



QN9020 Easy ACI Programming Guide

Version 0.6

Copyright ©2012-2014 by Quintic Corporation
Confidential Information contained herein is covered under Non-Disclosure Agreement (NDA)

Table of Contents

1.	Introduction	4
2.	Interface Configuration.....	4
2.1	Wakeup between host and controller	4
2.2	Uart Configuration.....	4
2.3	SPI Configuration.....	5
3.	Packet Format	5
3.1	Format Definition	5
3.2	Field Definition	5
3.3	Byte Ordering	5
3.4	Command	6
3.5	Event.....	7
3.6	Data Request.....	8
3.7	Data Indication.....	8
4.	Generic.....	9
4.1	Advertising command	9
4.2	Scan command	9
4.3	Connect command	10
4.4	Disconnect command.....	10
4.5	Set Device Name command.....	11
4.6	Bond command	11
4.7	Central Update Parameter command	12
4.8	Peripheral Update Parameter command	12
4.9	Advertising status event	13
4.10	Scan result event	13
4.11	Scan complete event	13
4.12	Connection complete event.....	13
4.13	Disconnection event	14
4.14	Set Device name event.....	14
4.15	Bond complete event	14
4.16	Central Update Parameter event.....	14
4.17	Peripheral Update Parameter event.....	15
5.	Profile Enable	15
5.1	Request.....	15
5.2	Indication	15
5.3	UUID	16
6.	Alert Notification Status Profile	17
6.1	Data Request.....	17
6.2	Data Indication.....	17
7.	Battery Profile.....	18
7.1	Data Request.....	18
7.2	Data Indication.....	18
8.	Blood Pressure Profile	19
8.1	Data Request.....	19

8.2	Data Indication.....	19
9.	Cycling Speed and Cadence Profile.....	20
9.1	Data Request.....	20
9.2	Data Indication.....	21
10.	Device Information Service Profile	21
10.1	Data Request.....	21
10.2	Data Indication.....	22
11.	Find Me Profile.....	22
11.1	Data Request.....	22
11.2	Data Indication.....	22
12.	Glucose Profile	23
12.1	Data Request.....	23
12.2	Data Indication.....	24
13.	Heart Rate Profile	25
13.1	Data Request.....	25
13.2	Data Indication.....	25
14.	Health Thermometer Profile	26
14.1	Data Request.....	26
14.2	Data Indication.....	26
15.	Phone Alert Status Profile.....	27
15.1	Data Request.....	27
15.2	Data Indication.....	27
16.	Proximity Profile	28
16.1	Data Request.....	28
16.2	Data Indication.....	28
17.	Running Speed and Cadence Profile	29
17.1	Data Request.....	29
17.2	Data Indication.....	30
18.	Scan Parameter Profile	30
18.1	Data Request.....	30
18.2	Data Indication.....	31
19.	Time Profile.....	31
19.1	Data Request.....	31
19.2	Data Indication.....	31
20.	Demo Example	32
20.1	Download - LPC1768	32
20.2	Download – QN90xx	32
20.3	Data Request.....	33
20.4	User Configuration	34
20.5	SPI Configuration.....	35
	Release History.....	36

1. Introduction

This document is intended for software engineers working with the Quintic-BLE stack software. The following diagram shows the relationships between QN9020 and MCU. Easy ACI(EACI) provides two kinds of interface, which are UART and SPI.

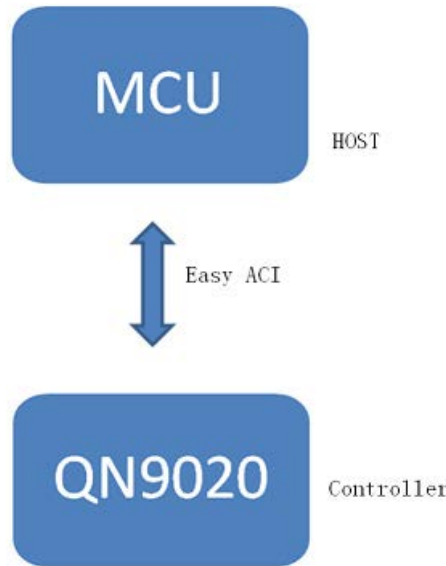


Figure 1:Easy ACI Architecture

2. Interface Configuration

2.1 Wakeup between host and controller

In the deep sleep mode, only the interrupt of GPIO can wakeup system. In this way, two GPIO should be connected between host unit and controller unit to wakeup each other:

Host wakeup Controller: GPIO 16, one 4ms duration positive pulse signal is needed.

Controller wakeup Host: GPIO 15, one 4ms duration positive pulse signal is needed.

2.2 Uart Configuration

1. Baud rate: default value is 9600 bps(supported 1200bps to 2000000bps)
2. No parity
3. LSB transmitting first
4. 8 bit data width
5. 1 bit stop

2.3 SPI Configuration

Host (SPI Master, LCP17xx)	Controller (SPI Slave, QN902x)
Data Rate: 1 Mbps	Supported Data Rate: up to AHB_CLK/2
Data Width: 8 bits	Data Width: 8 bits
Mode: CPOL_0, CPHA_0	Mode: CPOL_0, CPHA_1

3. Packet Format

This chapter defines the generic packet format of the message communicated between host unit and controller unit. The message can be divided into 4 kinds of types: command, event, data request and data indication.

3.1 Format Definition

All messages packet are composed with the following format:

LSB				MSB
MSG_TYPE	MSG_ID	PAR_LEN	PARAMS	

3.2 Field Definition

Field	Size in Byte	Description
MSG_TYPE	1	0xEA: command, 0xEB: data request, 0xEC: data indication, 0xED: event,
MSG_ID	1	Message identifier, depend on MSG_TYPE
PAR_LEN	1	Parameter length
PARAMS	n	Parameter data

3.3 Byte Ordering

All commands are used with the EACI should be transmitted with the least significant octet first(Little Endian). Over the air where MSG_TYPE is sent first and PARAMS is sent last.

3.4 Command

The command is sent from MCU to QN9020. MSG_TYPE is 0xEA.

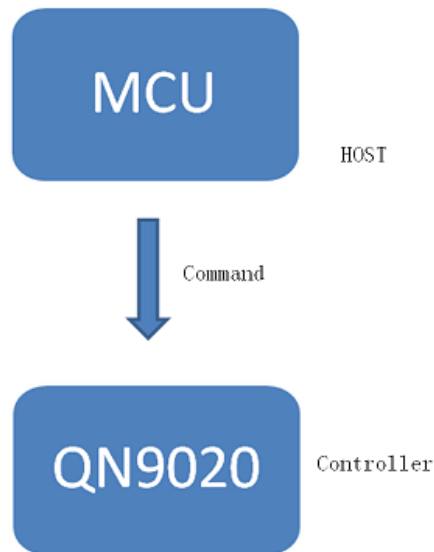


Figure 2: Command

Command List:

MSG_ID	Description
0x01	Advertising Command
0x02	Scan Command
0x03	Connection Command
0x04	Disconnection Command
0x05	Set Device Name Command
0x06	Bond Command
0x07	Central Update Param Command
0x08	Peripheral Update Param Command

3.5 Event

The event is sent from QN9020 to MCU. MSG_TYPE is 0xED.

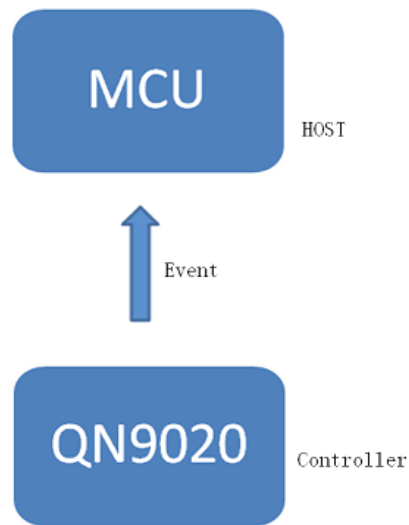


Figure 3: Event

Event List:

MSG_ID	Description
0x01	Advertising command complete event
0x02	Scan result event
0x03	Scan complete event
0x04	Connection complete event
0x05	Disconnection event
0x06	Device Name complete event
0x07	Bond complete event
0x08	Central Update param event
0x0A	Peripheral Update param event

3.6 Data Request

The Data Request is sent from MCU to QN9020. MSG_TYPE is 0xEB.

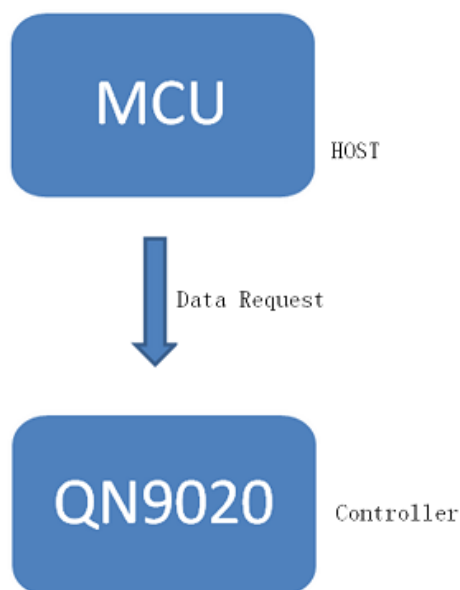


Figure 4: Request

Format Definition

Fields	Description
MSG_ID	Message identifier of data request
PAR_LEN	The length of PARAMS
PARAMS	Data

3.7 Data Indication

The Data Indication is sent from QN9020 to MCU. MSG_TYPE is 0xEC.

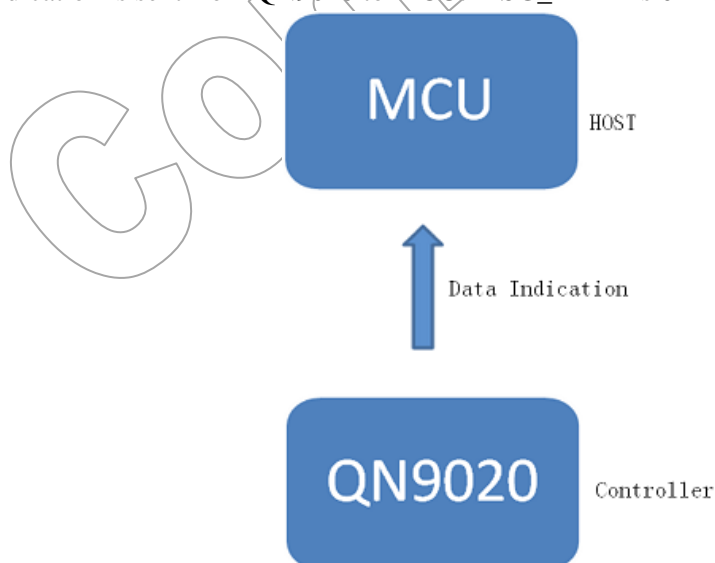


Figure 5: Indication

Format Definition

Fields	Description
MSG_ID	Message identifier of data indication
PAR_LEN	The length of PARAMS
PARAMS	Data

4. Generic

4.1 Advertising command

- Description**

This command is used to star or stop the advertising procedure.

- Format definition**

Fields	Description
MSG_ID	0x01, advertising command
PAR_LEN	0x05
flag	0x01: start advertising, 0x00: stop advertising
Adv_intv_min	Default: 0x0030 (Range: 0x0006 to 0x0C80) Min Advertising Interval (0x0030 * 0.625 ms)
Adv_intv_max	Default: 0x0064 (Range: 0x0006 to 0x0C80) - Max Advertising Interval (0x0064 * 0.625 ms)

- Hex Command Data**

EA 01 05 01 30 00 64 00 (start advertising)

EA 01 05 00 00 00 00 00 (stop advertising)

- Event generated**

See section 4.9 Advertising status event

4.2 Scan command

- Description**

This command is used to scan the nearby BLE devices.

- Format definition**

Fields	Description
MSG_ID	0x02, Scan command
PAR_LEN	0x01
flag	0x01: start scan, 0x00:stop scan

- **Hex Command Data**

EA 02 01 01 (start scan)

EA 02 01 00 (stop scan)

- **Event generated**

See section 4.10 Scan result event and 4.11 Scan complete event

4.3 Connect command

- **Description**

This command is used to connect with the specified BLE device.

- **Format definition**

Fields	Description
MSG_ID	0x03, Connect command
PAR_LEN	0x0D
bd_addr_type	1 bytes, (0x00 – public; 0x01 random) Bluetooth device address type
Conn_intv_min	Default: 0x0018 (Range: 0x0006 to 0x0C80) – Minimum connection interval(0x0018 * 1.25 ms)
Conn_intv_max	Default: 0x0028 (Range: 0x0006 to 0x0C80) – Maximum connection interval(0x0028 * 1.25 ms)
Cnnn_timeout	Default: 0x07D0 (Range: 0x000A to 0x0C80) - Connection supervision timeout(0x07D0 * 10 ms)
bd_addr	6 bytes, Bluetooth device address

- **Hex Command Data**

EA 03 0D XX 18 00 28 00 D0 07 XX XX XX XX XX XX

- **Event generated**

See section 4.12 Connection complete event

4.4 Disconnect command

- **Description**

This command is used to disconnect the specified connection.

- **Format definition**

Fields	Description
MSG_ID	0x04, disconnect command
PAR_LEN	0x06
bd_addr	6 bytes, Bluetooth device address

- **Hex Command Data**

EA 04 06 XX XX XX XX XX XX

- **Event generated**

See section 4.13 Disconnection event

4.5 Set Device Name command

- **Description**

This command is used to set device name.

- **Format definition**

Fields	Description
MSG_ID	0x05, set device name
PAR_LEN	name_length
Dev_name	Device name

- **Hex Command Data**

EA 06 07 51 75 69 6e 74 69 63 (Quintic)

- **Event generated**

See section 4.14 Set Device name event

4.6 Bond command

- **Description**

This command is used to bond the peer device.

- **Format definition**

Fields	Description
MSG_ID	0x06, bond command
PAR_LEN	0x06
bd_addr	6 bytes, Bluetooth device address

- **Event generated**

See section 4.15 Bond complete event

4.7 Central Update Parameter command

- **Description**

This command is used to send parameters update change by master.

- **Format definition**

Fields	Description
MSG_ID	0x07, update parameter
PAR_LEN	0x10
Result	Result of the connection parameters request, 0x0000: accept 0x0001: reject
Conn_intv_min	Default: 0x0018 (Range: 0x0006 to 0x0C80) – Minimum connection interval(0x0018 * 1.25 ms)
Conn_intv_max	Default: 0x0028 (Range: 0x0006 to 0x0C80) – Maximum connection interval(0x0028 * 1.25 ms)
Latency	Default: 0x0000
Cnnn_timeout	Default: 0x07D0 (Range: 0x000A to 0x0C80) - Connection supervision timeout(0x07D0 * 10 ms)
bd_addr	6 bytes, Bluetooth device address

- **Hex Command Data**

EA 08 10 01 00 18 00 28 00 00 00 D0 07 XX XX XX XX XX XX

- **Event generated**

See section 4.16 Central Update Parameter

4.8 Peripheral Update Parameter command

- **Description**

This function is used to change a set of new connection parameters.

- **Format definition**

Fields	Description
MSG_ID	0x08, update parameter
PAR_LEN	0x08
Conn_intv_min	Default: 0x0018 (Range: 0x0006 to 0x0C80) – Minimum connection interval(0x0018 * 1.25 ms)
Conn_intv_max	Default: 0x0028 (Range: 0x0006 to 0x0C80) – Maximum connection interval(0x0028 * 1.25 ms)
Latency	Default: 0x0000
Cnnn_timeout	Default: 0x07D0 (Range: 0x000A to 0x0C80) - Connection supervision timeout(0x07D0 * 10 ms)

- **Hex Command Data**

EA 09 08 18 00 28 00 00 00 D0 07

- **Event generated**

See section 4.17 Peripheral Update Parameter

4.9 Advertising status event

Fields	Description
MSG_ID	0x01, Advertising command complete event
PAR_LEN	0x01
status	1: started, 0: stopped

4.10 Scan result event

- **Description**

This event is used to report the scan result to host.

- **Format definition**

Fields	Description
MSG_ID	0x02, scan result event
PAR_LEN	0x07 + n
bd_addr_type	1 bytes, Bluetooth device address type
bd_addr	6 bytes, Bluetooth device address
dev_name	n byte, device name of the found device

4.11 Scan complete event

- **Description**

This event is used to report the scan finished event to host.

- **Format definition**

Fields	Description
MSG_ID	0x03, scan complete event
PAR_LEN	0x1
num	Total found device numbers

4.12 Connection complete event

- **Description**

This event is used to report the connection complete event.

- **Format definition**

Fields	Description
MSG_ID	0x04, Connection complete event
PAR_LEN	0x07
status	1 byte, Connection status (0x00 Success)
bd_addr	6 bytes, Bluetooth device address

4.13 Disconnection event

- **Description**

This event is used to report the connection status event.

- **Format definition**

Fields	Description
MSG_ID	0x05, Disconnection event
PAR_LEN	0x07
reason	1 byte, Disconnection reason
bd_addr	6 bytes, Bluetooth device address

4.14 Set Device name event

- **Description**

This event is used to inform the application that the set device name request has completed.

- **Format definition**

Fields	Description
MSG_ID	0x06, Set Device Name status
PAR_LEN	0x01
status	1 byte, status

4.15 Bond complete event

- **Description**

This event is used to report the bond complete event to host.

- **Format definition**

Fields	Description
MSG_ID	0x07, bond complete event
PAR_LEN	0x08
status	1 byte, status
bonded	1 byte, bonded
bd_addr	6 bytes, Bluetooth device address

4.16 Central Update Parameter event

- **Description**

This event is used to inform the application of the outcome of change connection parameter by master.

- **Format definition**

Fields	Description
MSG_ID	0x08, central update status
PAR_LEN	0x01
status	1 byte, status

4.17 Peripheral Update Parameter event

- **Description**

This event is used to inform the application of the outcome of the connection parameter update by slave.

- **Format definition**

Fields	Description
MSG_ID	0x0A, peripheral update status
PAR_LEN	0x01
status	1 byte, status

5. Profile Enable

5.1 Request

Format definition

Fields	Description	Byte
MSG_Type	0xEB, Data Request	1
MSG_ID	0x01, Enable Data Request	1
PAR_LEN	0x08	1
uuid	0x0000	2
bd_addr	Bluetooth device address	6

5.2 Indication

Format definition

Fields	Description	Byte
MSG_Type	0xEC, Data Indication	1
MSG_ID	0x01, Enable Data Indication	1
PAR_LEN	0x03	1
uuid	0x0000	2
status	0x00	1

5.3 UUID

UUID	Profile
0x1811	Alert Notification Status profile
0x180F	Battery Service Profile
0x1810	Blood Pressure Profile
0x1816	Cycling Speed and Cadence profile
0x180A	Device Information Service Profile
0x1802	Find Me Profile
0x1808	Glucose Profile
0x180D	Heart Rate Profile
0x1809	Health Thermometer Profile
0x180E	Phone Alert Status Profile
0x1803	Proximity Profile
0x1814	Running Speed and Cadence profile
0x1813	Scan Parameter Profile
0x1805	Time Profile

Example: (Heart Rate - bd_addr – 0x 11 22 33 44 55 66)

Enable Request(Hex): EB 01 08 0D 18 66 55 44 33 22 11

Enable Indication(Hex): EC 01 03 0D 18 00 (If the data is 0x00 think it is right, else is error.)

6. Alert Notification Status Profile

6.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)		bd_addr
Client	0x02(Read Characteristic Value)	0x09	0x1811	read_code(1) = 0x10(New Alert CFG.); = 0x11(Unread Alert Status CFG.).		Bluetooth device address
Client	0x03(Write Characteristic Value)	0x0B	0x1811	write_code	value	Bluetooth device address
				0x04 (Alert Notification Control Point)	value[0](Command ID)(NOTE1) value[1](Category ID)(NOTE2)	
				0x10(New Alert CFG.)	Stop(2) – 0x0000 , Notify(2) – 0x0001	
				0x11(Unread Alert Status CFG.)	Stop(2) – 0x0000 , Notify(2) – 0x0001	
Server	0x04(New Alert)	0x06 + info_str_len	0x1811	operation	value	NULL
				0x03(Update New Alert Char. value)	info_str_len(1) and cat_id(1) and nb_new_alert(1) and str_info(info_str_len)	
		0x05		0x04(Update Unread Alert Status Char. value)	value[0](Category ID) value[1](Number of alert)	

6.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Client	0x02(Write status)	0x03	0x1811	status(1)
Client	0x03(new alert cfg.)	0x04	0x1811	Stop(2) – 0x0000 or Notify(2) – 0x0001
Client	0x04(unread alert cfg.)	0x04	0x1811	Stop(2) – 0x0000 or Notify(2) – 0x0001
Client	0x05(new alert)	0x04	0x1811	cat_id(1) and nb_new_alert(1)
Client	0x06(unread alert status)	0x04	0x1811	cat_id(1) and nb_unread_alert(1)
Server	0x07(Update New Alert Char. value)	0x03	0x1811	status(1)
Server	0x08(Update Unread Alert Status Char. value)	0x03	0x1811	status(1)

NOTE1: Command ID = 0;(Enable New Incoming Alert Notification)
 = 1;(Enable Unread Category Status Notification)
 = 2;(Disable New Incoming Alert Notification)
 = 3;(Disable Unread Category Status Notification)
 = 4;(Notify New Incoming Alert immediately)
 = 5;(Notify Unread Category Status immediately)

NOTE2: Category ID = 0;(Simple Alert)
 = 1;(Email)
 = 2;(News Feed)
 = 3;(Incoming Call)
 = 4;(Missed Call)
 = 5;(SMS/MMS)
 = 6;(Voice Mail)
 = 7;(Schedule)
 = 8;(High Priority Alert)
 = 9;(Instant Message)
 = 255;(All supported category)

7. Battery Profile

7.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)		bd_addr
Client	0x02(Read Characteristic Value)	0x0A	0x180F	0x00(1)(Battery Level) 0x11(1)(Battery Level CFG.)		Bluetooth device address
Client	0x03(Notify)	0x0A	0x180F	Stop(2) – 0x0000 or Notify(2) – 0x0001		Bluetooth device address
Server	0x04(Battery Level)	0x0A	0x180F	Battery Instances (1)	Battery Level(1)	Bluetooth device address

7.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Client	0x02(Battery Level)	0x03	0x180F	Batter Level(1)
Client	0x03(Battery CFG.)	0x04	0x180F	Stop(2) – 0x0000 or Notify(2) – 0x0001
Client	0x04(Write RSP)	0x03	0x180F	Status(1)
Server	0x05(Battery Level Update Confirm)	0x03	0x180F	Status(1)

8. Blood Pressure Profile

8.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)				bd_addr
Collector	0x02(Read Char Code)	0x09	0x1810	char code (1) = 0x02(Blood Pressure Feature); = 0x10(Blood Pressure Measurement client config); = 0x11(Intermediate Cuff pressure client config).				Bluetooth device address
Collector	0x03(Configuration characteristics)	0x0B	0x1810	char code (1) = 0x01(Blood Pressure Measurement); = 0x02(Intermediate Cuff Pressure Measurement);	Stop(2) – 0x0000 , Notify(2) – 0x0001 or Indicate(2) – 0x0002			Bluetooth device address
Sensor	0x04(Send Pressure)	0x04 + Len_1 + Len_2 + Len_3 + Len_4 + Len_5	0x1810	flag_interm (1)	flag(1)	flag		
						Bit	Name	Data
						3	User ID	0 Len_1 = 0
								1 Len_1 = 1
						0	Pressure in pascal	0 Len_2 = 6
								1 Len_2 = 6
						2	Pulse Rate present	0 Len_3 = 0
								1 Len_3 = 2
						4	Measurement Status present	0 Len_4 = 0
								1 Len_4 = 2
						1	Time Stamp present	0 Len_5 = 0
								1 Len_5 = 7

8.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	
Collector	0x02(Read Char RSP)	0x05	0x1810	status(1)	Holder of retrieved data(2)
Collector	0x03(Write Char RSP)	0x03	0x1810	status(1)	
Collector	0x04(BP Measurement)	Show 11.1 (Sensor) 0x04(Send Pressure)			
Sensor	0x05(Configure Indication)	0x04	0x1810	Stop(2) – 0x0000 , Notify(2) – 0x0001 or Indicate(2) – 0x0002	
Sensor	0x06(Send Meas confirm)	0x03	0x1810	status(1)	

NOTE: flag_interm = 0 (Stable Blood Pressure Measurement);

flag = 1 (Intermediate Cuff Pressure Measurement).

9. Cycling Speed and Cadence Profile

9.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)		bd_addr
Client	0x02(Read Characteristic Value)	0x09	0x1816	read_code(1) = 0x01(CSC Feature); = 0x02(Sensor Location); = 0x10(CSC Measurement CFG.); = 0x11(SC Control Point CFG.);		Bluetooth device address
Client	0x03(Configure)	0x0B	0x1816	desc_code	ntfind_cfg	Bluetooth device address
				0x10(CSC Measurement CFG.)	Stop(2) – 0x0000 or Notify(2) – 0x0001	
				0x11(SC Control Point CFG.)	Stop(2) – 0x0000 or Indicate(2) – 0x0002	
Client	0x04(SC Control Point)	0x0D	0x1816	op_code	value	Bluetooth device address
				0x01(Set Cumulative)	value(1) = 0x00; value(2) = 0x00; value(3) = 0x00; value(4) = 0x00.	
				0x03(Update Sensor Location)	value(1) - Sensor Location; value(2) = 0x00; value(3) = 0x00; value(4) = 0x00.	
				0x04(Request Supported Sensor Locations)	value(4) - Cumulative value	
Server	0x05(Send the CSCP Sensor Measurement)	0x0B	0x1816	Flags(1) cumul_crank_rev(2) last_crank_evt_time(2) last_wheel_evt_time(2) wheel_rev(2)		NULL
Server	0x06(Control Point Response)	0x07	0x1816	op_code	value	NULL
				0x01(Set Cumulative Value)	cumul_wheel_rev(4)	

9.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Client	0x02(read status)	0x03	0x1816	status(1)
Client	0x03(SC Control Point CFG.)	0x04	0x1816	Stop(2) – 0x0000 or Indication(2) – 0x0002
Client	0x04(CSC Measurement CFG.)	0x04	0x1816	Stop(2) – 0x0000 or Notify(2) – 0x0001
Client	0x05(Indicated SC Control Point)	0x06	0x1816	req_op_code(1) and resp_value(1) and supp_loc(2)
Client	0x06(Notify CSC meas)	0x0D	0x1816	flags(1) and cumul_crank_rev(2) and last_crank_evt_time(2) and last_wheel_evt_time(2) and cumul_wheel_rev(4)
Client	0x07(Write CFG. Status)	0x03	0x1816	status(1)
Client	0x08(Write Control Point CFG. Status)	0x03	0x1816	status(1)
Client	0x09(Read CSC Feature)	0x04	0x1816	sensor_feat(2)
Client	0x0A(Read Sensor Location)	0x03	0x1816	sensor_loc (1)
Server	0x0B(Send CSC Measurement Status)	0x03	0x1816	status(1)

10. Device Information Service Profile

10.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	bd_addr
Client	0x02(Read Characteristic Value)	0x09	0x180A	char_code(1)	Bluetooth device address
Server	0x03(Write Characteristic Value)	0x03 + val_length	0x180A	char_code(1) + val	NULL

NOTE: char_code = 0x00 (Manufacturer Name);
char_code = 0x01 (Model Number String);
char_code = 0x02 (Serial Number String);
char_code = 0x03 (Hardware Revision String);
char_code = 0x04 (Firmware Revision String);
char_code = 0x05 (Software Revision String);
char_code = 0x06 (System ID);
char_code = 0x07 (IEEE);
char_code = 0x08 (PnP ID).

10.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	
Client	0x02(Characteristic Value)	0x03 + data_length	0x180A	Status(1)	data[data_length]

11. Find Me Profile

11.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	bd_addr
Locator	0x02(Write Alert Level)	0x09	0x1802	None(1) – 0x00, Mild(1) - 0x01,or High(1) - 0x02	Bluetooth device address

11.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	
Target	0x02(Alert Level)	0x03	0x1802	None(1) – 0x00, Mild(1) - 0x01,or High(1) - 0x02	

12. Glucose Profile

12.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	bd_addr(Byte)						
Client	0x02(Register)	0x09	0x1808	meas_ctx_en(1)		Bluetooth device address(6)				
Client	0x03(Read Features)	0x08	0x1808	Bluetooth device address(6)						
				operator	seq_num (MIN)	seq_num (MAX)				
Client	0x04(Report Stored Records)	0x0D	0x1808	(1)	(2)	(2)	Bluetooth device address(6)			
Client	0x05>Delete Stored Records)	0x0D	0x1808				Bluetooth device address(6)			
Client	0x06(Abort Operation)	0x0D	0x1808				Bluetooth device address(6)			
Client	0x07(Report Number of Stored Records)	0x0D	0x1808				Bluetooth device address(6)			
Service	0x08(Send Glucose measurement without context information)	0x0C + Len_0 + Len_1 + Len_3	0x1808	Flags(1)	Sequence Number(2)	Year(2)+ Month(1)+ Day(1)+ Hour(1)+ Min(1)+ Sec(1)	Flags			
							Bit	Name	Value	Data_len
							0	Time Offset	0	Len_0(0)
									1	Len_0(2)
							1	Concentration, type and location	0	Len_1(0)
									1	Len_1(4)
3	Sensor Status Annunciation	0	Len_3(0)							
		1	Len_3(2)							
Service	0x09(Send Glucose measurement with context information)	See Table 1	0x1808	See Table 1						

12.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Client	0x02(register confirmation)	0x03	0x1808	status(1)
Client	0x03(read features response)	0x05	0x1808	Data[0] – status Data[2] << 8 Data[1] - Features
Client	0x04(Glucose measurement without context information)		0x1808	See Data Request MSG_ID – 0x08
Client	0x05(Glucose measurement with context information)		0x1808	See Data Request MSG_ID – 0x09
Client	0x06(RACP response)	0x05	0x1808	op_code
				Data(Byte)
				Number of stored records response (0x05)
Service	0x07(cfg)	0x03	0x1808	Response Code (0x06)
				op_code_req (1) + 0x00
				evt_cfg(1)

Table 1

PAR_LEN	Data(Byte)												
0x0C + Len_0 + Len_1 + Len_3 + Len_4 + Len_5 + Len_6 + Len_7 + Len_8 + Len_9 + Len_10	Flags (1)	Sequence Number(2)	Year(2)+ Month(1)+ Day(1)+ Hour(1)+ Min(1)+ Sec(1)	Flags(1)				flag (1)	flag				
				Bit	Name	Value	Data_len		Bit	Name	Value	Data_len	
				0	Time Offset	0	Len_0(0)		7	Extended Flags	0	Len_4(0)	
						1	Len_0(2)				1	Len_4(1)	
				1	Concentration, type and location	0	Len_1(0)		0	Carbohydrate ID And Carbohydrate Present	0	Len_5(0)	
						1	Len_1(4)				1	Len_5(3)	
				3	Sensor Status Annunciation	0	Len_3(0)		1	Meal Present	0	Len_6(0)	
											1	Len_6(1)	
									2	Tester-Health Present	0	Len_7(0)	
											1	Len_7(2)	
				3	Exercise Duration & Exercise Intensity Present	0	Len_8(0)						
						1	Len_8(3)						
				4	Medication ID And Medication Present	0	Len_9(0)						
						1	Len_9(3)						

									5	Medication Value Units	0	Kilograms
											1	Liters
									6	HbA1c Present	0	Len_10(0)
						1	Len_3(2)				1	Len_10(2)

13. Heart Rate Profile

13.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	bd_addr
Collector	0x02(Read Body Sensor Location)	0x08	0x180D	NULL	Bluetooth device address
Collector	0x03(Configure Measurement)	0x0A	0x180D	Stop(2) – 0x0000 , Notify(2) – 0x0001	Bluetooth device address
Sensor	0x04(Heart Rate measurement value)	0x04	0x180D	Data(2)	NULL

13.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Collector	0x02(Read Body Sensor Location rsp)	0x04	0x180D	<div> Status(1) – (0x00 is OK) </div> <div> value(1) = 0x00 (Other); = 0x01 (Chest); = 0x02 (Wrist); = 0x03 (Finger); = 0x04 (Hand); = 0x05 (Ear Lobe); = 0x06 (Foot). </div>
Collector	0x03(WR RSP.)	0x03	0x180D	Status(1) – (0x00 is OK)
Collector	0x04(Heart Rate value)	0x04	0x180D	Heart rate value(2)
Sensor	0x05(configure)	0x04	0x180D	Stop(2) – 0x0000 or Notify(2) – 0x0001

14. Health Thermometer Profile

14.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	bd_addr
Collector	0x02(Temperature Measurement)	0x0A	0x1809	Stop(2) – 0x0000 or Indicate(2) – 0x0002	Bluetooth device address
Collector	0x03(Intermediate Temperature cfg.)	0x0A	0x1809	Stop(2) – 0x0000 or Notify(2) – 0x0001	Bluetooth device address
Collector	0x04(Measurement Interval cfg.)	0x0A	0x1809	Stop(2) – 0x0000 or Indicate(2) – 0x0002	Bluetooth device address
Collector	0x05(read measurement interval)	0x08	0x1809	NULL	Bluetooth device address
Thermometer	0x06(Send Measurement Interval value)	0x04	0x1809	Measurement Interval value(2)	NULL
Thermometer	0x07(Send Temperature Measurement value)	0x10	0x1809	Temperature Measurement Value Structure(0x0E)	NULL

14.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Collector	0x02(Read Char RSP)	0x05	0x1809	Data[0] (0x00 Success) Data[2] << 8 Data[1](Interval)
Collector	0x03(WR RSP.)	0x03	0x1809	Status(1) – (0x00 is OK)
Collector	0x04(Temperature Measurement)	0x08	0x1809	Data[0](0x00-Intermediate 0x01-Stable); Data[1](0x00-Celsius 0x01-Fahrenheit); Data [5] << 24 Data [4] << 16 Data [3] << 8 Data [2] (Temperature Measurement Value)
Collector	0x05(Measurement Interval)	0x04	0x1809	Measurement Interval Value(2)
Thermometer	0x06(Thermometer Measurement Interval)	0x04	0x1809	Thermometer Measurement Interval Value(2)
Thermometer	0x07(Thermometer cfg. indication)	0x04	0x1809	Stop(2) – 0x0000 , Notify(2) – 0x0001 or Indicate(2) – 0x0002

15. Phone Alert Status Profile

15.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)		bd_addr
Client	0x02(Read Characteristic Value)	0x09	0x180E	read_code(1) = 0x00(Read PAS Alert Status); = 0x01(Read PAS Ringer Setting); = 0x10(Read PAS Alert Status CFG.); = 0x11(Read Ringer Setting CFG.);		Bluetooth device address
Client	0x03(Write)	0x0B	0x180E	write_code	value	Bluetooth device address
				0x02(Ringer Control Point)	value[0] = 0x01 (Silent Mode) = 0x02 (Mute Once) = 0x03(Cancel Silent Mode); Value[1] = 0x00.	
				0x10(Alert Status CFG.)	Stop(2) – 0x0000 or Notify(2) – 0x0001	
				0x11(Ringer Setting CFG.)	Stop(2) – 0x0000 or Notify(2) – 0x0001	
Server	0x04(Update value)	0x04	0x180E	operation	value	NULL
				0x03(Update Alert Status Char. value)	value(1)	
				0x04(Update Ringer Setting Char. value)	value(1)	

15.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Client	0x02(read status)	0x03	0x180E	status(1)
Client	0x03(Write status)	0x03	0x180E	status(1)
Client	0x04(Read Alert Status)	0x03	0x180E	alert_status(1)
Client	0x05(Read Ringer Setting)	0x04	0x180E	ringer_setting(2)
Client	0x06(Read Alert Status CFG.)	0x04	0x180E	alert_status_ntf_cfg(2)
Client	0x07(Read Ringer Setting CFG.)	0x04	0x180E	ringer_setting_ntf_cfg(2)
Server	0x08(Update Alert Status status)	0x03	0x180E	status(1)
Server	0x09(Update Ringer Setting status)	0x03	0x180E	status(1)

16. Proximity Profile

16.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	bd_addr
Monitor	0x02(Read TX Power)	0x08	0x1804	NULL	Bluetooth device address
Monitor	0x03(Write IAS data)	0x09	0x1804	None(1) – 0x00, Mild(1) - 0x01,or High(1) - 0x02	Bluetooth device address
Monitor	0x04(Write LLS data)	0x09	0x1803	None(1) – 0x00, Mild(1) - 0x01,or High(1) - 0x02	Bluetooth device address

16.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Monitor	0x02(TX Power Level)	0x03	0x1804	value(1)
Reporter	0x03(Alert Level)	0x03	0x1804	None(1) – 0x00, Mild(1) - 0x01,or High(1) - 0x02
Reporter	0x04(LLS)	0x03	0x1803	None(1) – 0x00, Mild(1) - 0x01,or High(1) - 0x02

17. Running Speed and Cadence Profile

17.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)		bd_addr
Client	0x02(Read Characteristic Value)	0x09	0x1814	read_code(1) = 0x01(Read RSC Feature); = 0x02(Read Sensor Location); = 0x10(Read RSC Measurement CFG.); = 0x11(Read SC Control Point CFG.);		Bluetooth device address
Client	0x03(CFG.)	0x0B	0x1814	desc_code	ntfind_cfg	Bluetooth device address
				0x10(RSC Measurement CFG.)	Stop(2) – 0x0000 or Notify(2) – 0x0001	
				0x11(SC Control Point CFG.)	Stop(2) – 0x0000 or Indication(2) – 0x0002	
Client	0x04(Write)	0x0D	0x1814	op_code	value	Bluetooth device address
				0x01(Set Cumulative Value)	cumul_val(4)	
				0x03(Update Sensor Location)	Value[0]= sensor_loc(1); Value[1] = 0x00; Value[2] = 0x00; Value[3] = 0x00.	
				0x04(Request Supported Sensor Locations)	Value[0] = 0x00; Value[1] = 0x00; Value[2] = 0x00; Value[3] = 0x00.	
Server	0x05(Send the RSC Sensor Measurement value)	0x0C	0x1814	flags(1) and inst_cad(1) and inst_speed(2) and inst_stride_len(2) and total_dist(4)		NULL

17.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Client	0x02(Read Status)	0x03	0x1814	status(1)
Client	0x03(Read CFG Status)	0x03	0x1814	status(1)
Client	0x04(Write CFG. Status)	0x03	0x1814	status(1)
Client	0x05(Write PT CFG. Status)	0x03	0x1814	status(1)
Client	0x06(Notified RSC Measurement)	0x0C	0x1814	flags(1) and inst_cad(1) and inst_speed(2) and inst_stride_len(2) and total_dist(4)
Client	0x07(RSC Feature)	0x04	0x1814	sensor_feat(2)
Client	0x08(Sensor Location)	0x03	0x1814	sensor_loc(1)
Client	0x09(SC Control Point)	0x06	0x1814	req_op_code(1) and resp_value(1) and supp_loc(2)
Client	0x0A(RSC Measurement CFG.)	0x04	0x1814	Stop(2) – 0x0000 or Notify(2) – 0x0001
Client	0x0B(SC Control Point CFG.)	0x04	0x1814	Stop(2) – 0x0000 or Indication(2) – 0x0001
Server	0x0C(Send RSC Measurement Status)	0x03	0x1814	status(1)
Server	0x0D(Set Cumulative Value Status)	0x03	0x1814	status(1)
Server	0x0E(Update Sensor Location Status)	0x03	0x1814	status(1)
Server	0x0F(Supported Sensor Locations Status)	0x03	0x1814	status(1)

18. Scan Parameter Profile

18.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	bd_addr
Client	0x02(Read Scan Refresh Cfg.)	0x08	0x1813	NULL	Bluetooth device address
Client	0x03(Send Scan Interval and Win)	0x0C	0x1813	scan_intv(2) scan_window(2)	Bluetooth device address
Client	0x04(Configure Scan Refresh)	0x0A	0x1813	Stop(2) – 0x0000 , Notify(2) – 0x0001	Bluetooth device address
Server	0x05(Send the scan refresh value)	0x02	0x1813	NULL	NULL

18.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Client	0x02(WR Status)	0x03	0x1813	Status(1) – 0x00 is OK.
Client	0x03(Scan Refresh Cfg)	0x04	0x1813	Stop(2) – 0x0000 , Notify(2) – 0x0001
Server	0x04(Scan Interval and Win)	0x06	0x1813	scan_intv(2) scan_window(2)
Server	0x05(Configure Scan Refresh)	0x04	0x1813	Stop(2) – 0x0000 , Notify(2) – 0x0001

19. Time Profile

19.1 Data Request

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)	bd_addr
Client	0x02(Read Characteristic)	0x09	0x1805	char_code(1)	Bluetooth device address
Client	0x03(Configuring the Current Time)	0x0A	0x1805	Stop(2) – 0x0000 , Notify(2) – 0x0001	Bluetooth device address
Client	0x04(Writing Time Control Point)	0x09	0x1805	Value(1)	Bluetooth device address
Server	0x05(update current time)	0x0A	0x1805	current_time(8)	NULL

NOTE: char_code = 0 (CTS Current Time);
char_code = 1 (CTS Local Time Info);
char_code = 2 (CTS Reference Time Info);
char_code = 3 (CTS Current Time Client Cfg. Desc);
char_code = 4 (NDCS Time With DST);
char_code = 5 (RTUS Time Update State);

19.2 Data Indication

Role	MSG_ID	PAR_LEN	UUID	Data(Byte)
Client	0x02(WR Status)	0x03	0x1805	Status(1) – 0x00 is OK.
Client	0x03(Current Time value)	0x0A	0x1805	day_date_time (8)
Client	0x04(Configuration Current Time)	0x04	0x1805	Stop(2) – 0x0000 , Notify(2) – 0x0001
Client	0x05(Local Time Info)	0x04	0x1805	lti_val(2)
Client	0x06(Reference Time Info)	0x06	0x1805	rli_val(4)
Client	0x07(Time With DST)	0x03	0x1805	dst_val(1)
Client	0x08(Time Update State)	0x04	0x1805	ctc_val(2)
Server	0x09(Current time Cfg)	0x04	0x1805	Stop(2) – 0x0000 , Notify(2) – 0x0001
Server	0x0a(Update Control Point value)	0x03	0x1805	ucp_val(1)

20. Demo Example

The chapter introduces how to study EACI on the EVB.

20.1 Download - LPC1768

You can get the example code in our SDK. Please choose a project file in the path “\Quintic Corporation\QBlue-x.x.x\Projects\BLE\prj_eaci_host\keil”.

You can download it with J-LINK. You must wiring referring to the following figure6(connect J4-11 to P2-7, J4-10 to P2-9).

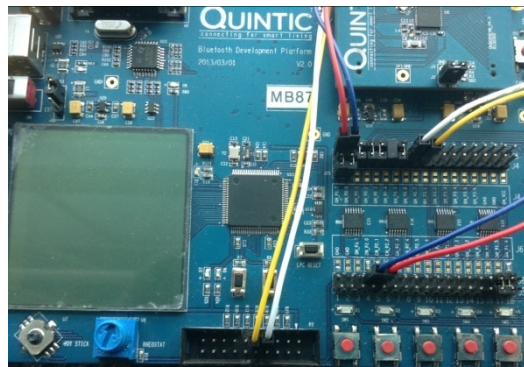


Figure 6: Download-LPC1768

20.2 Download – QN90xx

You can get the example code in our SDK. Please choose a project file in the path “\Quintic Corporation\QBlue-x.x.x\Projects\BLE\prj_eaci_controller\keil”.

You can download it with J-LINK. You must wiring referring to the following figure7(connect J4-2 to J6-18).

Note: J6_PIN: QN_P1.6, QN_P2.0 and QN_P2.1 need to resistance, the resistance value is 0 ohm.

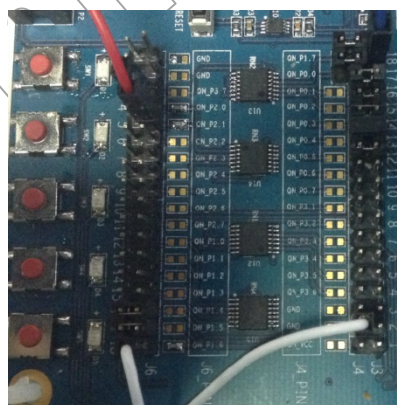


Figure 7: Download – QN90xx

20.3 Data Request

You can send and receive data through Hyper Terminal.

Step 1:

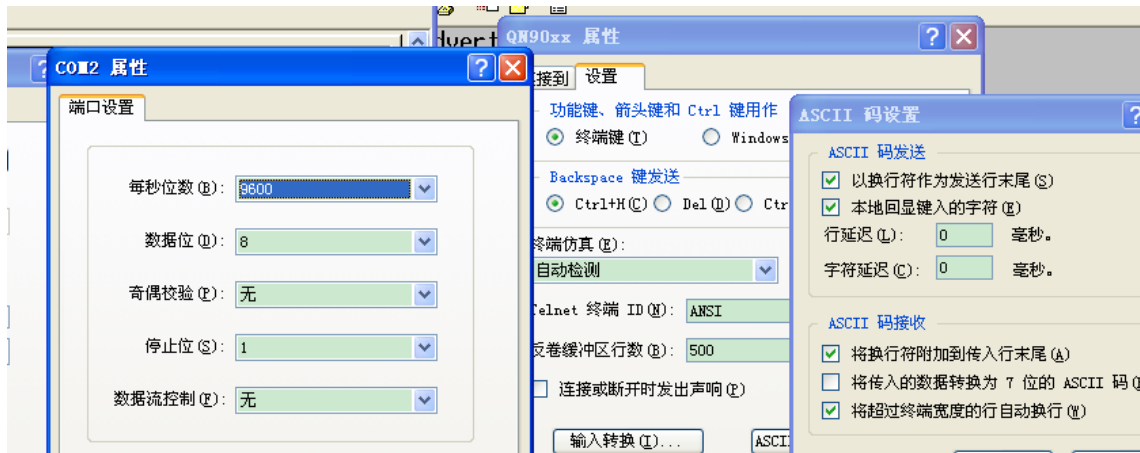


Figure 8: Hyper Terminal Configuration

Step 2:

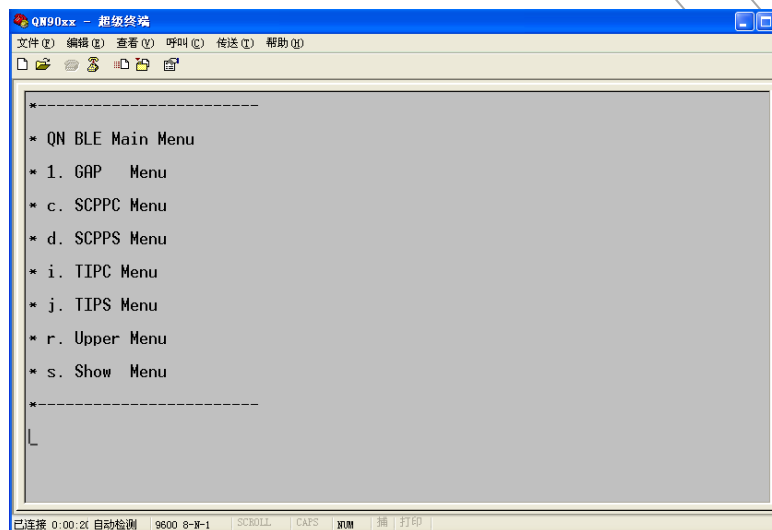


Figure 9: Idle

Step 3:

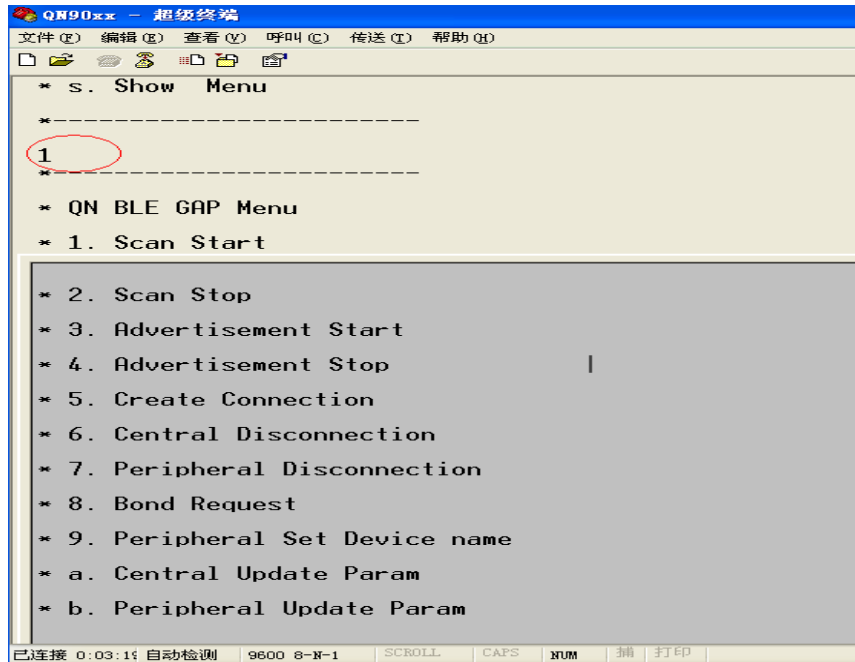


Figure 10: Input “1”

Step 4:

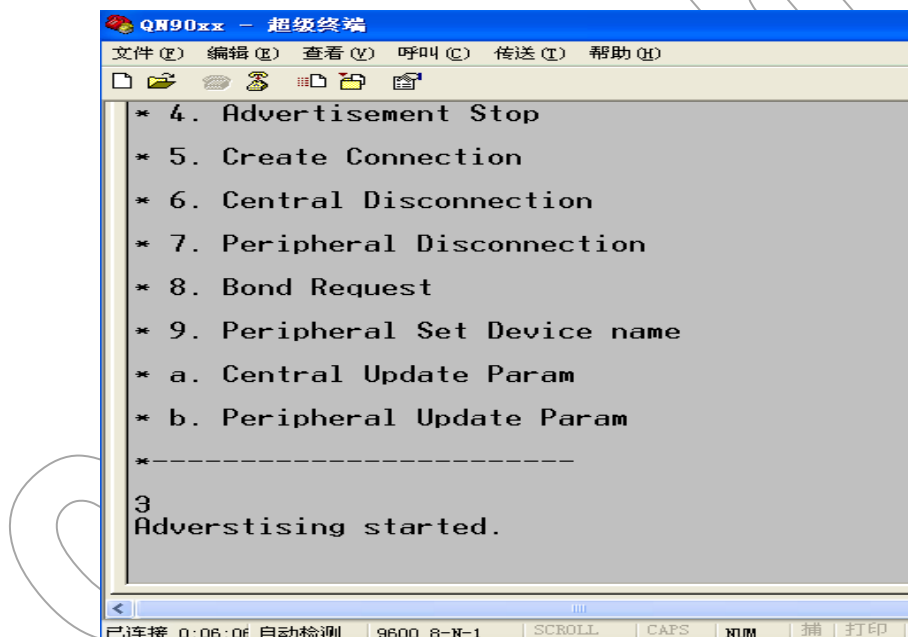


Figure11: Input “3” - Advertising

20.4 User Configuration

In order to simplify the development of EACI, Quintic provides a user configuration file (usr_config.h) to customize QN90xx’s application setting.

UART: If you want to use UART, the macro ‘CFG_HCI_UART’ and ‘CFG_EACI_GPIO_WAKEUP_QN_MCU’ should be defined.

Copyright ©2012-2014 by Quintic Corporation

Confidential Information contained herein is covered under Non-Disclosure Agreement (NDA) Page 34

SPI: When communicates by SPI, the macro 'CFG_HCI_SPI' and 'CFG_HCI_SPI_WR_CTRL_PIN' should be defined.

20.5 SPI Configuration

When communicates by SPI, please wiring as figure 12 shows:

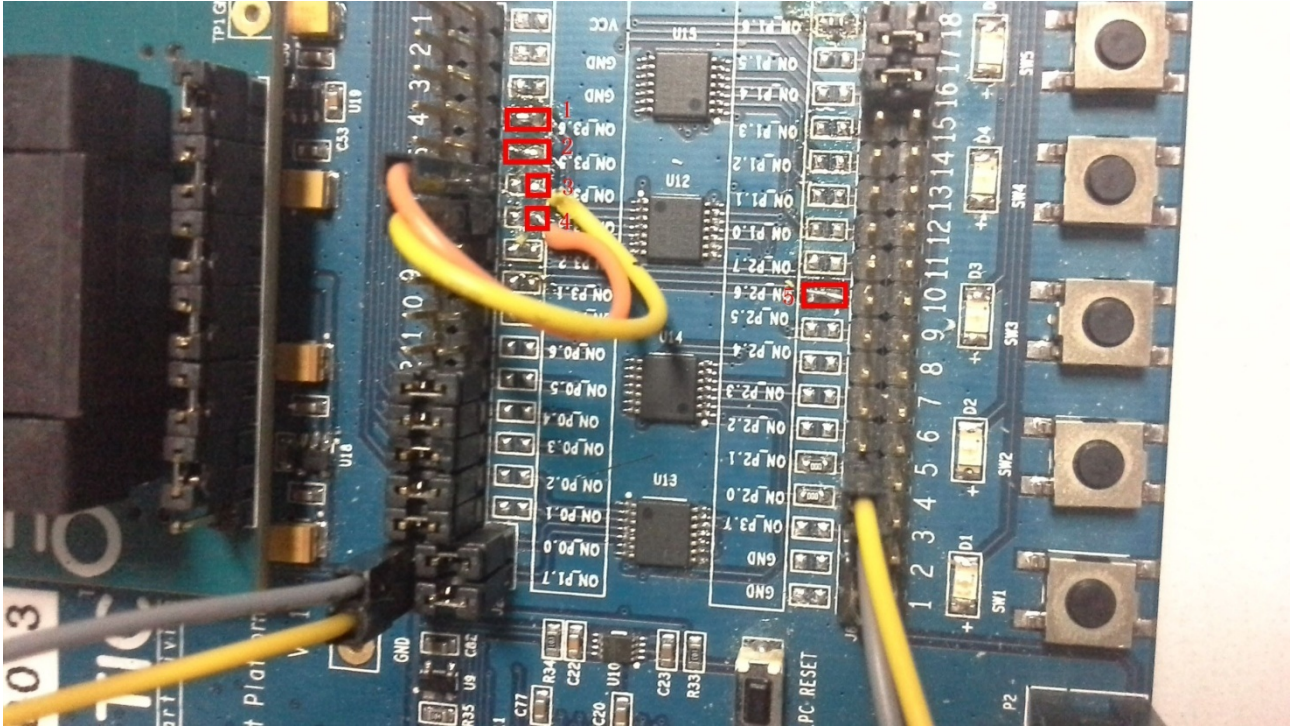


Figure 12: SPI wiring

The pins with red line around them should be connected.

If there is no pin named "QN_P3.0" on the motherboard, please connect the pins as follows:

- Pin 1: LPC_SPI0-CS should be connected to J4-4:QN_P3.6_SPI0-CS
- Pin 2: LPC_SPI0_CLK should be connected to J4-5:QN_P3.5_SPI0_CLK
- Pin 3: LPC_SPI0_MOSI should be connected to J4-6:QN_P3.3
- Pin 4: LPC_SPI0_MISO should be connected to J4-7:QN_P3.4
- Pin 5: LPC_P0.3 should be connected to J6-10: QN_P2.6

Otherwise, please connect the pins as follows:

- Pin 1: LPC_SPI0-CS should be connected to J4-4:QN_P3.5_SPI0-CS
- Pin 2: LPC_SPI0_CLK should be connected to J4-5:QN_P3.4_SPI0_CLK
- Pin 3: LPC_SPI0_MOSI should be connected to J4-6:QN_P3.2
- Pin 4: LPC_SPI0_MISO should be connected to J4-7:QN_P3.3
- Pin 5: LPC_P0.3 should be connected to J6-10: QN_P2.6

Release History

REVISION	CHANGE DESCRIPTION	DATE
0.1	Initial release	2013-05-17
0.2	Update BLPS and GPIO command description	2013-05-29
0.3	Support B0 version	2013-07-11
0.4	Support AN,CSC,PAS and RSC Profiles	2013-10-10
0.5	Support B1 version	2013-11-08
0.6	Add SPI Interface and Support B2 version	2014-3-19

Confidential