



DK6 Production Flash Programmer User Guide

JN-UG-3127
Revision 3.0
16 January 2020

**DK6 Production Flash Programmer
User Guide**

Contents

Contents	3
Preface	5
Organisation	5
Conventions	5
Acronyms and Abbreviations	6
Related Documents	6
Support Resources	6
Trademarks	6
Chips	6
1. Getting Started	7
1.1 Overview	7
1.2 Installing the Tool	7
1.3 Running the Tool	8
1.4 Loading Application Binary File into Flash Memory	9
2. Command Reference	11
2.1 Command Options	11
2.2 Notes and Clarifications	14
2.2.1 Verbosity (-V)	14
3. Example Commands	17
3.1 List Available Serial Connections	17
3.2 Load Binary File into Flash Memory	17
3.3 Overriding the IEEE802.15.4 MAC Address	18
3.4 Overriding the BLE MAC Address	19
3.5 Dump the Contents of Flash Memory	19
3.6 Program Multiple Devices Simultaneously	20
3.7 Attempt to Open Non-existent Communications Port	21
3.8 Attempt to Access Inaccessible Device	21
Appendices	22
A. Identifying PC Communications Port Used	22

Contents

Preface

This manual describes use of NXP's DK6 Production Flash Programmer software tool. This is a command-line flash programming tool for use with the QN9030/9090, K32W041/061 and JN518x family of wireless microcontrollers, and this manual details the command options and provides example commands.

Organisation

This manual consists of 3 chapters and an appendix, as follows:

- [Chapter 1](#) introduces the tool, and describes how to install and use the tool
- [Chapter 2](#) details the available command options
- [Chapter 3](#) provides example commands
- The [Appendix](#) describes how to identify which serial communications port on the PC is being used for Flash programming

Conventions

Files and folders are represented in **bold type**.

Code fragments are represented in the `Courier New` typeface.



This is a **Tip**. It indicates useful or practical information.



This is a **Note**. It highlights important additional information.



*This is a **Caution**. It warns of situations that may result in equipment malfunction or damage.*

Acronyms and Abbreviations

CLI	Command Line Interface
SDK	Software Developer's Kit
Device	JN518x(T), QN9030(T), QN9090(T), K32W041, or K32W061

Related Documents

K32W061/041 Product Datasheet
QN9090(T)/QN9030(T) Product Datasheet
JN5188(T)/JN5189(T) Product Datasheet

Support Resources

To access online support resources such as SDKs, Application Notes and User Guides, visit the Wireless Connectivity section of the NXP web site:

www.nxp.com/products/wireless-connectivity

Trademarks

All trademarks are the property of their respective owners.

Chips

The DK6 Production Flash Programmer described in this manual can be used to program the following NXP wireless microcontroller devices:

JN5188, JN5188T, JN5189, JN5189T
QN9030, QN9030T, QN9090, QN9090T
K32W041, K32W061

1. Getting Started

This chapter introduces the DK6 Flash Programmer CLI (Command Line Interface) as well as describing how to install and use the tool.

1.1 Overview

The DK6 Flash Programmer CLI tool allows data and applications to be loaded from a host machine (such as a PC) into the Flash memory associated with an NXP wireless microcontroller. This memory may be the SPI-connected Flash memory chip on a associated module or dongle, or the on-chip Flash memory of device.

The CLI is provided as an executable (.exe file) in the software installer JN-SW-4407. The supplied executable is a MINGW program for Windows.

The CLI tool described in this manual is useful in a production environment where JN518x, QN9030/90, or K32W041/61 based products are pre-programmed with firmware applications and data (such as a MAC address, security key and user data).

1.2 Installing the Tool

The DK6 Flash Programmer CLI is supplied in the Windows installer file **JN-SW-4407.exe**. To install the tool on a Windows-based machine:

1. Store the installer **JN-SW-4407.exe** anywhere on the host machine.
2. Run this installer and follow the on-screen instructions.
3. Check that the tool executable **DK6Programmer.exe** has been installed in the following directory:

C:\INXP\DK6ProductionFlashProgrammer

You can now run the tool as described in [Section 1.3](#).

1.3 Running the Tool

Once installed (as described in [Section 1.2](#)), the tool can be run from the Windows command prompt or from a MINGW console. For example, to launch the Command Prompt in Windows 7, from the **Start** icon (bottom-left of screen) follow the path:

Start > All Programs > Accessories > Command Prompt

The tool is run by moving to the directory where the executable is located (see [Section 1.2](#)) and entering the name of the executable at the command prompt:

DK6Programmer.exe

Command line options are added to specify the particular commands to be performed. For example, to list all the available connections to serial devices, use the `-l` option:

```
$ DK6Programmer.exe -l
```

which will yield results in the following format:

```
Available connections:
COM3
COM32
```

All the available command options are listed and described in [Chapter 2](#), and example commands are provided in [Chapter 3](#). A procedure for loading an application binary file into the Flash memory of a device is provided in [Section 1.4](#).

1.4 Loading Application Binary File into Flash Memory

The procedure below outlines how to use the DK6 Flash Programmer CLI to load an application **.bin** file into the Flash memory associated with a device.

- Step 1** Connect a USB port of your PC to the target device (which may be mounted on a board or module) using a USB-to-serial cable (UART0 on the device is used for this connection). At this point, you may be prompted to install the driver for the cable.
- Step 2** Determine which serial communications port on your PC has been allocated to the USB connection - to identify the relevant port, refer to [Appendix A](#).
- Step 3** On your PC, open a command window - for example, as described in [Section 1.3](#).
- Step 4** In the command window, navigate to the Flash Programmer directory:

C:\INXP\DK6ProductionFlashProgrammer

- Step 5** Run the Flash programmer to download your binary file to Flash memory by entering a command with the following format at the command prompt:

```
DK6Programmer.exe <command options> -p <path to .bin file>
```

where **<command options>** can be one or more of the options listed and described in [Chapter 2](#), and must include **-s <comport>**.



Note: When attempting to program a DK6 on the OM15076 development board (DK6 type) through the LPC-LINK2 USB connector, the Flash Programmer does not automatically assert the program/reset lines. These lines must be manually put into programming mode before the tool is run by holding down the ISP button while pressing and releasing the RESET button.

Programming through the FTDI USB connector the assertion of the Program/Reset lines will be handled automatically.

- Step 6** Once the download has successfully completed, disconnect the serial cable and, if required, reset the board or module to run the application.

Chapter 1
Getting Started

2. Command Reference

The chapter lists and describes the command options that are available with the DK6 Flash Programmer CLI.

2.1 Command Options

The table below lists the command options (short and long forms) with their parameters and also provides a brief description of each.



Note: The `-s` option, which specifies a serial connection, must be used in conjunction with all other options except the `-l` option.

Option	Long Option	Parameter	Description
General Options			
-h	--help		To get chip specific help (for instance the list of aliases), use the -h option in association with the -s <comport> option. Example: DK6Programmer.exe -s COM56 -h
-l	--list		List all detected serial connections
-V	--verbosity	<verbosity>	Set verbosity level to the specified integer value in the range 0-2 (see Section 2.2.1). This increases/ decreases the amount of information displayed. The default value is 1
-Y	--force		Force operation. This option prevents the display of the confirmation message when programming Flash memory or loading incompatible files.
Connection Options			
-I	--initialbaud	<rate>	Set initial baud rate of serial connection to the specified value
-P	--programbaud	<rate>	Set programming baud rate of serial connection

Table 1: Command Options

Chapter 2
Command Reference

Option	Long Option	Parameter	Description
-s	--serial	<serial device>	Specify the serial device/connection at which the command is directed, e.g. COM1, /dev/ttyS1. May be specified multiple times for multiple devices. This option is required when specifying any other option with the exception of -1
Programming Options			
-m	--memory		List the available memories
-p	--program	[[memory] [:length] [@offset]=] <filename>	Load the contents of the specified file of the specified length (in bytes) starting at the specified offset (in bytes) of the specified memory
-d	--dump	[[memory] [:length] [@offset]=] <filename>	Save to the specified file the data contents of the specified length (in bytes) starting at the specified offset (in bytes) of the specified memory
-w	--write	<memory> [:length] [@offset]=<data>	Write the specified data of the specified length (in bytes) starting at the specified offset (in bytes) of the specified memory
		<alias>=<data>	Write to the specified alias with the specified data. Data should be a set of hexadecimal digits with no spaces, see Section 3.3 for an example
-r	--read	<memory> [:length] [@offset]	Read the specified length of data (in bytes) starting at the specified offset (in bytes) of the specified memory
		row:<alias>	Read the specified alias contents
-e	--erase	<memory> [:length] [@offset]	Erase the data contents of the specified length (in bytes) starting at the specified offset (in bytes) of the specified memory
-v	--verify		Verify the contents of flash once programmed
-N	--noresetdevice		Do not reset the device when all operations on the command line are completed

Table 1: Command Options



Note: Some of the command parameters contain optional elements (indicated within square brackets). If these are not specified, the default behaviour assumes the devices internal Flash memory is to be accessed at a zero offset and with an indefinite data length.

2.2 Notes and Clarifications

This section provides further details on certain command options listed and described in [Section 2.1](#).

2.2.1 Verbosity (-V)

The verbosity option `-V` determines the way in which reported information is presented in the command window. The option can take one of three values:

- **0:** Regular progress information is not displayed but a message to indicate that a command is being processed is displayed in the command window
- **1 (default):** Progress information is displayed in a Curses window overlaid on the command window
- **2:** Regular progress information is displayed directly in the command window

Illustrations of the outputs with verbosity levels 0 and 2 are shown below.

In the following command, the verbosity level is set to 0:

```
$ DK6Programmer.exe -V 0 -s COM32
  -p AN1180_154_HomeSensorCoord_JN5189.bin
COM32: Detected JN5189 with MAC address 00:15:8D:00:00:32:DB:4E
COM32: Programming Flash
COM32: Memory programmed successfully
```

The above output simply reports once on each operation requested in the command.

In the following command, the verbosity level is set to 2:

```
$ DK6Programmer.exe -V 2 -s COM32
  -p AN1180_154_HomeSensorCoord_JN5189.bin
COM32: Detected JN5189 with MAC address 00:15:8D:00:00:32:DB:4E
COM32: Setting baudrate: 1000000
COM32: Programming Flash
COM32: Erasing Flash
COM32: 0
COM32: Erasing Flash
COM32: 100
COM32: Programming Flash
COM32: 0
COM32: Programming Flash
COM32: 0
...
...
COM32: Programming Flash
COM32: 99
COM32: Programming Flash
```

```
COM32: 100  
COM32: Memory programmed successfully  
COM32: Setting baudrate: 38400
```

The above output contains regular reports on each operation requested in the command.

Chapter 2
Command Reference

3. Example Commands

This chapter contains a number of example commands that can be issued using the DK6 Flash Programmer CLI. The tool is run and commands issued from the command line on a PC as described in [Section 1.3](#).

The sections below provide example commands for common tasks.



Note 1: It is assumed that a target QN9030/90, K32W041/61 or JN518x-based device is connected to the PC via a USB port. The PC serial connection allocated to this device can be identified as described in [Appendix A](#).

3.1 List Available Serial Connections

The following command uses the `-l` option to list all the current connections to serial devices:

```
$ DK6Programmer.exe -l
```

This displays results in the following format in the command window:

```
Available connections:  
COM24  
COM41
```

3.2 Load Binary File into Flash Memory

The following command uses the `-p` option to load the specified binary file into the Flash memory of the JN518x device on the serial connection specified using the `-s` option:

```
$ DK6Programmer.exe -s COM24  
-p AN1180_154_HomeSensorCoord_JN5189.bin
```

This results in the following output in the command window:

```
MINGW32/c/NXP/DK6ProductionFlashProgrammer  
NXP DK6 Device Programmer (Build 2273)  
COM24  
Detected K32W061 with MAC address 00:15:8D:00:03:1F:0C:F1  
Programming FLASH  
[=====] 68%
```

3.3 Overriding the IEEE802.15.4 MAC Address

The following command uses the `--write` option to override the internal IEEE802.15.4 the MAC address of the device on the serial connection specified using the `-s` option:

```
$ DK6Programmer.exe -s COM24 --write 154_cmac0=00158d0012345678
```



Note 1: Using the alias for one of the IEEE802.15.4 MAC address values (154_fmac0, 154_cmac0, 154_cmac1) causes the data from the command line to be written in the appropriate order for it to be used by the MAC software as expected. For all other writes, the data is written little-endian, so each set of 4 bytes is stored in the reverse order to how it appears on the command line.

3.4 Overriding the BLE MAC Address

The BLE MAC address is held in PFLASH.

The `--write` option can be used to override the internal MAC address of the device on the serial connection that is specified using the `-s` option. For example:

- DK6Programmer.exe `-s COMxx --write ble_cmac0=0000nnmmpqqrrss`, will program a MAC address in the form `nn:mm:pp:qq:rr:ss.`, the first 2 bytes set here to 0 are insignificant with respect to the MAC address.
- DK6Programmer.exe `-s COMxx --read ble_cmac0` can be used to read-back this address.

For instance after `DK6Programmer.exe -s COMxx --write ble_cmac0=0000123456789abc` the device will use address `12:34:56:78:9a:bc` at next reboot.

If this address has been left blank, the PFLASH, the BLE device will generate a random device address and store it to the PFLASH, until reprogrammed by the above described method.

3.5 Dump the Contents of Flash Memory

The following command uses the `-d` option to set save the contents of the internal Flash memory of the device on the serial connection specified using the `-s` option:

```
$ DK6Programmer.exe -s COM24 -d flash.bin
```

The data is saved to the file **flash_00-15-8D-00-00-04-92-22.bin** - the filename specified in the command is appended with the IEEE802.15.4 MAC address of the device.

This results in the following output in the command window:

```

MINGW32/c/NXP/DK6ProductionFlashProgrammer
NXP DK6 Device Programmer (Build 2273)
COM24
Detected K32W061 with MAC address 00:15:8D:00:03:1F:0C:F1
Dumping FLASH
[===== ] 7%
  
```

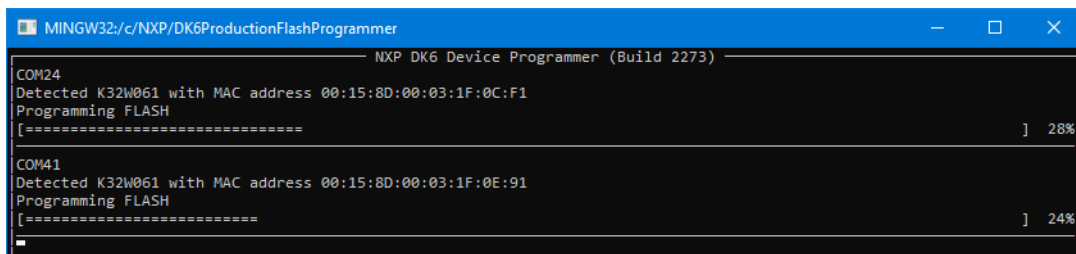
3.6 Program Multiple Devices Simultaneously

The tool allows the same operation to be performed on more than one connected device at the same time - for example, the same binary file can be loaded into the Flash memory of several devices. In this case, the `-s` option is used more than once in a single command.

The following command uses the `-p` option to load the specified binary file into the Flash memory of the devices on the (two) serial connections specified using the `-s` options:

```
$ DK6Programmer.exe -s COM24 -s COM41  
                                -p AN1180_154_HomeSensorCoord_JN5189.bin
```

This results in the following output in the command window.



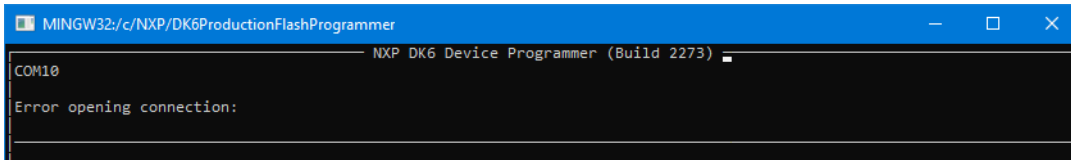
```
MINGW32/c/NXP/DK6ProductionFlashProgrammer  
NXP DK6 Device Programmer (Build 2273)  
COM24  
Detected K32W061 with MAC address 00:15:8D:00:03:1F:0C:F1  
Programming FLASH  
[=====] 28%  
COM41  
Detected K32W061 with MAC address 00:15:8D:00:03:1F:0E:91  
Programming FLASH  
[=====] 24%
```

3.7 Attempt to Open Non-existent Communications Port

The following command uses the `-s` option to attempt to access a non-existent serial communications port (one which has not been allocated to a current connection):

```
$ DK6Programmer.exe -s COM10
```

This results in the following output in the command window.



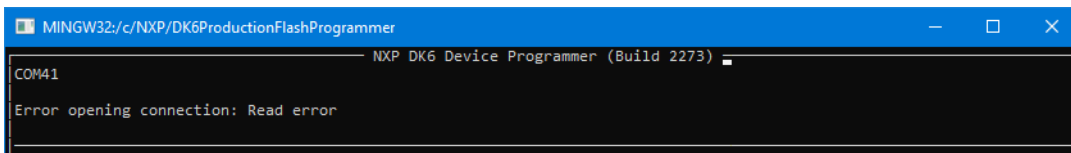
```
MINGW32:/c/NXP/DK6ProductionFlashProgrammer
NXP DK6 Device Programmer (Build 2273)
COM10
Error opening connection:
```

3.8 Attempt to Access Inaccessible Device

The following command uses the `-s` option to attempt to access a device on a valid communications port but the device cannot be accessed for some reason (e.g. USB cable has been disconnected or target device is not in programming mode):

```
$ DK6Programmer.exe -s COM24
```

This results in the following output in the command window.



```
MINGW32:/c/NXP/DK6ProductionFlashProgrammer
NXP DK6 Device Programmer (Build 2273)
COM41
Error opening connection: Read error
```

Chapter 3
Example Commands

Appendices

A. Identifying PC Communications Port Used

In order to use the DK6 Flash Programmer CLI, you need to find out which serial communications port your PC has allocated to the connection with the board/module containing the target device - this is described below (for Windows 7,8,and 10).

For Windows 7 and Windows 8:

Step 1 In the Windows **Start** menu, open the **Control Panel** by following the menu path:

Start>Control Panel (Windows 7) or

Start>All Apps>Control Panel (Windows 8)

Step 2 From the **Control Panel**, open the **Device Manager** by following the path:

System>Device Manager (Windows 7) or

System & Security>Device Manager (Windows 8)

Step 3 Within the **Device Manager** screen:

a) Look for the **Ports** folder in the list of devices and unfold it.

b) Identify the port which is connected to the serial device (it will be labelled 'USB Serial Port') and make a note of it (e.g. COM1).

You will need to specify this port in the option `-s <comport>` when accessing the target device using the DK6 Flash Programmer CLI (see [Section 2.1](#)).

For Windows 10:

Step 1 In the Windows **Start** Menu, open the **Device Manager** by following the path:

Start>Device Manager.

Step 2 Look for the **Ports** folder in the list of devices and unfold it.

Step 3 Identify the port which is connected to the serial device (it will be labelled 'USB Serial Port') and make a note of it (e.g. COM1).

You will need to specify this port in the option `-s <comport>` when accessing the target device using the DK6 Flash Programmer CLI (see [Section 2.1](#)).

Revision History

Version	Date	Comments
1.0	21 June 2018	Initial release
1.1	23 November 2018	Updated Command Options
2.0	19 November 2019	Added support for additional devices
3.0	16 January 2020	Overriding the BLE MAC Address

Important Notice

Limited warranty and liability - Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use - NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications - Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control - This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

NXP Semiconductors

For online support resources and contact details of your local NXP office or distributor, refer to:

www.nxp.com

