AN12307

QN908x USB Interface Certification Guide

Rev. 1 — 5 March 2019 Application note

Document information

Info	Content
Keywords	QN908x, USB2.0 IF certification, Power consumption, Suspend/Resume
Abstract	This application note describes how to use the QN908x Evaluation boards and SDK for USB2.0 IF certification, including hardware reworking, software customization.

Revision history

Rev	Date	Description
0	2018/12/05	QN9080 USB2.0 IF certification application note only contains modify software, hardware adjustment and how to use USB20CV to test.
1	2019/5/03	Changed the Application Note name USB 2.0 IF certification use QN9080 DK to QN908x USB Interface Certification Guide

Contact information

For more information, please visit: http://www.nxp.com

1. Introduction

The QN908x is an ultra-low-power Bluetooth LE SoC with USB 2.0 full speed device interface. The USB interface has passed certification test with QN9080DK.

For ensuring compliance, there are always requirements to certify USB interface on product. The content below aims to ease the setup of the test and speed up the certification process.

2. Test Setup

1. Hardware setup

Target board: QN9080DK is the target board to pass the USB certification.

USB connector: The connector J1 is connecting with the USB interface of QN9080 and supply power to the DK.

Test PC: Desktop computer with the PCIe USB extension board

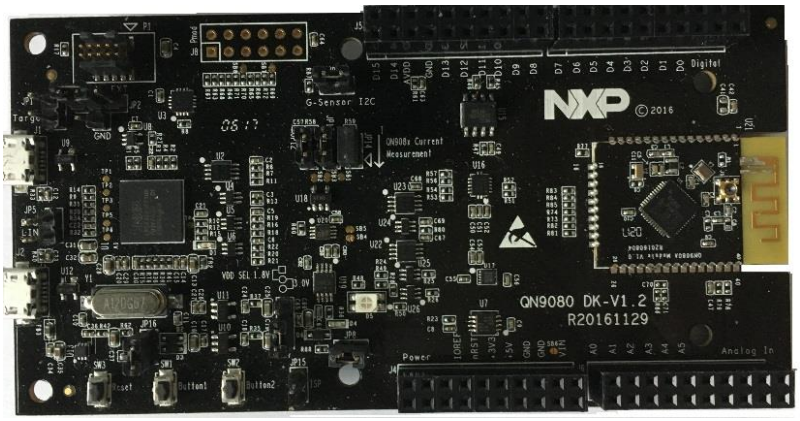

Fig 1. QN9080 DK

2. Software setup

   Test project: The SDK project usb_suspend_resume_device_hid_mouse in the SDK folder.

   Test utility: The utilities specified by USB.org, such as USB2.0 Command Verifier.

3. Customization for Passing the Suspend State Current Test

3.1 Hardware Reworking

The power consumption of QN9080DK under suspend state is higher than spec. In order to reduce the power consumption, rework the board as following:

1. The onboard current sensing circuit has to be turned off by pull up pin1 on U19. Please refer Fig 2 for details:

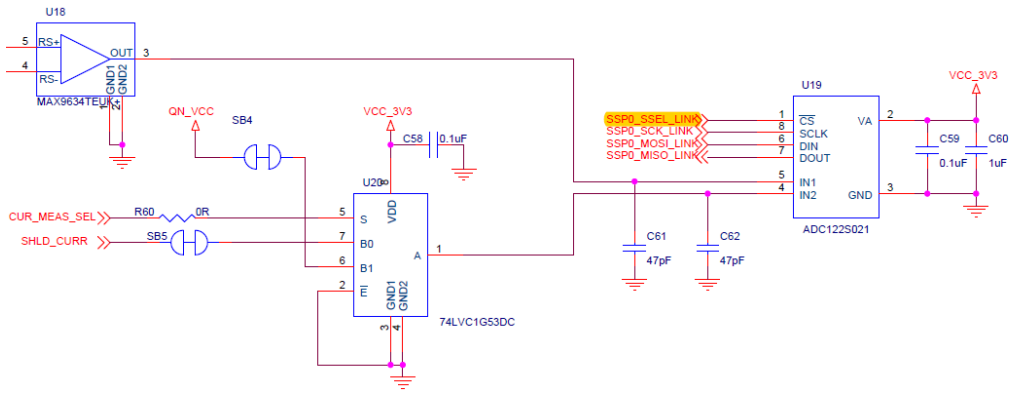

Fig 2. QN9080 DK current sensing circuit

2. Remove the resistor R36 to eliminate the current consumed by D4, refer to Fig 3 for details:

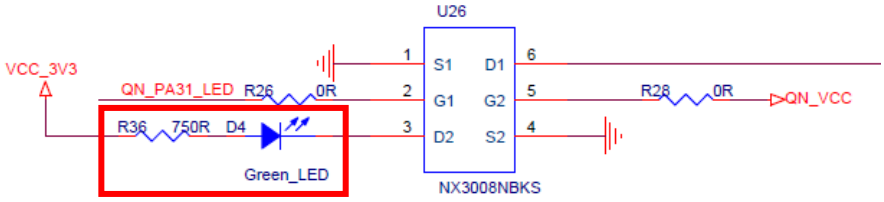

Fig 3. QN9080 DK power indicating circuit

3.2 Software Customization

1. Add necessary initialization in main.c

As illustrate in Fig 4 below, the initialization in the demo project is not enough for the power suspend test.

```
759
760 #if defined(__CC_ARM) || defined(__GNUC__)
761 int main(void)
762 #else
763 void main(void)
764 #endif
765 {
766
767     BOARD_BootClockRUN();
768     BOARD_InitDebugConsole();
769     BOARD_InitPins();
```

Fig 4. Initialization in main.c

Fig 4 project initialization

1) Enable DCDC to lower the power consumption of QN9080 by add the sentence below:

POWER_EnableDCDC(gDCDC_Mode);

Note: the macro gDCDC_Mode needs to be set to 1 in board.h

2) Set GPIO state in consistent with button:

BOARD_InitButtons();

3) Set triple-color LED to be off by default:

BOARD_InitLEDs();

4. Customization for fixing the USB HID descriptor failure

The USB HID descriptor is the information of USB HID devices. One of the tasks to be performed by the system device enumeration is to obtain the relevant information about the settings and then the devices can be identified and used by the system.

In project usb_suspend_resume_device_hid_mouse, the getting descriptor error reported when testing with the USB2.0 IF certification command verifier test. The prototype of the function is:

```
usb_status_t USB_DeviceGetHidDescriptor(usb_device_handle handle,
usb_device_get_hid_descriptor_struct_t *hidDescriptor)
{
   return kStatus_USB_InvalidRequest;
}
```

Change the function as the following to fix the issue：

```
usb_status_t USB_DeviceGetHidDescriptor(usb_device_handle handle,
usb_device_get_hid_descriptor_struct_t *hidDescriptor)
{
     if (USB_HID_MOUSE_INTERFACE_INDEX == hidDescriptor->interfaceNumber)
```

```
        {
            hidDescriptor->buffer =
                    &g_UsbDeviceConfigurationDescriptor[USB_DESCRIPTOR_LENGTH_C
                    ONFIGURE + USB_DESCRIPTOR_LENGTH_INTERFACE];
            hidDescriptor->length = USB_DESCRIPTOR_LENGTH_HID;
        }
    else
    {
        return kStatus_USB_InvalidRequest;
    }
    return kStatus_USB_Success;
}
```

5. Test Procedure with USB2.0 Command Verifier

USB2.0 Command Verifier(USB20CV) is the compliance test tool which evaluates High, Full and Low-speed USB devices for conformance to the USB Device Framework (Chapter 9), Hub device class (Chapter 11), HID class, and OTG specifications.
All USB peripherals are required to pass the Device Framework tests in order to gain certification. The other tests are mandatory for certification when supported.

5.1 Download USB20CV

User can use the USB20CV to do verification test in lab before going lab for certification test. The tool can be downloaded from the link http://www.usb.org/developers/tools/usb20_tools/.

5.2 Test with USB20CV

1. Select appropriate host controller and click Continue button.

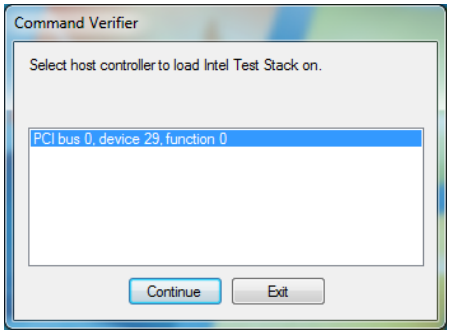

Fig 5.

2. Click the run button to get the test started.

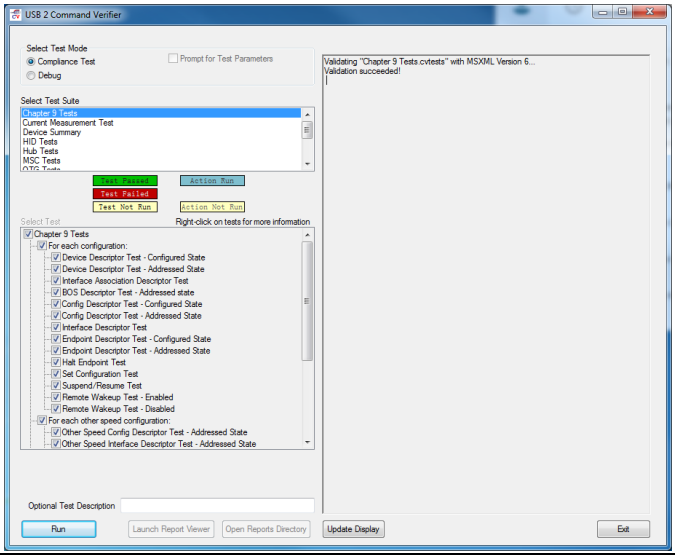

Fig 6. USB command verifier

3. Choose the QN9080 device(VID = 1fc9 and PID = 0091) and click OK to get the test started.

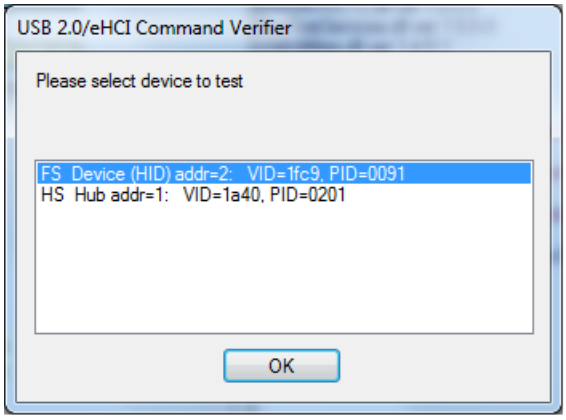

Fig 7. select device

4. If the suspend/resume test is conducted, a dialog as Fig 8 will pop up. Press button 2 on QN9080 DK board, the test will continue.

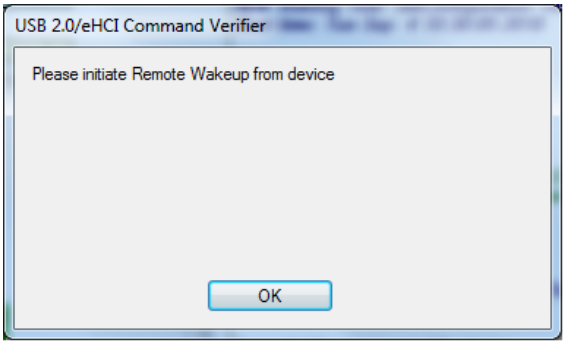

Fig 8. suspend/resume

5. If remote wakeup test is conducted, a dialog as Fig 9 will pop up. Press OK button the test will continue.

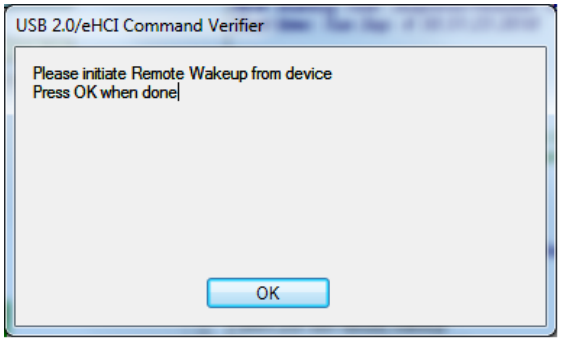

Fig 9. remote wakeup test

6. After all tests pass, the test result is displayed as in the Fig 10.

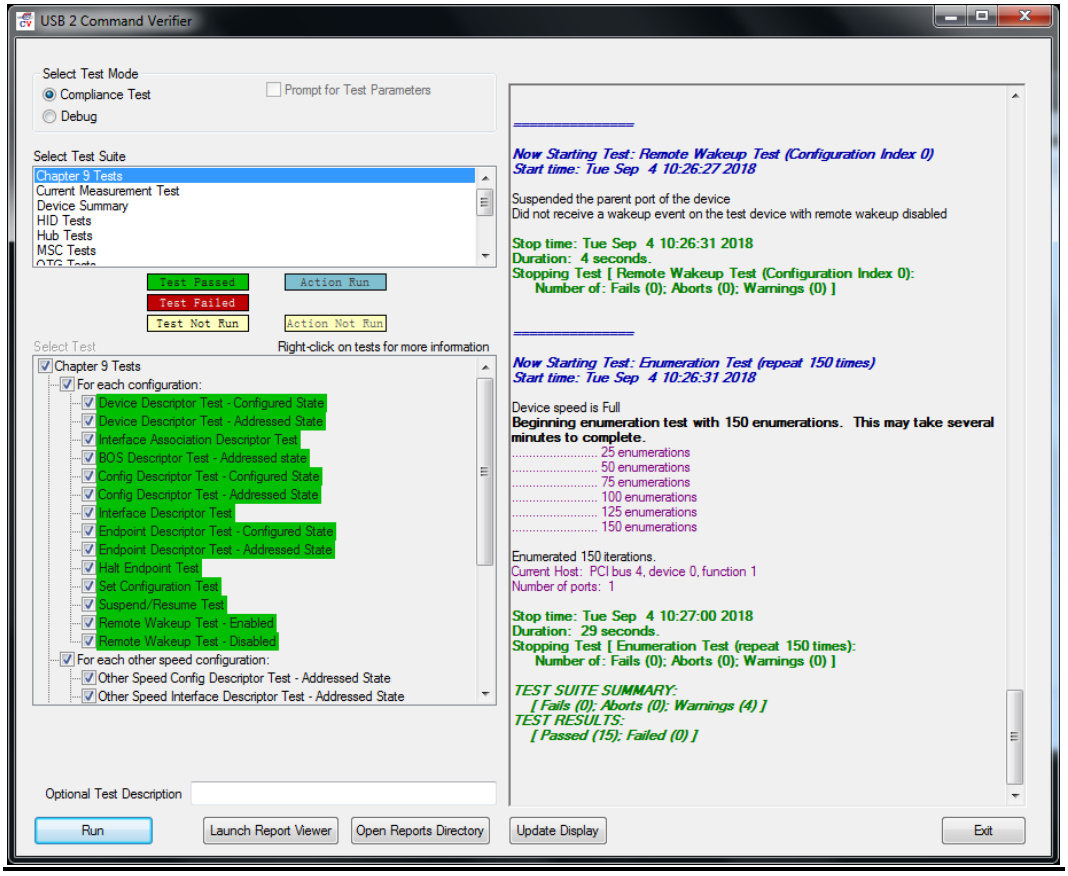

Fig 10. Test pass

6. Test Report of QN9080 USB Certification

QN9080 DK has passed the USB certification test. The test report can be found by the link https://cms.usb.org/company-products.

7. Index

Contents

8. List of figures

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

For more information, visit: http://www.nxp.com

Date of release: 05 March 2019
Document identifier: AN12307