up on the LCD screen, and it moves to the RRTP mode.

II. Press the upper right SW "Enter key" three times briefly. "ADVT" blinks on the LCD screen, and it is in Advertising state. ("BLEC" lights up-> "ENAB" lights up-> "ADVT" blinking will be displayed on the LCD screen in this order.)

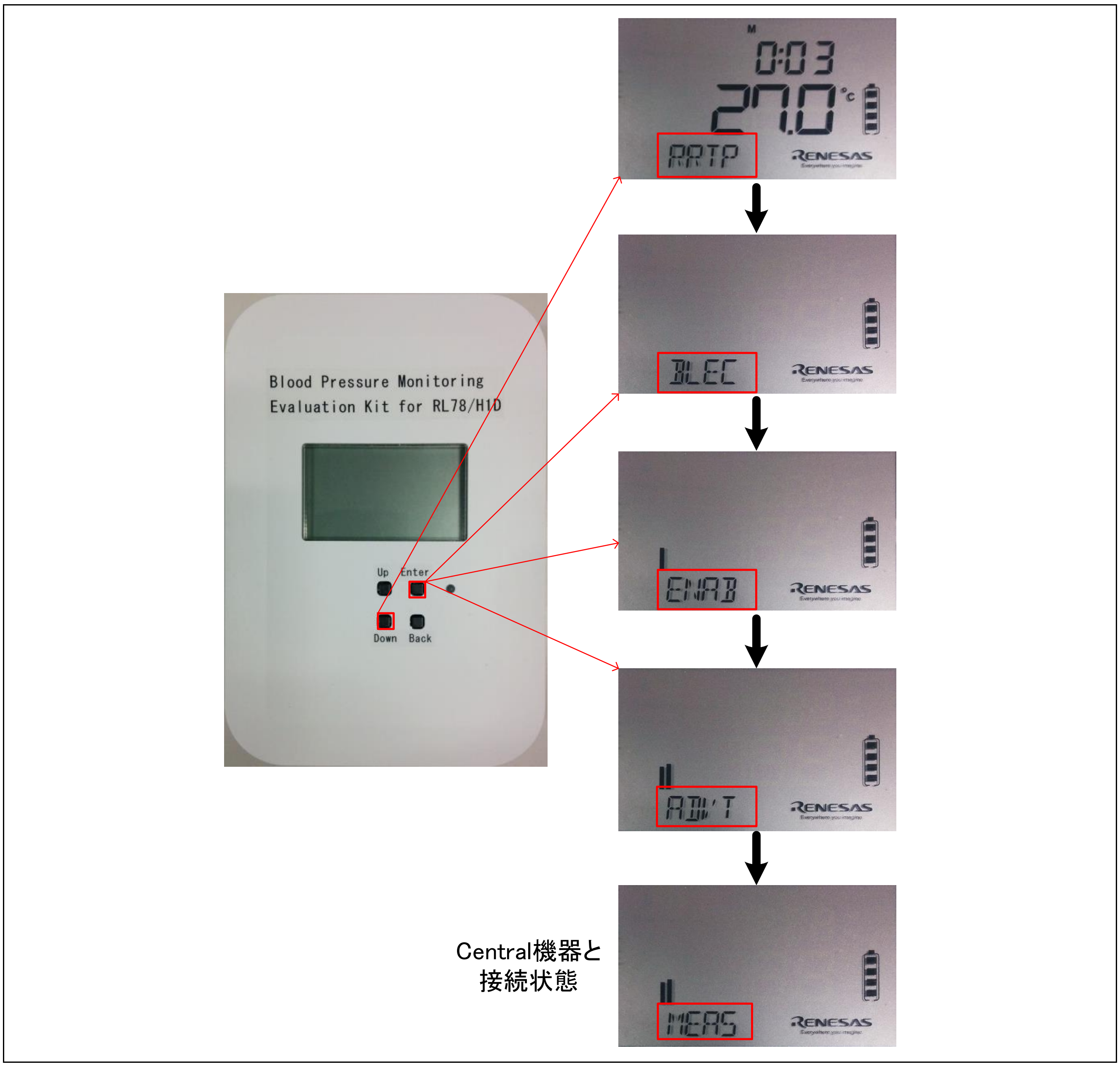
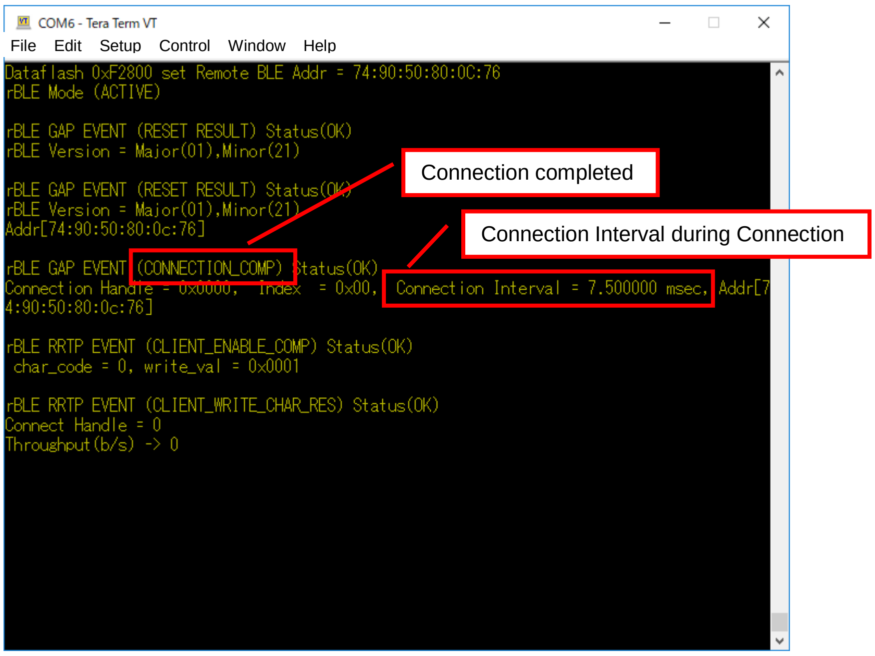


Figure 3-4 LCD Screen Display Transition

III. The Central device (RL78/G1D evaluation board) performs connection processing automatically. When connection is completed, "MEAS" lights on the LCD screen of the BPMEK.


COM6 - Tera Term VT
File Edit Setup Control Window Help
Connection completed
Connection Interval during Connection

```
Dataflash 0xF2800 set Remote BLE Addr = 74:90:50:80:0C:76
rBLE Mode (ACTIVE)

rBLE GAP EVENT (RESET_RESULT) Status(OK)
rBLE Version = Major(01),Minor(21)

rBLE GAP EVENT (RESET_RESULT) Status(OK)
rBLE Version = Major(01),Minor(21)
Addr[74:90:50:80:0c:76]

rBLE GAP EVENT (CONNECTION_COMP) Status(OK)
Connection Handle = 0x0000,  Index  = 0x00,  Connection Interval = 7.500000 msec, Addr[7
4:90:50:80:0c:76]

rBLE RRTP EVENT (CLIENT_ENABLE_COMP) Status(OK)
 char_code = 0, write_val = 0x0001

rBLE RRTP EVENT (CLIENT_WRITE_CHAR_RES) Status(OK)
Connect Handle = 0
Throughput(b/s) -> 0
```

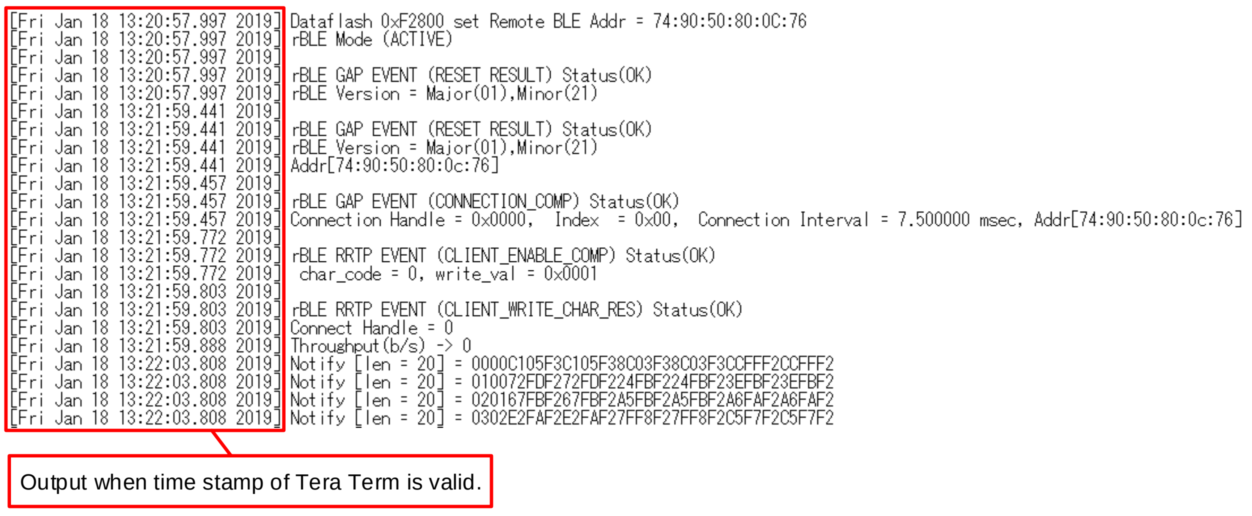
Figure 3-5 Automatic Connection Completion Log Output

5. Measurement start and data transmission with the Peripheral device (Blood Pressure Monitoring Evaluation Kit)

Press the upper right SW "Enter key" on the BPMEK once briefly. The air pump motor of the BPMEK is started, measurement starts, and measurement data is sent.

6. Monitor on PC connected to the Central device (RL78/G1D evaluation board)

The data received by the Central device (RL78/G1D evaluation board) is displayed on the terminal screen.
When measurement is completed, "RRTP" is displayed on the LCD screen of the BPMEK, and log output is stopped.
If you want to stop measurement on the way, press and hold the lower right SW "Back key". “RRTP” is displayed on the LCD screen of the BPMEK, and the log output is stopped.



```
[Fri Jan 18 13:20:57.997 2019] Dataflash 0xF2800 set Remote BLE Addr = 74:90:50:80:0C:76
[Fri Jan 18 13:20:57.997 2019] rBLE Mode (ACTIVE)
[Fri Jan 18 13:20:57.997 2019]
[Fri Jan 18 13:20:57.997 2019] rBLE GAP EVENT (RESET_RESULT) Status(OK)
[Fri Jan 18 13:20:57.997 2019] rBLE Version = Major(01),Minor(21)
[Fri Jan 18 13:21:59.441 2019]
[Fri Jan 18 13:21:59.441 2019] rBLE GAP EVENT (RESET_RESULT) Status(OK)
[Fri Jan 18 13:21:59.441 2019] rBLE Version = Major(01),Minor(21)
[Fri Jan 18 13:21:59.441 2019] Addr[74:90:50:80:0c:76]
[Fri Jan 18 13:21:59.457 2019]
[Fri Jan 18 13:21:59.457 2019] rBLE GAP EVENT (CONNECTION_COMP) Status(OK)
[Fri Jan 18 13:21:59.457 2019] Connection Handle = 0x0000,  Index  = 0x00,  Connection Interval = 7.500000 msec, Addr[74:90:50:80:0c:76]
[Fri Jan 18 13:21:59.772 2019]
[Fri Jan 18 13:21:59.772 2019] rBLE RRTP EVENT (CLIENT_ENABLE_COMP) Status(OK)
[Fri Jan 18 13:21:59.772 2019]  char_code = 0, write_val = 0x0001
[Fri Jan 18 13:21:59.803 2019]
[Fri Jan 18 13:21:59.803 2019] rBLE RRTP EVENT (CLIENT_WRITE_CHAR_RES) Status(OK)
[Fri Jan 18 13:21:59.803 2019] Connect Handle = 0
[Fri Jan 18 13:21:59.888 2019] Throughput(b/s) -> 0
[Fri Jan 18 13:22:03.808 2019] Notify [len = 20] = 0000C105F3C105F38C03F38C03F3CCFFF2CCFFF2
[Fri Jan 18 13:22:03.808 2019] Notify [len = 20] = 010072FDF272FDF224FBF224FBF23EFBF23EFBF2
[Fri Jan 18 13:22:03.808 2019] Notify [len = 20] = 020167FBF267FBF2A5FBF2A5FBF2A6FAF2A6FAF2
[Fri Jan 18 13:22:03.808 2019] Notify [len = 20] = 0302E2FAF2E2FAF27FF8F27FF8F2C5F7F2C5F7F2
```

Figure 3-6 Log Output Save Data of Tera Term (Text Output)

7. Close the log file

   Select "Display Log Dialog" of "File" menu on the terminal software and press the "Close" button.

8. Close the log file

   Operate in the following order.

   I. Start up the terminal software, select "Log…" of the "File" menu and set the log output file name. If you do not change the file name, the file is overwritten.

   II. Reset the Central device (RL78/G1D evaluation board).
   Refer to “3 Reset the Central device (RL78/G1D evaluation board)”.

   III. Press the upper right SW "Enter key" on the Peripheral device (Blood Pressure Monitoring Evaluation Kit) three times briefly. It will be connected.
   Refer to "4 The Central device (RL78/G1D evaluation board) connects with the Peripheral device (Blood Pressure Monitoring Evaluation Kit)". Since it is already in the RRTP mode, perform "II Press the upper right SW "Enter key" three times briefly." It will be in the connection state.

   IV. Press the upper right SW "Enter key" of BPMEK once briefly. Measurement starts and data is sent.
   Refer to “5 Measurement start and data transmission with the Peripheral device (Blood Pressure Monitoring Evaluation Kit)”.

3.3.2 Demonstration Procedure

The log output storage data can be divided into "measurement data 24-bit AD conversion value" and "data after digital filtering of measurement data", and each data can be graphed. Use the macro of "RRTP_graph_sample_rev100.xlsm" file created with Microsoft® Excel® bundled for the graphing. Save the log in order to use the log of the RL78/G1D evaluation board displayed by the terminal software.

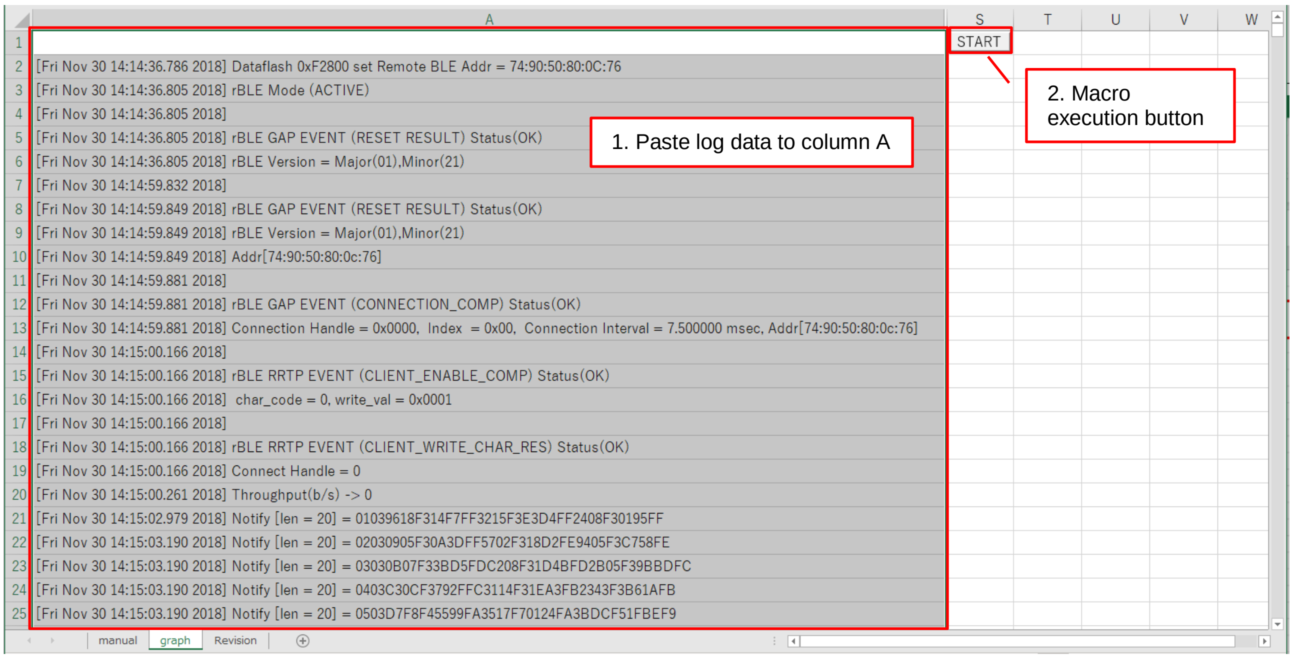
Activate the macro of Excel file and draw the graph in the following procedure.

1. Paste log output save data to row A of "graph" sheet

   Copy the output save log acquired in "3.3.1 6 Monitor on PC connected to the Central device (RL78/G1D evaluation board)" and paste it in row A of the "graph" sheet of the Excel file.

2. Press the "START" button of the macro in the S1 cell of the "graph" sheet

   Push "Execute" button. Two graphs are generated: "AD converted value of measurement data and waveform after filtering" and "Throughput value of BLE communication measured from received data on the Central device".
   Note: The execution time of the macro depends on the amount of log data.



```
1
2  [Fri Nov 30 14:14:36.786 2018] Dataflash 0xF2800 set Remote BLE Addr = 74:90:50:80:0C:76
3  [Fri Nov 30 14:14:36.805 2018] rBLE Mode (ACTIVE)
4  [Fri Nov 30 14:14:36.805 2018]
5  [Fri Nov 30 14:14:36.805 2018] rBLE GAP EVENT (RESET RESULT) Status(OK)
6  [Fri Nov 30 14:14:36.805 2018] rBLE Version = Major(01),Minor(21)
7  [Fri Nov 30 14:14:59.832 2018]
8  [Fri Nov 30 14:14:59.849 2018] rBLE GAP EVENT (RESET RESULT) Status(OK)
9  [Fri Nov 30 14:14:59.849 2018] rBLE Version = Major(01),Minor(21)
10 [Fri Nov 30 14:14:59.849 2018] Addr[74:90:50:80:0c:76]
11 [Fri Nov 30 14:14:59.881 2018]
12 [Fri Nov 30 14:14:59.881 2018] rBLE GAP EVENT (CONNECTION_COMP) Status(OK)
13 [Fri Nov 30 14:14:59.881 2018] Connection Handle = 0x0000,  Index  = 0x00,  Connection Interval = 7.500000 msec, Addr[74:90:50:80:0c:76]
14 [Fri Nov 30 14:15:00.166 2018]
15 [Fri Nov 30 14:15:00.166 2018] rBLE RRTP EVENT (CLIENT_ENABLE_COMP) Status(OK)
16 [Fri Nov 30 14:15:00.166 2018]  char_code = 0, write_val = 0x0001
17 [Fri Nov 30 14:15:00.166 2018]
18 [Fri Nov 30 14:15:00.166 2018] rBLE RRTP EVENT (CLIENT_WRITE_CHAR_RES) Status(OK)
19 [Fri Nov 30 14:15:00.166 2018] Connect Handle = 0
20 [Fri Nov 30 14:15:00.261 2018] Throughput(b/s) -> 0
21 [Fri Nov 30 14:15:02.979 2018] Notify [len = 20] = 01039618F314F7FF3215F3E3D4FF2408F30195FF
22 [Fri Nov 30 14:15:03.190 2018] Notify [len = 20] = 02030905F30A3DFF5702F318D2FE9405F3C758FE
23 [Fri Nov 30 14:15:03.190 2018] Notify [len = 20] = 03030B07F33BD5FDC208F31D4BFD2B05F39BBDFC
24 [Fri Nov 30 14:15:03.190 2018] Notify [len = 20] = 0403C30CF3792FFC3114F31EA3FB2343F3B61AFB
25 [Fri Nov 30 14:15:03.190 2018] Notify [len = 20] = 0503D7F8F45599FA3517F70124FA3BDCF51FBEF9
```

Figure 3-7 Log Data Graphing Operation

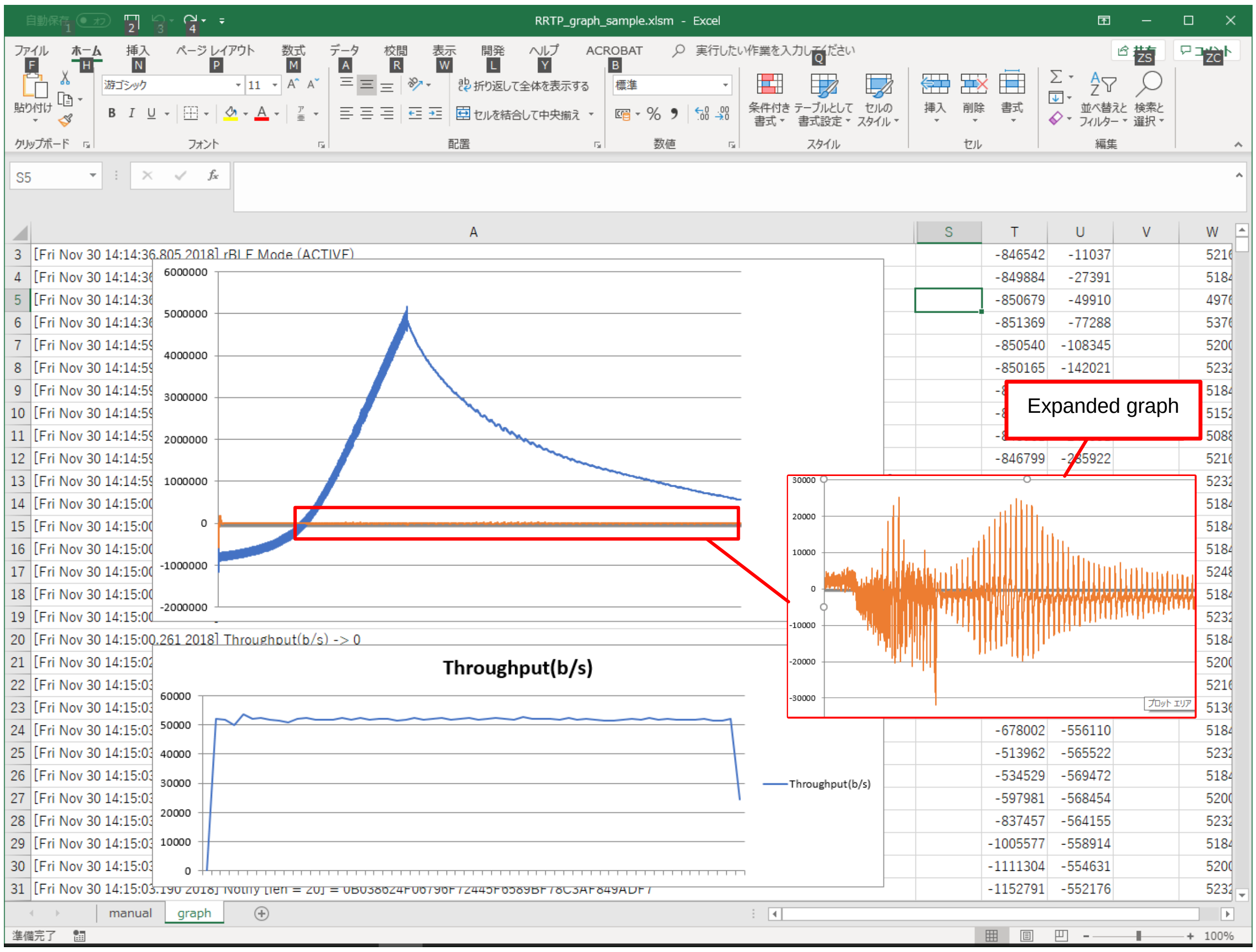
Sample of log output save data is stored in the following folder.

Table 3-4 Sample File Storage Location of Log Output Save Data

```
├─tools
│   ├─RRTP sample.log
```

The Following shows a graph of the sample log output save data.

On the upper graph, the blue graph is "24-bit AD conversion value of measurement data" and the red graph is "data after digital filtering of measurement data". Also, the lower graph shows the throughput results.



```
3  [Fri Nov 30 14:14:36.805 2018] rBLE Mode (ACTIVE)      -846542   -11037
4  [Fri Nov 30 14:14:36                                  -849884   -27391
5  [Fri Nov 30 14:14:36                                  -850679   -49910
6  [Fri Nov 30 14:14:36                                  -851369   -77288
7  [Fri Nov 30 14:14:59                                  -850540  -108345
8  [Fri Nov 30 14:14:59                                  -850165  -142021
12 [Fri Nov 30 14:14:59                                  -846799
20 [Fri Nov 30 14:15:00.261 2018] Throughput(b/s) -> 0
24 [Fri Nov 30 14:15:03                                  -678002  -556110
25 [Fri Nov 30 14:15:03                                  -513962  -565522
26 [Fri Nov 30 14:15:03                                  -534529  -569472
27 [Fri Nov 30 14:15:03                                  -597981  -568454
28 [Fri Nov 30 14:15:03                                  -837457  -564155
29 [Fri Nov 30 14:15:03                                 -1005577  -558914
30 [Fri Nov 30 14:15:03                                 -1111304  -554631
31 [Fri Nov 30 14:15:03.190 2018] Notify [len = 20] = 0B03B624F06796F72445F6589BF78C3AF849ADF7   -1152791  -552176
```

Figure 3-8 Example of Macro Execution Result of Reception Log Data

Revision History

Rev.	Date	Description	
		Page	Summary
1.01	May.31.19	-	First release

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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

Trademarks