UM11356

QN9090 Getting Started

Rev.1.0 — 14 January 2020   **User manual**

Revision history

Rev	Date	Description
1.0	20200114	Initial Revision

Contact information

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

1. Introduction

QN9090 Development Kit (DK) is designed for evaluating and developing Bluetooth Low Energy (BLE) solutions based on QN9090. The module features GPIO, USB, PMod, and SWD interface. J-Link and J-Trace functions are both supported for debugging. ISP download function is also supported with the QN9090 DK.

This Quick Start document provides an overview about the QN9090 DK and its software tools and lists the steps to install the hardware and the software. The document also describes how to run the demo example and create a new application project.

2. Kit contents

The QN9090 DK is composed of four parts.

- **QN9090 Carrier Board**: A hardware carrier board host a QN9090 for BLE application development.

  Carrier Board comprises NFC tag and antenna, GPIO connector, buttons, external flash, LEDs and Arduino connector used to connect Expansion Board. The LPC4322 Onboard (OB) debugger is used to bridge QN9090 SWD and UART interface to PC, download program, and debug from PC. There also have FTDI USB in board provided interface that can connect QN9090 UART directly and download program via ISP.

  For detailed information, refer to *QN9090 DK User Guide*.

Fig 1.   QN9090 Carrier Board

- **M10 Module:** A module including the QN9090T standard-power module pre-fitted to the Carrier Board.

Fig 2.   QN9090T M10 Module

- **Expansion Board**: A generic switch expansion board (OM15082) which can be mounted on the Carrier Board.
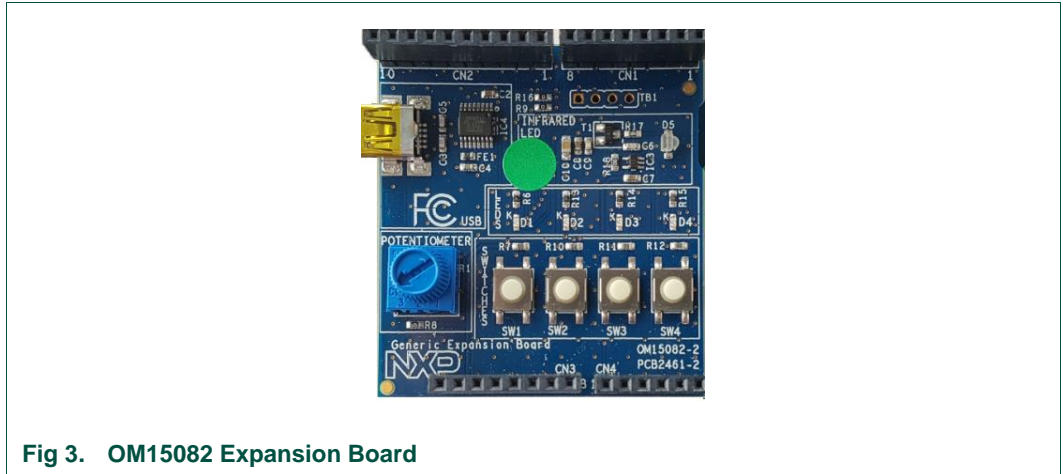

Fig 3.   OM15082 Expansion Board

- **USB Cable**: A cable to connect the QN9090 carrier board with the PC via Mini USB interface.

3.   Download and install software

Before connecting the QN9090 Carrier Board to PC, install/download the following software tools on PC:

- MCUXpresso SDK
- IAR Embedded Workbench for Arm v8.31.1 or MCUXpresso IDE v11.0.1*, User can choose any one of these IDEs for the development on QN9090DK. Higher versions of those IDEs are also supported.
- Python 2.7

Optional:

- Connectivity QTool: Tool to communicate with module over BLE
- DK6 Production Flash Programmer: Programming the device over the UART

3.1 System requirements

The development on the QN9090-DK has the following minimum system requirements:

- PC running with Microsoft® Windows® 7/8/10 (32-bit or 64-bit)
- 2 GB RAM and 8 GB hard-disk space
- Minimum of two USB ports

3.2 QN9090 SDK/PC tool installation

QN9090 SDK is a package containing source code and documents necessary for application firmware development. To ease debug and developing of QN9090.

3.2.1 SDK Download process

Refer to the following steps to download the SDK:

1. Click the link below and navigate to the SDK building page:
    https://www.nxp.com/support/developer-resources/software-development-tools/mcuxpresso-software-and-tools/mcuxpresso-software-development-kit-sdk:MCUXpresso-SDK?fsrch=1&sr=2&pageNum=1.

2. Click the button **Download** to start the SDK builder.

3. Click **Select Development Board** and log-in to start building the QN908x SDK.

    Note: The SDK can also be found by the link: Use the direct URL for QN9090: https://mcuxpresso.nxp.com/en/select?device=QN9090.

4. Navigate to the folder specified in Step 3 to open the SDK project.

3.2.2 PC software tool Download process

Refer to the following steps to download the PC tools:

1. Click the link below to download PC Tools.

    -   **Connectivity QTool:**
        https://www.nxp.com/webapp/sps/download/license.jsp?colCode=Connectivity-QTool-Setup

    -   **DK6 Production Flash Programmer**

        This installer is located in the Tools folder of the SDK.

2. Navigate to the folder where you downloaded the QTool.

3. Click the *.exe files to start the installation.

3.2.3 PC software tool introduction

1. Connectivity QTool is a software tool that runs on a PC and talks with the USB Dongle to act as a peripheral/central device. It provides ease in the debugging and development of a QN908x-based device.

2. DK6 Production Flash Programmer is available as a standalone application which can be used to program the QN9090 device, and it is a software tool that runs on a PC and talks with the QN9090 DK for firmware uploading via UART ISP interface.

3.3 Install MCUXpresso

MCUXpresso provides comprehensive compiler supporting QN, Kinetis, and LPC Microcontrollers.

You can download it from the following URL:

https://www.nxp.com/support/developer-resources/software-development-tools/mcuxpresso-software-and-tools/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE.

Double-click the file to run and complete the installation.

3.4 Install Python 2.7

QN9090 bootloader requires an image signature to verify the validity of the image. The Binary image generated is signed after the image is built in a two-stage process. The image signing tool is implemented in python. This requires an installation of python to exist. Python 2.7 is required.

Python can be downloaded from the following URL:

https://www.python.org/downloads/release/python-2713/

Once python is installed, it should be added to the windows system environment variable path, like below:
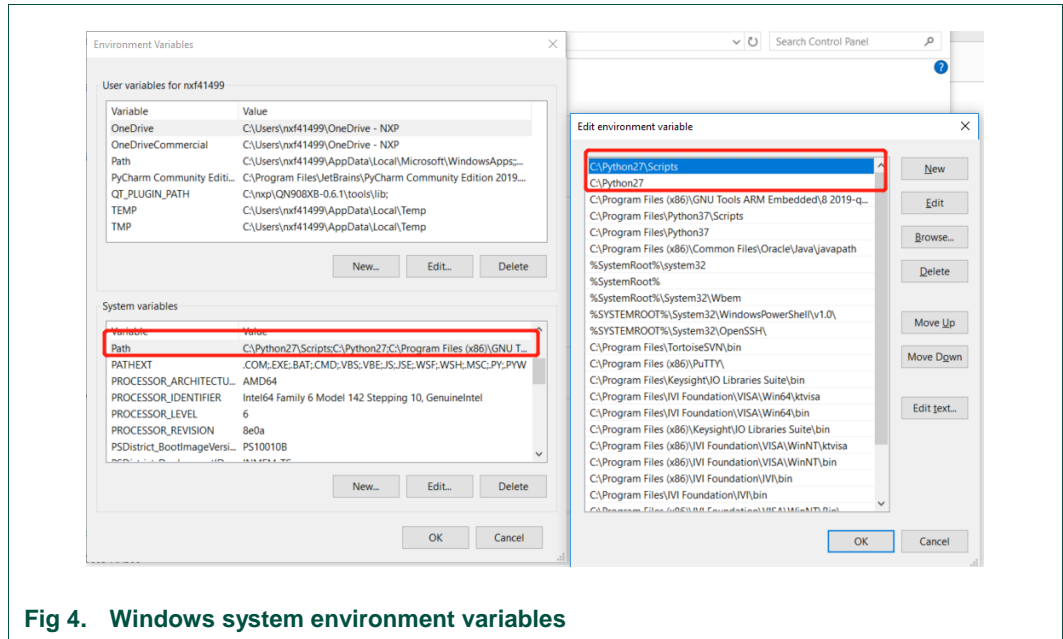
Fig 4.   Windows system environment variables

The python file is now using a crypto module, therefore you have to install the module to compile successfully, open the command prompt and enter the following command:

    C:\Python27 install path\Scripts>pip install pycryptodome

Note: python 2.7.13 (https://www.python.org/downloads/release/python-2713/) is required to have the pip tool.

4.   Connect hardware and install drivers

Before connect carrier board to PC, make sure M10 module is mounted on carrier board, like figure 6.
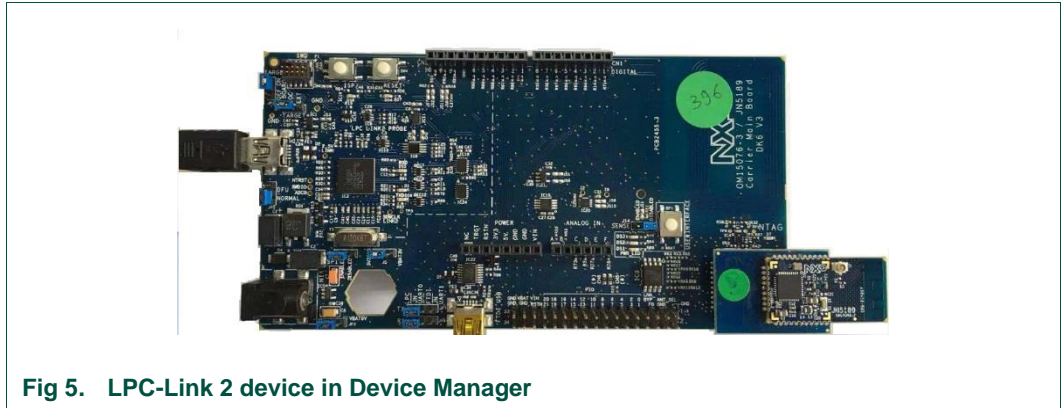
Fig 5.   LPC-Link 2 device in Device Manager

4.1   Connecting the QN9090 carrier board

4.1.1   Connect debugger USB

Connect the QN9090 carrier board to a PC with a Mini USB cable. Use connector "**USB to JTAG/SWD**" on the QN9090 carrier board.

Note: To connect debugger to QN9090 UART, make sure pin LPC and UART0 are connected on both JP7 and JP4.

Once connected to the PC, Windows will detect the new hardware. Run the driver installation tool lpc_driver_installer.exe, found inside of the LPSCrypt installation direct at C:\NXP\LPCScrypt\Drivers. This will install the drivers for both the virtual COM port and CMSIS-DAP debugger. After drivers are installed correctly, the device is shown in Device Manager as Figure 7.

Fig 6.   LPC-Link 2 device in Device Manager

4.1.2   Connect FTDI USB

Connect the QN9090 carrier board to a PC with a Mini USB cable. Use connector "**FTDI USB**" on the QN9090 carrier board.

Note: To make FTDI USB available, make sure pin FTDI and UART0 are connected on both JP7 and JP4.

The FTDI driver must be installed on you PC hard drive from: http://www.ftdichip.com/Drivers/CDM/CDM21228_Setup.zip

After drivers are installed correctly, the device is shown in Device Manager as Figure 8

Fig 7. LPC-Link 2 device in Device Manager

4.2 Connect BLE dongle and install driver

QN9090 BLE dongle (available separately) also uses FTDI as bridge to connect QN9090 UART with USB. The procedure of the driver installation is the same as connect the QN9090 carrier board FTDI USB, refer chapter 4.1.2.

4.3 Update debugger firmware to support J-Link/CMSIS-DAP

LPC-Link 2 is integrated on the QN9090 DK by the chip LPC4322, CMSIS-DAP is supported by default. J-Link is also supported by LPC-Link 2 by changing firmware inside the LPC4322.

Currently, only J-Link is supported on the SWD firmware updating function of QN9090 Programming Tool.

LPCScrypt is used to update the firmware for the LPC4322 on the QN9080 DK. The procedure below is about how to update firmware to support J-Link, and vice versa.

1. Short JP5-2 to JP5-3 to enable DFU boot. By default, Jp5-1 short to Jp5-2.

2. Short JP1-2 to Jp1-3 to off board target to debug location.

3. Then power cycle the QN9080 DK.

4. Go to **Start Menu** of Windows.

5. Click the option, *"Program LPC-Link2 with SEGGER J-Link"* under LPCScrypt.
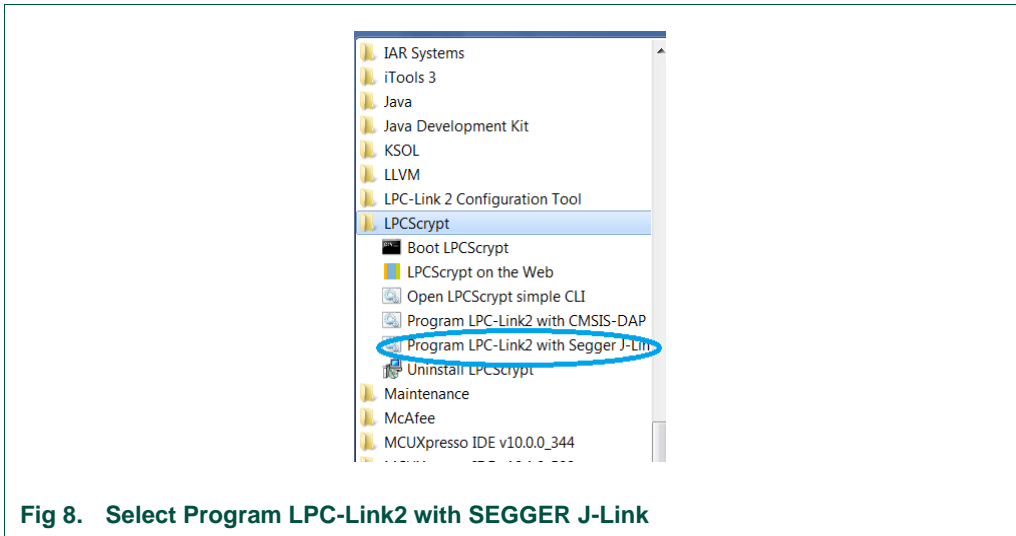
Fig 8. Select Program LPC-Link2 with SEGGER J-Link

6. Press any key in the Window "Program LPC-Link2 with SEGGER J-Link", and the firmware update starts.
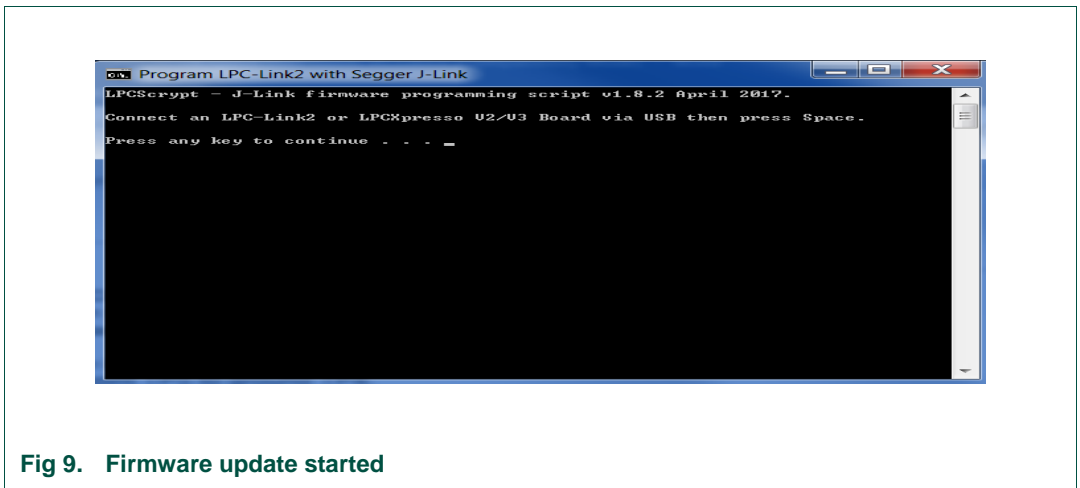

Fig 9. Firmware update started

7. The content in the Window when firmware updating complete is shown below.
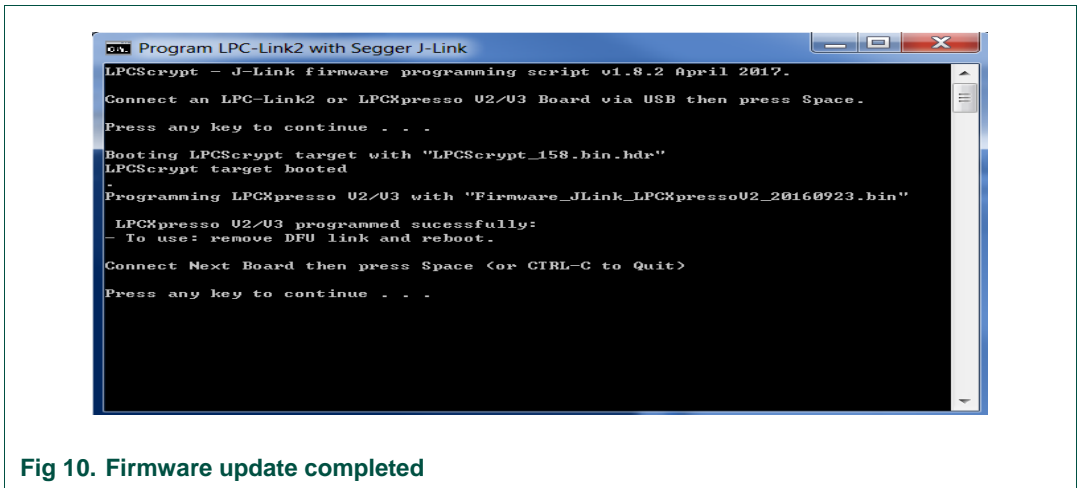

Fig 10. Firmware update completed

8. Restore JP5 and JP1 to the default setting and power cycle the board, J-Link is being used now.

9. Click the option "Program LPC-Link2 with CMSIS-DAP" in step 2 when CMSIS-DAP needs to be supported, other steps are the same as J-Link firmware updating.

5. Developing advanced applications

Two IDEs, IAR and MCUXpresso are supported in the QN9090 SDK. For more information, refer to the document **BLE Quick Start Guide.pdf** in the SDK path …\docs\wireless\Bluetooth elaborated how to open/compile a project and run a BLE or other demo project.

Note: For the IAR, make sure the SDK store path does not contain Spaces or special characters, otherwise a compilation error will occur.

6. Using the Production Flash Programmer

DK6 Production Flash Programmer support UART hardware interface to update firmware for QN9090. Both debugger USB and FTDI USB can used but FTDI is recommended as the interface supports higher baud rates and the ISP_ENTRY And reset pins are toggled automatically.

The DK6 Production Flash Programmer tool can be run from the Windows command prompt, it supports reading, writing flash, etc. For detailed information, refer to **JN-UG-3127-DK6-Production-Flash-Programmer.**

6.1 Programming QN9090 t

QN9090 bootloader supports firmware updating by ISP mode.

1. Connect debugger USB port with Mini USB cable. Make sure pins **LPC** and **UART0** are connected on both JP7 and JP4,

    OR

    Connect FTDI USB port to PC with Mini USB cable. Make sure pins **FTDI** and **UART0** are connected on both JP7 and JP4

2. Held press ISP button on the carrier board and then press once RESET button, then QN9090 should work in ISP programming mode (not needed when using the FTDI USB interface as this is automatic).

3. Put the binary that need to download to QN9090 on the DK6 Production Flash Programmer directory where the executable is located.

4. Launch the Windows command prompt, navigate to the DK6 Production Flash Programmer directory.

5. Download binary to flash by entering command:

    DK6Programmer.exe -V2 -P 1000000 -s COM21 -p NTAG_Pairing_Demo_qn9090dk6.bin

6. The Programmer will ask if want to erase the flash area, show in figure below.
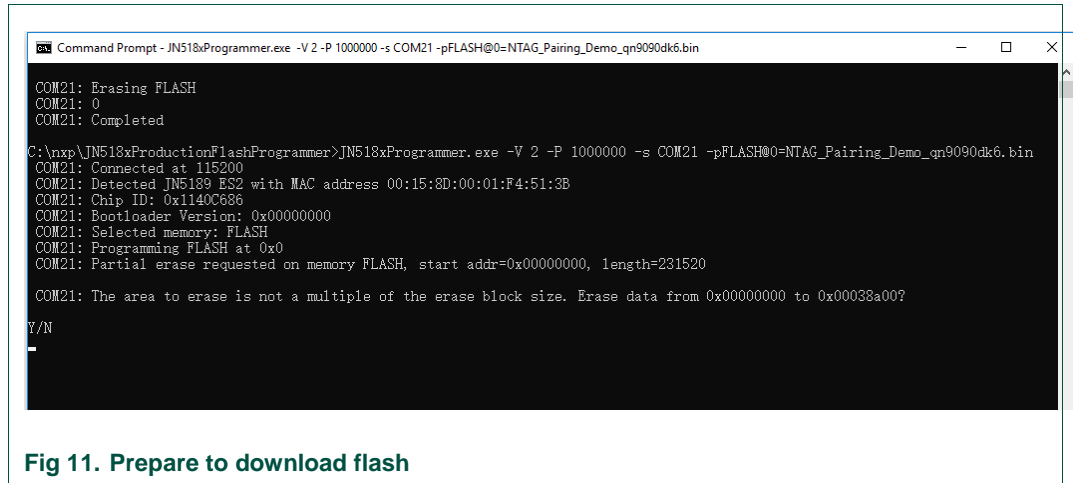
Fig 11. Prepare to download flash

7. Entering 'Y' to start erase and download.

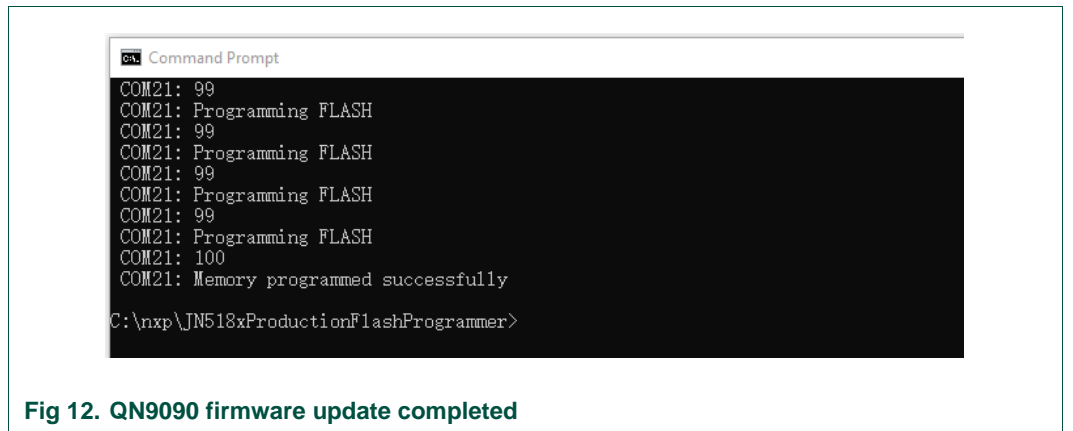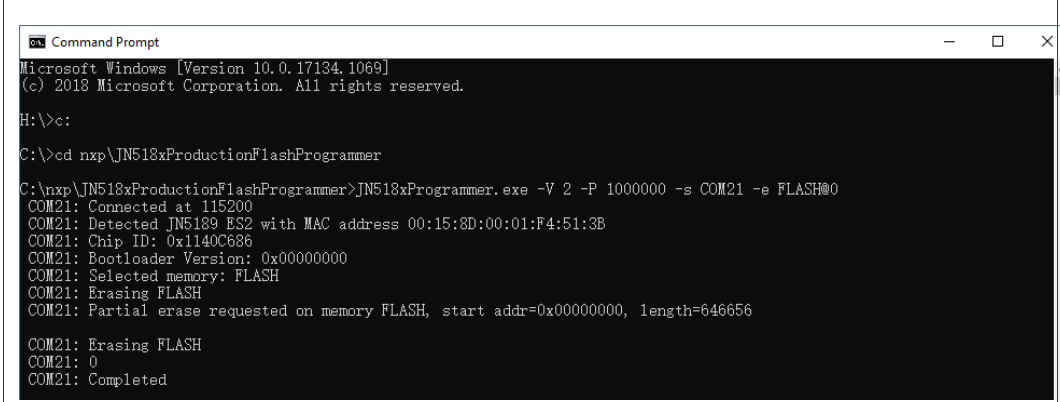8. The content in the Window when download complete is shown in the figure below.

Fig 12. QN9090 firmware update completed

9. You can erase the entire flash memory on the QN9090 by entering the command:

   DK6Programmer.exe -V 2 -P 1000000 -s COMxx -e

10. The content in the Window when flash erase complete is shown in the figure below.

Fig 13. QN9090 Flash erase completed

6.2   Updating the FTDI USB configuration

Unlike update via debugger USB, FTDI does not require the user to manually control QN9090 into ISP mode, but requires FTDI to be properly configured, after which FTDI is properly configured user could follow the procedures start from step 3 of chapter 6.1 to download flash. The FTDI only require configure once and is already done on all NXP production parts.

1. Connect debugger USB port to PC with Mini USB cable, on both JP7 and JP4, make sure pin FTDI and UART0 are short.

2. Download and install the FT_PROG tool from FTDI

   https://www.ftdichip.com/Support/Utilities/FT_Prog_v3.6.88.402%20Installer.exe

3. Navigate to the Startup menu of Windows.

4. Open the tool under the folder Start menu > FTDI > FT_Prog.

5. On the EEPROM window, click DEVICES >Scan and Parse.

6. Click FT EEPROM > Hardware Specific > CBUS Signals to check if both property C2 and C3 value are **GPIO**.

7. If C2 and C3 value are not **GPIO**, then set the value to GPIO, then click DEVICES > program.

7.   Appendix A: Additional Tools

7.1   Connectivity QTool

Connectivity QTool is a PC tool that works with the DK or dongle to ease the development of a BLE project in SDK. Refer to the procedure below on how to run Connectivity QTool to talk with QN9090 DK.

1. Connect the QN9090DK to the USB port of a PC.

2. Go to the **Startup** menu of Windows, open the tool under the folder **Start menu > NXP > Connectivity QTool**.

3. Choose the COM port of BLE Dongle and click **Open**.

UM11356

User manual **Rev.1.0 — 14 January 2020** **12 of 16**

4. Click the **Start Scanning** button to find the BLE device.

5. Click the **Stop Scanning** button when found desirable BLE device by Mac address.

6. Click the device found in the **Device** window at left side.

7. Click **Connect** button on the **Settings** window to connect *QN9090*.

8. The status changes to *Connected at Device* window at left side.

7.2 CMSIS-DAP or J-Link

QN9090 DK is shipping out with CMSIS-DAP enabled by default as debugger. All SDK projects are configured as CMSIS-DAP as debugger.

To use the PC tools contained in the SDK package you may need to enable J-Link. The tool LPCScrypt is used to update firmware of the debugger chip LPC4322 on QN9090 DK.

7.2.1 Install LPC-Link2 software

LPCScrypt is a command-line based, fast flash, EEPROM, OTP and security programming tool that supports LPC-Link2 Debug Firmware Programming.

You can download it from the following URL:

https://www.nxp.com/support/developer-resources/software-development-tools/lpc-developer-resources-/lpc-microcontroller-utilities/lpcscrypt-v1.8.2:LPCSCRYPT?tab=Design_Tools_Tab.

Double-click the file to run and complete the installation.

7.2.2 Install J-Link software

For the cases of J-Link needed, J-Link software must be installed as J-Link driver resides in the package.

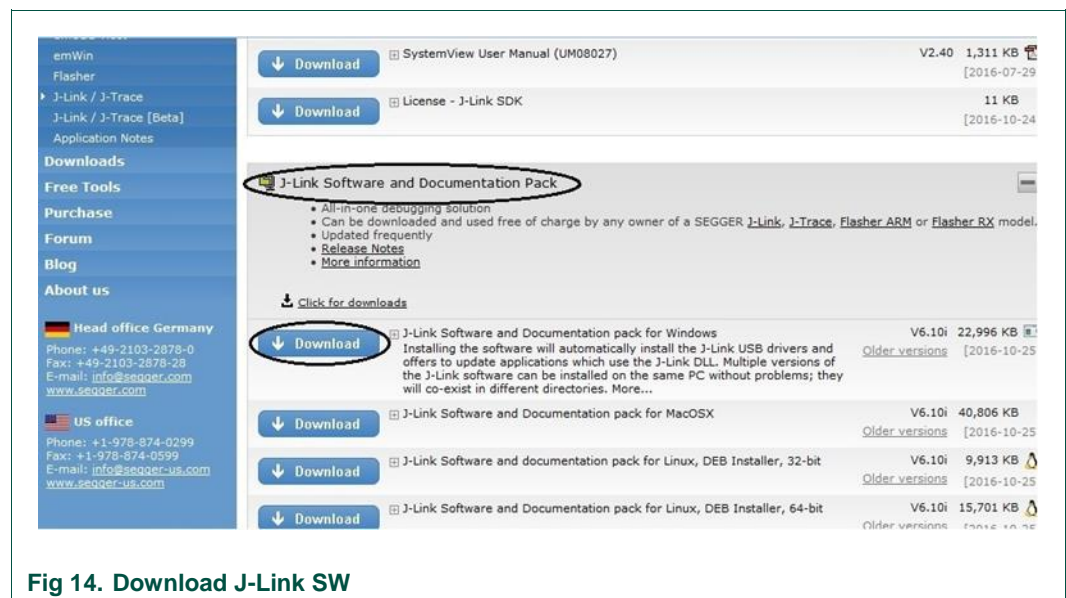The J-Link software and documentation package is available for download from https://www.segger.com/downloads/jlink.

Fig 14. Download J-Link SW

8. Abbreviations

The following abbreviations are used in the document.

Table 1. Abbreviations

Name	Description
DK	Development Kit
SoC	System on Chip
BLE	Bluetooth Low Energy
GPIO	General Purpose Input Output
OB	On Board
ISP	In System Program
SPI	Serial Port Interface
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver/Transmitter
MDK	Microcontroller Development Kit

9. References

- BLE Host Stack FSCI Reference Manual
- UM11085 Connectivity QTool User Manual
- DK6 Production Flash Programmer User Guide
- BLE Demo Applications User's Guide
- BLE Application Developer's Guide
- QN9090 DK User's Guide

UM11356

User manual **Rev.1.0 — 14 January 2020** **14 of 16**

10. Legal information

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.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C‑5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C‑Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, UMEMS, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V.

All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

11. Contents