



QN9020 Easy API Programming Guide

Version 0.4

Copyright ©2012-2013 by Quintic Corporation

Confidential Information contained herein is covered under Non-Disclosure Agreement (NDA)

Table of Contents

1.	Introduction	3
2.	Generic Access Profile (GAP)	3
2.1	eapi_app_gap_advertising()	3
2.2	eapi_app_gap_dev_scan()	3
2.3	eapi_app_gap_conn_req()	3
2.4	eapi_app_gap_security_req ()	4
3.	Profile Client.....	4
3.1	eapi_prf_client_enable_req()	4
3.2	eapi_prf_client_rd_char_req ().....	4
3.3	eapi_prf_client_write_char_req().....	5
4.	Profile Server	5
4.1	eapi_prf_server_value_send()	5
5.	CallBack	6
5.1	app_task_msg_hdl ()	6
	Release History	19

1. Introduction

The purpose of this document is to give guide of the Quintic QN9020 Bluetooth Low Energy (BLE) Easy API programming for software development. This document details the EAPI (Easy Application Programming Interface) definitions.

2. Generic Access Profile (GAP)

2.1 eapi_app_gap_advertising()

Description:

This function is used to set the device to advertising or not.

Prototype:

```
void eapi_app_gap_advertising(bool flag);
```

Parameters:

flag

flag = FALSE; stop advertising

flag = TRUE; start advertising

2.2 eapi_app_gap_dev_scan()

Description:

This function is used to search devices within range or stop the search procedure.

Prototype:

```
void eapi_app_gap_dev_scan(bool flag);
```

Parameters:

flag

flag = FALSE; stop device scan

flag = TRUE; start device scan

2.3 eapi_app_gap_conn_req()

Description:

This function is used to create a connection to a connectable device or disconnect a connection.

Prototype:

```
void eapi_app_gap_conn_req(bool flag, struct bd_addr * addr);
```

Parameters:

flag

flag = FALSE, disconnect

flag = TRUE, connect

addr

The address of the remote device to which the connection will be created

2.4 eapi_app_gap_security_req ()

Description:

This function is used to initiate encryption or pairing procedure.

Prototype:

```
void eapi_app_gap_security_req (struct bd_addr * addr);
```

Parameters:

addr

Pointer to device address of peer

3. Profile Client

3.1 eapi_prf_client_enable_req()

Description:

This function is used to enable the specific profile.

Prototype:

```
void eapi_prf_client_enable_req(uint16_t prf_id, uint16_t conhdl);
```

Parameters:

prf_id

prfile id

conhdl

connection handle

3.2 eapi_prf_client_rd_char_req ()

Description:

This function is used to read the specific characteristic.

Prototype:

```
void eapi_prf_client_rd_char_req (uint16_t prf_id, uint16_t conhdl, uint8_t char_code);
```

Parameters:

prf_id

profile id

conhdl

connection handle

char_code

service characteristics

3.3 eapi_prf_client_write_char_req()

Description:

This function is used to write the specific characteristic.

Prototype:

```
void eapi_prf_client_write_char_req (uint16_t prf_id, uint16_t conhdl, uint16_t char_code,
                                     uint16_t cfg_val);
```

Parameters:

prf_id
profile id
conhdl
connection handle
char_code
service characteristics
cfg_val
configuration characteristics

4. Profile Server

4.1 eapi_prf_server_value_send()

Description:

This function is used to send the value to the client.

Prototype:

```
void eapi_prf_server_value_send(uint16_t prf_id, uint16_t conhdl, union prf_server_value
                                *par);
```

Parameters:

prf_id
profile id
conhdl
connection handle
par
Pointer to the structure related to the value

5. Callback

5.1 app_task_msg_hdl ()

Description:

This is the function for users to call back parameters by messages.

Prototype:

```
void app_task_msg_hdl (ke_msg_id_t msgid,, void const *param));
```

Parameters:

msgid
message id
param
data pointer

The following is a list of all messages that users can call back:

```
*****
*****
```

GAP:

```
*****
*****
```

1.GAP_DEV_INQ_RESULT_EVT

Description: Return result of the inquiry command.

2.GAP_DEV_INQ_REQ_CMP_EVT

Description: Complete event of the device search.

3.GAP_SET_MODE_REQ_CMP_EVT

Description: Complete event of the set mode command.

4.GAP_ADV_REQ_CMP_EVT

Description: Complete event for set advertising parameters and start data broadcast.

5.GAP_LE_CREATE_CONN_REQ_CMP_EVT

Description: Complete event for LE create and cancel connection establishment.

6.GAP_DISCON_CMP_EVT

Description: Complete event of LE connection detachment.

7.GAP_BOND_REQ_CMP_EVT

Description: Complete event for bonding command. This will return the status of the operation (if the bonding procedure has been successful).

BLE_HT_COLLECTOR:

8.HTPC_ENABLE_CFM

Description: This API message is used by the Collector to either send the discovery results of HTS on the Thermometer then confirm enabling of the Collector role, or to simply confirm enabling of Collector role if it is a normal connection and the attribute details are already known.

9.HTPC_RD_CHAR_RSP

Description: This API message is used by the Collector role to inform the Application of a received read response. The status and the data from the read response are passed directly to Application, which must interpret them based on the request it made.

10.HTPC_TEMP_IND

Description: This API message is used by the Collector role to inform the Application of a received temperature value, either by notification (*flag_stable_meas* = intermediate) or indication (*flag_stable_meas* = stable). No confirmation of reception is needed because the GATT sends it directly to the peer.

11.HTPC_MEAS_INTV_IND

Description: This API message is used by the Collector role to inform the Application of a received Measurement Interval Char. Indication and the value it indicates. This value should be used by the Application as seen fit. No response is necessary (the GATT sends the necessary confirmation to the Indication PDU).

BLE_HT_THERMOM:

12.HTPT_MEAS_INTV_CHG_IND

Description: This message is used by the HTPT to inform the application that the measurement interval value has changed. The application uses the new value to either decide to stop periodic measurements if the value of the interval has changed from non 0 to 0, or the opposite, to start periodic measurements using the interval value, if the value has changed from 0 to non 0.

This message will only be issued if the new value that the Collector is trying to write is valid (within the Valid Range descriptor minimum and maximum values). If the value is not within range, this message is never received by the application because the HTPT will send an Error Response to the Collector with the 'Out of Range' code 0x80 and the new value will never be set.

13.HTPT_CFG_INDNTF_IND

Copyright ©2012-2013 by Quintic Corporation

Confidential Information contained herein is covered under Non-Disclosure Agreement (NDA) Page 7

Description: This message is used to inform the Application of a new value set in one of the 3 Client Characteristic Configuration Descriptors in HTS. It allows the application to be aware of its current settings for HTS and to alter its behavior accordingly if the implementation desires it.

BLE_BP_COLLECTOR:

14.BLPC_ENABLE_CFM

Description: This API message is used by the Collector to either send the discovery results of BPS on the Blood Pressure then confirm enabling of the Collector role, or to simply confirm enabling of Collector role if it is a normal connection and the attribute details are already known.

15.BLPC_RD_CHAR_RSP

Description: This API message is used by the Collector role to inform the Application of a received read response. The status and the data from the read response are passed directly to Application, which must interpret them based on the request it made.

16.BLPC_BP_MEAS_IND

Description: This API message is used by the Collector role to inform the Application of a received blood pressure value, either by notification (flag_interm_cp = intermediate) or indication (flag_interm_cp = stable). The application will do what it needs to do with the received measurement. No confirmation of reception is needed because the GATT sends it directly to the peer.

BLE_BP_SENSOR:

17.BLPS_CFG_INDNTF_IND

Description: This message is used by BLPS to inform application that peer device has changed notification configuration.

BLE_HR_COLLECTOR:

18.HRPC_ENABLE_CFM

Description: This API message is used by the Collector to either send the discovery results of HRS on the Heart Rate then confirm enabling of the Collector role, or to simply confirm enabling of Collector role if it is a normal connection and the attribute details are already known.

19.HRPC_RD_CHAR_RSP

Description: This API message is used by the Collector role to inform the Application of a received read response. The status and the data from the read response are passed directly to Application, which must interpret them based on the request it made.

20.HRPC_HR_MEAS_IND

Description: This API message is used by the Collector role to inform the Application of a received Heart Rate value by notification. The application will do what it needs to do with the received measurement. No confirmation of reception is needed because the GATT sends it directly to the peer.

BLE_HR_SENSOR:

21.HRPS_CFG_INDNTF_IND

Description: This message is used by HRPS to inform application that peer device has changed notification configuration.

22.HRPS_ENERGY_EXP_RESET_IND

Description: This message is used by HRPS to inform application that Energy Expanded value shall be reset.

BLE_GL_COLLECTOR:

23.GLPC_ENABLE_CFM

Description: This API message is used by the Collector to either send the discovery results of GLS on the Glucose sensor then confirm enabling of the Collector role, or to simply confirm enabling of Collector role if it is a normal connection and the attribute details are already known.

24.GLPC_REGISTER_CFM

Description: This API message is used by the Collector role to inform the Application about Glucose sensor event registration status.

25.GLPC_READ_FEATURES_RSP

Description: This API message is used by the Collector role to inform the Application of received peer Glucose sensor features.

26.GLPC_RACP_RSP

Description: This API message is used by the Collector role to inform the Application of a status of Record Access Control Point Action. It shall contain status of executed request or number of stored measurement

records if GLP_REQ_REP_NUM_OF_STRD_RECS has been requested.

27.GLPC_MEAS_IND

Description: This API message is used by the Collector role to inform the Application of a received Glucose measurement value. This value should be received within a RACP request (GLP_REQ_REP_STRD_RECS), but it could be send out of request by Glucose sensor.

28.GLPC_MEAS_CTX_IND

Description: This API message is used by the Collector role to inform the Application of a received Glucose measurement context value. This value should be received within a RACP request (GLP_REQ_REP_STRD_RECS), but it could be send out of request by Glucose sensor. It shall be trigger by Glucose sensor only if corresponding glucose measurement previously received has GLP_MEAS_CTX_INF_FOLW in its measurement flag.

BLE_GL_SENSOR:

29.GLPS_CFG_INDNTF_IND

Description: Event triggered when peer device modify notification/indication configuration of Glucose Sensor role characteristics. If peer device has been bonded, configuration that collector has set in GLS attributes (evt_cfg) shall be kept by application in a non-volatile memory for next time this profile role is enabled.

30.GLPS_RACP_REQ_IND

Description: This message is trigged by glucose sensor role when peer collector request to perform a Record Access Control Point (RACP) action.

This action could be report glucose measurements, report number of measurement, delete measurements or abort an on-going operation (see Record Access Control Point (RACP) OP Code). This action contains a filter describing which glucose measurement are concerned by the operation.

Possible operations:

- GLP_REQ_REP_STRD_RECS: Report stored records
- GLP_REQ_REP_NUM_OF_STRD_RECS: Report number of stored records
- GLP_REQ_DEL_STRD_RECS: Delete stored records
- GLP_REQ_ABORT_OP: Abort on-going operation.

Note: During an on-going operation, any other request from peer device will be automatically refused by Glucose service, except GLP_REQ_ABORT_OP (Abort operation). In that case on-going operation shall be stopped. Finally GLPS_RACP_RSP_REQ message shall be sent by application with GLP_REQ_ABORT_OP op_code and status equals GLP_RSP_SUCCESS.

Copyright ©2012-2013 by Quintic Corporation

Confidential Information contained herein is covered under Non-Disclosure Agreement (NDA)Page 10

BLE_FINDME_LOCATOR:

31.FINDL_ENABLE_CFM

Description: This API message is used by the Locator to either send the discovery results of IAS on Target then confirm enabling of the Locator role, or to simply confirm enabling of Locator role if it is a normal connection and the IAS details are already known. An error may have also occurred and is signaled.

BLE_FINDME_TARGET:

32.FINDT_ALERT_IND

Description: This API message is used by the Target role to inform the Application of a valid alert level written by the peer in the IAS Alert Level Characteristic. The message is created and sent after reception and check of a GATT_WRITE_CMD_IND forwarded by TASK_SVC. The connection handle and application task ID stored in the Target environment are used for the creation of the kernel message. Only if a valid level has been written by peer (“No Alert” = 0, “Mild Alert” = 1, “High Alert” = 2) will this message be sent to the application. In any other case the attribute value may be updated in the database, but it will not trigger profile functionality.

The Application alone is responsible for actually triggering/stopping a noticeable visual/audio alert on the device upon reception of this indication.

BLE_PROX_MONITOR:

33.PROXM_ENABLE_CFM

Description: This API message is used by the Monitor to either send the discovery results of LLS, IAS and TPS on Reporter then confirm enabling of the Monitor role, or to simply confirm enabling of the Monitor role if it is a normal connection and the LLS, IAS and TPS details are already known.

34.PROXM_RD_CHAR_RSP

Description: This API message is used by the Monitor role to send the Application the characteristic read response data and the status of the read characteristic request. The application is in charge of deciphering the data or of the next step if an error is received.

BLE_PROX_REPORTER:

35.PROXR_ALERT_IND

Description: This API message is used by the Reporter role to request the Application to start the alert on the device considering the indicated alert level. The message may be created and sent on two conditions:

- The IAS alert level characteristic has been written to a valid value, in which case *alert_lvl* will be set to the IAS alert level value.
- A disconnection with a reason other than the normal local/remote link terminations has been received, in which case *alert_lvl* will be set to the LLS alert level value.

The connection handle and application task ID stored in the Target environment are used for the creation of the kernel message.

The Application actions following reception of this indication are strictly implementation specific (it may try to reconnect to the peer and stop alert upon that, or timeout the alert after acertain time, please see the specification).

BLE_TIP_CLIENT:

36.TIPC_ENABLE_CFM

Description: This API message is used by the Client to either send the discovery results of CTS, NDCS or RTUS and confirm enabling of the Client role (status = PRF_ERR_OK), or to simply confirm enabling of Client role if it is a normal connection and the attribute details are already known (status = PRF_ERR_OK), or to inform the application that the discovery process has been stopped because of a missing attribute (status = PRF_ERR_STOP_DISC_CHAR_MISSING).

37.TIPC_CT_IND

Description: This API message is used by the Client role to inform the Application of a received current time value. The ind_type parameter informs the application if the value has been notified by the Time Client or if it has been received as a read response.

38.TIPC_CT_NTF_CFG_RD_RSP

Description: This API message is used by the Client role to inform the application that the notification configuration value for the Current Time characteristic has been successfully read and to provide this value.

39.TIPC_LTI_RD_RSP

Description: This API message is used by the Client role to inform the application that the LTI characteristic value has been successfully read and to provide this value.

40.TIPC_RTI_RD_RSP

Description: This API message is used by the Client role to inform the application that the RTI characteristic value has been successfully read and to provide this value.

41.TIPC_TDST_RD_RSP

Description: This API message is used by the Client role to inform the application that the TDST characteristic value has been successfully read and to provide this value.

42.TIPC_TUS_RD_RSP

Description: This API message is used by the Client role to inform the application that the TUS characteristic value has been successfully read and to provide this value.

BLE_TIP_SERVER:

43.TIPS_CURRENT_TIME_CCC_IND

Description: This API message is used to inform the application about a modification of the Current Time Client Configuration characteristic value.

44.TIPS_TIME_UPD_CTLN_PT_IND

Description: This API message is used to inform the application about a modification of the Time Update Control Point Characteristic value.

BLE_SP_CLIENT:

45.SCPPC_ENABLE_CFM

Description: This API message is used by the Client role task to either send the discovery results of SCPS on the peer device then confirm enabling of the Client role, or to simply confirm enabling of Client role if it is a normal connection and the attribute details are already known.

46.SCPPC_SCAN_REFRESH_NTF_CFG_RD_RSP

Description: This API message is sent to the application to inform it about the read Client Characteristic Configuration Descriptor value for the Scan Refresh Characteristic.

BLE_SP_SERVER:

47.SCPPPS_SCAN_INTV_WD_IND

Description: This API message informs the application that the Scan Interval Window Characteristic value has been written by the peer device.

48.SCPPPS_SCAN_REFRESH_NTF_CFG_IND

Description: This API message is sent to the application to inform it that the peer device has enabled or disabled sending of notifications for the Scan Refresh Characteristic.

BLE_DIS_CLIENT:

49.DISC_ENABLE_CFM

Description: This API message is used by the Client to either send the discovery results of DIS on Server then confirm enabling of the Client role, or to simply confirm enabling of Client role if it is a normal connection and the DIS details are already known. An error may have also occurred and is signaled.

50.DISC_RD_CHAR_RSP

Description: This API message is used by the Client role to inform the Application of a received read response. The status and the data from the read response are passed directly to Application, which must interpret them based on the request it made.

BLE_BATT_CLIENT:

51.BASC_ENABLE_CFM

Description: This API message is used by the Client role task to either send the discovery results of HID on the peer device and confirm enabling of the Client role, or to simply confirm enabling of Client role if it is a normal connection and the attribute details are already known.

52.BASC_BATT_LEVEL_NTF_CFG_RD_RSP

Description: This API message is sent to the application to inform it about the read Client Characteristic Configuration Descriptor value for the specified Battery Level Characteristic.

53.BASC_BATT_LEVEL_PRES_FORMAT_RD_RSP

Description: This API message is sent to the application to inform it about the read Characteristic Presentation Format Descriptor value for the specified Battery Level Characteristic.

54.BASC_BATT_LEVEL_IND

Description: This API message is sent to the application when a Battery Level Characteristic value has

been received either upon reception of a notification, or upon reception of the read response.

BLE_BATT_SERVER:

55.BASS_BATT_LEVEL_NTF_CFG_IND

Description: This API message is sent to the application when the notification configuration has been modified for one of the Battery Level Characteristics.

SMP:

56. SMPC_SEC_STARTED_IND

Description: This function is used to initiate a encryption or pairing procedure. For a unbonded device, this will initiate pairing, or this will initiate encryption.

BLE_AN_CLIENT:

57. ANPC_VALUE_IND

Description: This API is sent to the application once an attribute value has been received from the peer device upon a notification or a read response message.

58. ANPC_CMP_EVT

Description: The API message is used by the ANPC task to inform the sender of a command that the procedure is over and contains the status of the procedure.

BLE_AN_SERVER:

59. ANPS_NTF_IMMEDIATE_REQ_IND

Description: This message is sent to the application when the Alert Notification Control Point Charactersitic value is written by the peer device with a 'Notify New Incoming Alert Immediately' or 'Notify Unread Category Status Immediately' command id.

The cat_ntf_cfg parameter provided information about the categories that has been required by the peer device and which can be notified (supported + enabled).

If there are no new alerts or no unread alerts for the specified category, the provided number of alert shall

be set to 0.

60. ANPS_NTF_STATUS_UPDATE_IND

Description: This message is sent to the application when the value of one of the two Client Characteristic Configuration descriptors has been written by the peer device to enable or disable sending of notifications. It is also sent upon reception of a write request for the Alert Notification Control Point characteristic with the following command ids:

- ☐ Enable New Incoming Alert Notifications
- ☐ Enable Unread Category Status Notifications
- ☐ Disable New Incoming Alert Notifications
- ☐ Disable Unread Category Status Notifications

61. ANPS_CMP_EVT

Description: The API message is used by the ANPS task to inform the sender of a command that the procedure is over and contains the status of the procedure.

BLE_CSC_COLLECTOR:

62. CSCPC_CMP_EVT

Description: The API message is used by the CSCPC task to inform the sender of a command that the procedure is over and contains the status of the procedure.

63. CSCPC_VALUE_IND

Description: This API message is sent to the application when a new value is received from the peer device within a read response, an indication, or a notification.

BLE_CSC_SENSOR:

64. CSCPS_SC_CTLN_PT_REQ_IND

Description: The message is sent to the application when the SC Control Point characteristic is written by the peer device. The application shall answer using the CSCPS_SC_CTLN_PT_CFM message.

65. CSCPS_NTF_IND_CFG_IND

Description: This message is sent to the application each time a peer device successfully writes the Client Characteristic Configuration descriptor of either the CSC Measurement characteristic or the SC Control Point characteristic.

66. CSCPS_CMP_EVT

Description: The API message is used by the CSCPS task to inform the sender of a command that the procedure is over and contains the status of the procedure.

BLE_PAS_CLIENT:

67. PASPC_CMP_EVT

Description: The API message is used by the PASPS task to inform the sender of a command that the procedure is over and contains the status of the procedure.

68. PASPC_VALUE_IND

Description: This API is sent to the application once an attribute value has been received from the peer device upon a notification or a read response message.

The content of the value parameter depends of the attribute code value which defines the attribute that has been updated.

BLE_PAS_SERVER:

69. PASPS_WRITTEN_CHAR_VAL_IND

Description: This API message is sent to the application to inform it that the value of one of the writable attribute has been successfully written by the peer device.

The content of the value parameter depends on the received attribute code.

When the ringer control point characteristic value is written, the task checks the current state of the device(Ringer Silent or Ringer Normal),if the state can be modified,the PASPC_WRITTEN_CHAR_VAL_IND message is sent to the application which decide if the state can be modified.

70. PASPS_CMP_EVT

Description: The API message is used by the PASPS task to inform the sender of a command that the procedure is over and contains the status of the procedure.

BLE_RSC_COLLECTOR:

71. RSCPC_CMP_EVT

Description: The API message is used by the RSCPC task to inform the sender of a command that the procedure is over and contains the status of the procedure.

72. RSCPC_VALUE_IND

Description: This API message is sent to the application when a new value is received from the peer device within a read response, an indication, or a notification.

BLE_RSC_SENSOR:

73. RSCPS_SC_CTLN_PT_REQ_IND

Description: The message is sent to the application when the SC Control Point characteristic is written by the peer device. The application shall answer using the RSCPS_SC_CTLN_PT_CFM message.

74. RSCPS_NTF_IND_CFG_IND

Description: This message is sent to the application each time a peer device successfully writes the Client Characteristic Configuration descriptor of either the RSC Measurement characteristic or the SC Control Point characteristic.

75. RSCPS_CMP_EVT

Description: The API message is used by the RSCPS task to inform the sender of a command that the procedure is over and contains the status of the procedure.

Release History

REVISION	CHANGE DESCRIPTION	DATE
0.1	Initial release	2013-05-16