QN902x QPP Programming Guide

Version 0.1

Table of Contents

1. Introduction

The QPP (Quintic Private Profile) is used to transfer the raw data between BLE devices.

2. QPP Server

2.1 Project Example

The project can be opened with the following IAR and KEIL workspace file:
C:\Quintic Corporation\QBlue-X.X.X\Projects\BLE\prj_qpps\iar\qpps.eww
C:\Quintic Corporation\QBlue-X.X.X\Projects\BLE\prj_qpps\keil\qpps.uvproj

2.2 Software Description

The QPP application is implemented in the following files:

- app_qpps.c: Application QPPS API
- app_qpps_task.c: Task handling functions
- qpp.lib and qpps_task.h and qpp_common.h: QPP Profile

2.2.1 User Configuration

The following macro shall be defined in the 'usr_config.h'.

- #define **CFG_PRF_QPPS**
- #define **CFG_TASK_QPPS** **TASK_PRF8** (Mandatory)

2.2.2 Initialization

The initialization of the application occurs in two phases: First, the **qpps_init()** function is called by the profiles register function(**prf_init_reg(prf_init)**). This function register QPPS task into kernel. Second, the **app_qpps_create_db(uint8_t char_num)** function is called by the **app_create_server_service_DB()** function. This function used to create server service database, application can define the number of Characteristic used to send data to client through notify.
NOTE: char_num: Max=7 Min = 0. If char_num increases transmission speed will be faster, but more and more space will be occupied.

2.2.3 Data Processing

The application has three data processing functions, app_qpps_data_send(), app_qpps_data_send_cfm_handler() and app_qpps_data_ind_handler(). The app_qpps_data_send() function

is used by the application to send a raw data. The app_qpps_data_send_cfm_handler() function is used to report to the application a confirmation. The app_qpps_data_ind_handler() function is used to handle the data sent form peer device.

The diagrams below shows the relationships between APP and Profile:

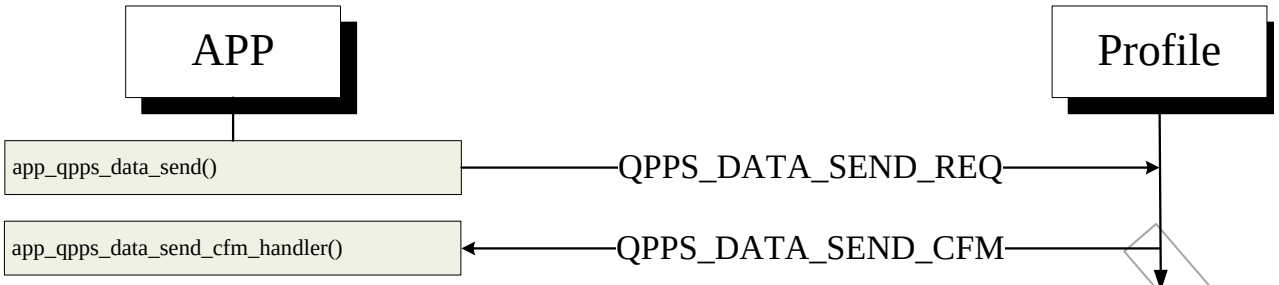

Figure 1 Data Sending

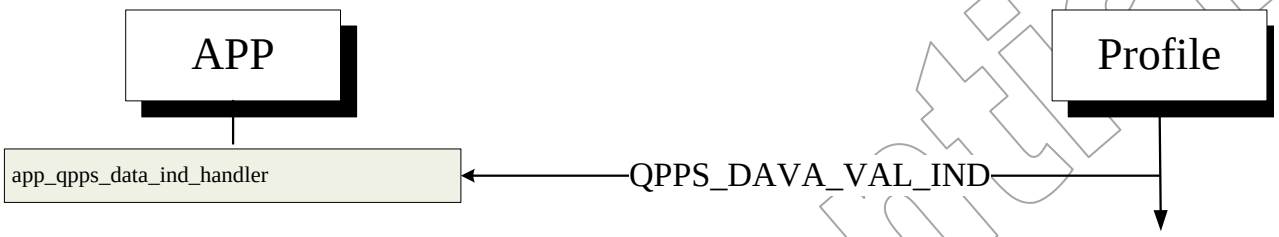

Figure 2 Data Receiving

3. API and Handler

3.1 qpps_init()

Prototype:

void qpps_init(void);

Description:

This function performs all the initializations of the QPPS module.

3.2 app_qpps_create_db()

Prototype:

void app_qpps_create_db (uint8_t char_num);

Parameters:

in	char_num	The number of Characteristic used to send data

Response:

QPPS_CREATE_DB_CFM

Description:

This function shall be used to add an instance of the Quintic Private Profile service into the database. This should be done during the initialization phase of the device.

Note:

Application can define the number of Characteristic used to send data to client through notify.

3.3 app_qpps_enable_req()

Prototype:

void app_qpps_enable_req (uint16_t conhdl, uint8_t sec_lvl, uint8_t con_type, uint16_t ntf_en).

Parameters:

in	conhdl	Connection handle
in	sec_lvl	Security level required for protection of HRS attributes: Service Hide and Disable are not permitted. Possible values are: PERM_RIGHT_ENABLE PERM_RIGHT_UNAUTH PERM_RIGHT_AUTH
in	con_type	Connection type: configuration(0) or discovery(1)
in	ntf_en	Notification configuration

Response:

None

Description:

This function is used for enabling the Server role of the Quintic Private service.

3.4 app_qpps_data_send()

Prototype:

void app_qpps_data_send (uint16_t conhdl, uint8_t index, uint8_t length, uint8_t * data).

Parameters:

in	conhdl	Connection handle
in	index	Index of Characteristic to be sent
in	length	Length of data to be sent Max = 20 Byte
in	data	Pointer to data to be sent

Response:

QPPS_DATA_SEND_CFM

Description:

This function is used by the application to send a raw data.

3.5 app_qpps_create_db_cfm_handler ()

Prototype:

int app_qpps_create_db_cfm_handler (ke_msg_id_t const msgid, struct qpps_create_db_cfm * param, ke_task_id_t const dest_id, ke_task_id_t const src_id)

Parameters:

in	msgid	QPPS_CREATE_DB_CFM
in	param	struct qpps_create_db_cfm
in	dest_id	TASK_APP
in	src_id	TASK_QPPS

Returns:

If the message was consumed or not.

Description:

This handler will be triggered after a database creation. It contains status of database creation.

3.6 app_qpps_disable_ind_handler ()

Prototype:

int app_qpps_disable_ind_handler (ke_msg_id_t const msgid, struct qpps_disable_ind * param, ke_task_id_t const dest_id, ke_task_id_t const src_id)

Parameters:

in	msgid	QPPS_DISABLE_IND
in	param	Pointer to the struct qpps_disable_ind
in	dest_id	TASK_APP
in	src_id	TASK_QPPS

Returns:

If the message was consumed or not.

Description:

This handler is used to inform the Application of a correct disable. The configuration that the client has set in ntf_en field must be conserved for bonded devices.

3.7 app_qpps_error_ind_handler ()

Prototype:

int app_qpps_error_ind_handler (ke_msg_id_t const msgid, struct qpps_error_ind * param, ke_task_id_t const dest_id, ke_task_id_t const src_id)

Parameters:

in	msgid	QPPS_ERROR_IND
in	param	Pointer to the struct qpps_error_ind
in	dest_id	TASK_APP
in	src_id	TASK_QPPS

Returns:

If the message was consumed or not.

Description:

This handler is used to inform the Application of an occurred error.

3.8 app_qpps_data_send_cfm_handler ()

Prototype:

int app_qpps_data_send_cfm_handler (ke_msg_id_t const msgid, struct qpps_data_send_cfm * param, ke_task_id_t const dest_id, ke_task_id_t const src_id)

Parameters:

in	msgid	QPPS_DATA_SEND_CFM
in	param	Pointer to the struct qpps_data_send_cfm
in	dest_id	TASK_APP
in	src_id	TASK_QPPS

Returns:

If the message was consumed or not.

Description:

This handler is used to report to the application a confirmation, or error status of a notification request being sent by application.

3.9 app_qpps_cfg_indntf_ind_handler ()

Prototype:

int app_qpps_cfg_indntf_ind_handler (ke_msg_id_t const msgid, struct qpps_cfg_indntf_ind * param, ke_task_id_t const dest_id, ke_task_id_t const src_id)

Parameters:

in	msgid	QPPS_CFG_INDNTF_IND
in	param	Pointer to the struct qpps_cfg_indntf_ind
in	dest_id	TASK_APP
in	src_id	TASK_QPPS

Returns:

If the message was consumed or not.

Description:

This handler is used to inform application that peer device has changed notification configuration.

3.10 app_qpps_data_ind_handler ()

Prototype:

int app_qpps_data_ind_handler (ke_msg_id_t const msgid, struct qpps_data_val_ind * param, ke_task_id_t const dest_id, ke_task_id_t const src_id)

Parameters:

in	msgid	QPPS_DAVA_VAL_IND
in	param	Pointer to the struct qpps_data_val_ind
in	dest_id	TASK_APP
in	src_id	TASK_QPPS

Returns:

If the message was consumed or not.

Description:

This handler is used to handle the data sent form peer device

References

Included with QUINTIC QBlue-X.X.X Release. The QBlue-X.X.X software has been installed to the default path 'C:\ Quintic Corporation\QBlue-X.X.X':

[1] C:\Quintic Corporation\QBlue-X.X.X\ Documents\Software\
QN9020 Software Developer's Guide v1.1.pdf

[2] C:\Quintic Corporation\QBlue-X.X.X\ Documents\Software\
Quintic Device Database for SW Development User Manual v0.4.pdf

[3] C:\Quintic Corporation\QBlue-X.X.X\ Documents\Software\
QN9020 API Programming Guide v0.8.pdf

Release History

REVISION	CHANGE DESCRIPTION	DATE
0.1	Initial release	2014-05-19

