



JN51xx Production Flash Programmer User Guide

JN-UG-3099
Revision 1.1
22 December 2015

JN51xx Production Flash Programmer
User Guide

Contents

About this Manual	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	8
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	13
2.2.1 Verbosity (-V)	13
2.2.2 Device Configuration (--deviceconfig=)	14
2.2.3 License File (-L)	15
3. Example Commands	17
3.1 List Available Serial Connections	17
3.2 Load Binary File into Flash Memory	17
3.3 Erase Complete Contents of EEPROM	18
3.4 Load Binary File, Verify Write and Erase EEPROM	18
3.5 Set MAC Address of JN51xx Device	19
3.6 Display the Device Configuration	19
3.7 Set the Device Configuration and AES Key	20
3.8 Dump the Contents of Flash Memory	21
3.9 Dump the Contents of EEPROM	21
3.10 Switch to External Flash Memory	22
3.11 Program Multiple Devices Simultaneously	23
3.12 Program Permanent User Data	24
3.13 Specify a License File	24

Contents

3.14 Load and Execute an Application in JN51xx RAM	25
3.15 Attempt to Open Non-existent Communications Port	26
3.16 Attempt to Access Inaccessible Device	26
Appendices	27
A. Identifying PC Communications Port Used	27

About this Manual

This manual describes use of NXP's JN51xx Production Flash Programmer software tool, which is provided in the software package JN-SW-4107. This is a command-line tool, and this manual details the command options and provides example commands.

Organisation

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

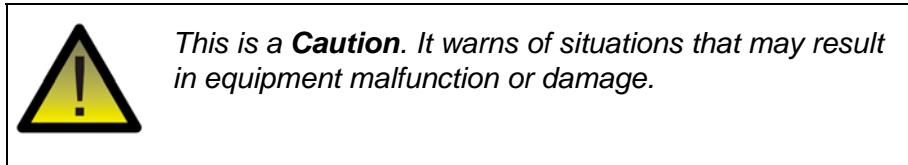
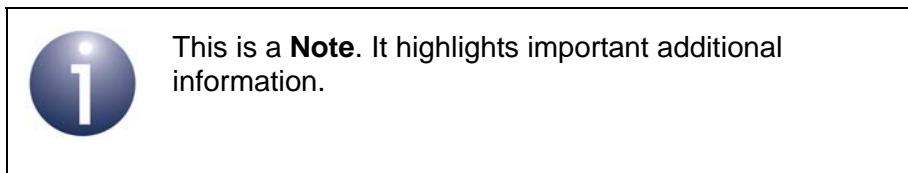
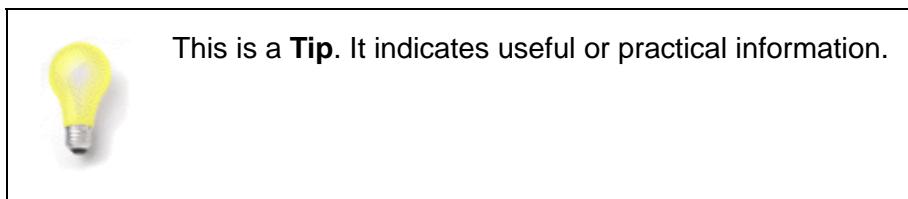
- [Chapter 1](#) introduces the tool, and describes how to install and use the tool
- [Chapter 2](#) details the available command options
- [Chapter 3](#) provides example commands
- The [Appendix](#) describes how to identify which serial communications port on the PC is being used for Flash programming

Conventions

Files, folders, functions and parameter types are represented in **bold** type.

Function parameters are represented in *italics* type.

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



Acronyms and Abbreviations

API	Application Programming Interface
IDE	Integrated Development Environment
SDK	Software Developer's Kit

Related Documents

JN-DS-JN516x	JN516x Data Sheet
JN5179	JN5179 Data Sheet

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/interface-and-connectivity/wireless-connectivity

All NXP resources referred to in this manual can be found at the above address, unless otherwise stated.

Trademarks

All trademarks are the property of their respective owners.

Chips

The JN51xx Production Flash Programmer described in this manual can be used to program chips from NXP's JN516x and JN517x families of wireless microcontroller. For convenience, these chips will be referred to as JN51xx throughout this manual.

1. Getting Started

This chapter introduces the JN51xx Flash Programmer CLI (Command Line Interface) as well as describing how to install and use the tool.

1.1 Overview

The JN51xx 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 JN51xx wireless microcontroller. This memory may be the SPI-connected Flash memory chip on a JN51xx module or dongle, or the on-chip Flash memory of a JN51xx microcontroller.

The CLI is provided as an executable (.exe file) in the software installer JN-SW-4107, which can be downloaded free-of-charge from the NXP web site (see “[Support Resources](#)” on page 6). The supplied executable is a MINGW program for Windows.



Note: A GUI (Graphical User Interface) version of the JN51xx Flash Programmer is also available. For JN516x chips, it is provided in the IDE (Integrated Development Environment) *Beyond Studio for NXP* (JN-SW-4141). For JN517x chips, it is provided as a plug-in for ‘LPCXpresso for JN517x’ in the relevant Software Development Kits (SDKs).

The CLI version of the tool, described in this manual, is useful in a production environment where JN51xx-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 JN51xx Flash Programmer CLI is supplied in the Windows installer file **JN-SW-4107.exe**. To install the tool on a Windows-based machine:

1. Store the installer **JN-SW-4107.exe** anywhere on the host machine.
2. Run this installer and follow the on-screen instructions.
3. Check that the tool executable **JN51xxProgrammer.exe** has been installed in the following directory:

C:\NXP\ProductionFlashProgrammer

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:

JN51xxProgrammer.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:

```
$ JN51xxProgrammer.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 JN51xx device is provided in [Section 1.4](#).

1.4 Loading Application Binary File into Flash Memory

The procedure below outlines how to use the JN51xx Flash Programmer CLI to load an application .bin file into the Flash memory associated with a JN51xx device.

- Step 1** Connect a USB port of your PC to the target JN51xx device (which may be mounted on a board or module) using a USB-to-serial cable (UART0 on the JN51xx 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:\NXP\ProductionFlashProgrammer

- 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:

```
JN51xxProgrammer.exe <command options> -f <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: The Flash programmer automatically asserts the program/reset lines on the current NXP evaluation kit boards (DK4 type) but older Jennic boards (DK2 type) must be manually put into programming mode before the tool is run - if not, the following message will be displayed: "Error reading module information - check cabling and power".

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

2. Command Reference

The chapter lists and describes the command options that are available with the JN51xx 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			
<code>-l</code>	<code>--list</code>		List all detected serial connections
<code>-v</code>	<code>--verbosity</code>	<code><verbosity></code>	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
<code>-Y</code>	<code>--force</code>		Force operation. This option prevents the display of the confirmation message when programming One Time Programmable (OTP) memory or loading incompatible files.
Connection Options			
<code>-I</code>	<code>--initialbaud</code>	<code><rate></code>	Set initial baud rate of serial connection to the specified value
<code>-P</code>	<code>--programbaud</code>	<code><rate></code>	Set programming baud rate of serial connection
<code>-s</code>	<code>--serial</code>	<code><serial device></code>	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 -l
Programming Options			
	<code>--deviceconfig</code>		Display JN51xx device configuration
	<code>--deviceconfig=</code>	<code><config></code>	Set JN51xx device configuration. Configuration is a comma-separated list (see Section 2.2.2). The device configuration data is one-time programmable (so cannot be erased)

Table 1: Command Options

Chapter 2
Command Reference

Option	Long Option	Parameter	Description
	--eepromoffset	<offset>	Load/dump the EEPROM at the specified number of bytes from the start.
	--flashoffset	<offset>	Load/dump the Flash memory at the specified number of bytes from the start.
-e	--loadEEPROM	<filename>	Load contents of specified file into JN51xx EEPROM
-E	--dumpEEPROM	<filename>	Save the contents of the JN51xx EEPROM to the specified file
	--eraseEEPROM	<full,pdm>	Erase the JN51xx EEPROM - persistent data only (pdm) or complete EEPROM erase (full)
-f	--loadFlash	<filename>	Load contents of specified file into Flash memory
-F	--dumpFlash	<filename>	Save the contents of Flash memory to the specified file
-k	--key	<AES key>	Set AES key of device - specify key as 128-bit hexadecimal string. This key is one-time programmable (so cannot be erased)
-L	--license	<license file>	Load MAC address, AES key and user data from specified license file into Flash memory (see Section 2.2.3). This data is one-time programmable (so cannot be erased)
-m	--mac	<MAC address>	Set MAC address of JN51xx device. The MAC address is one-time programmable (so cannot be erased)
-r	--loadRAM	<filename>	Load contents of specified file directly into JN51xx RAM and then execute it (the contents are not written to Flash memory)
-S	--selectFlash	<internal,external>	Choose the Flash memory device with which to interact - internal or external
-u	--userdata	<word=data>	Write one word of user data to the Flash memory. Three 128-bit words are available for user data in Flash memory, identified by 0, 1 and 2 - the parameter <code>word</code> is used to identify the word to be written to. The data is specified in the parameter <code>data</code> as a 128-bit hexadecimal string. This user data is one-time programmable (so cannot be erased)
-v	--verify		Verify that the programmed Flash image has been loaded correctly

Table 1: Command Options

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 overlayed 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:

```
$ JN51xxProgrammer.exe -V 0 -s COM32
-f AN1180_154_HomeSensorCoord_JN5168.bin -v --eraseeprom=full
COM32: Detected JN5168 with MAC address 00:15:8D:00:00:32:DB:4E
COM32: Programming JN516x Internal Flash
COM32: Flash programmed successfully
COM32: Verifying JN516x Internal Flash
COM32: Flash verified successfully
COM32: EEPROM erased 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:

```
$ JN51xxProgrammer.exe -V 2 -s COM32
-f AN1180_154_HomeSensorCoord_JN5168.bin -v --eraseeprom=full
COM32: Detected JN5168 with MAC address 00:15:8D:00:00:32:DB:4E
COM32: Setting baudrate: 1000000
COM32: Programming JN516x Internal Flash
COM32: Erasing JN516x Internal Flash
COM32: 0
COM32: Erasing JN516x Internal Flash
COM32: 100
COM32: Programming JN516x Internal Flash
COM32: 0
COM32: Programming JN516x Internal Flash
COM32: 0
...
...
...
```

Chapter 2

Command Reference

```
COM32: Programming JN516x Internal Flash
COM32: 99
COM32: Programming JN516x Internal Flash
COM32: 100
COM32: Flash programmed successfully
COM32: Verifying JN516x Internal Flash
COM32: Verifying JN516x Internal Flash
COM32: 0
COM32: Verifying JN516x Internal Flash
COM32: 0

...
...
COM32: Verifying JN516x Internal Flash
COM32: 100
COM32: Flash verified successfully
COM32: Erasing EEPROM
COM32: 0
COM32: Erasing EEPROM
COM32: 100
COM32: EEPROM erased successfully
COM32: Setting baudrate: 38400
```

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

2.2.2 Device Configuration (--deviceconfig=)

The JN51xx device configuration can be set using the `--deviceconfig=<config>` option, where `<config>` is a comma-separated list of one or more of the following:

Enable/Disable JTAG access:	<code>JTAG_ENABLE</code> / <code>JTAG_DISABLE</code>
Set brown-out voltage:	<code>VBO_195 (1.95V)</code> / <code>VBO_200 (2.0V)</code> <code>VBO_210 (2.1V)</code> / <code>VBO_220 (2.2V)</code> <code>VBO_230 (2.3V)</code> / <code>VBO_240 (2.4V)</code> <code>VBO_270 (2.7V)</code> / <code>VBO_300 (3.0V)</code>
Bootloader code protection:	<code>CRP_LEVEL0 (None)</code> <code>CRP_LEVEL1 (Flash read disabled)</code> <code>CRP_LEVEL2 (all access disabled)</code>
External Flash is encrypted:	<code>EXTERNAL_FLASH_NOT_ENCRYPTED</code> <code>EXTERNAL_FLASH_ENCRYPTED</code>
External Flash image loading:	<code>EXTERNAL_FLASH_LOAD_ENABLE</code> <code>EXTERNAL_FLASH_LOAD_DISABLE</code>

For an example of using this option, refer to [Section 3.7](#).

2.2.3 License File (-L)

An NXP-supplied license file can be used to program a MAC address into a JN51xx device. This license file contains a list of MAC addresses and, optionally, an AES key and/or user data for each address. When programming a device, use of the license file can be specified using the `-L` option. The tool will then program the device with the next available (previously unused) MAC address from the file. If a corresponding AES key and/or user data are also supplied in the file, these will also be programmed.

Each line of the license file contains a single entry in the following format:

```
<flag>,<mac address>,<optional AES key>,<optional userdata word 0>,<optional userdata word 1>,<optional userdata word 2>
```

The leading flag indicates whether the entry has been used (1) or remains unused (0). The user data comprises up to three 128-bit words, numbered 0, 1 and 2 (Flash memory words are reserved for this data).

Note that the MAC address, AES key and user data are one-time programmable, so are permanent and cannot be erased.

Example license file entries are provided below.

Unused entry with just MAC address

```
0, 0x00158d0012345678
```

Unused entry with MAC address and AES key

```
0, 0x00158d0012345678, 0x0123456789ABCDEF0123456789ABCDEF
```

Unused entry with MAC address, AES key and word 0 of user data

```
0, 0x00158d0012345678, 0x0123456789ABCDEF0123456789ABCDEF,  
0x0123456789ABCDEF0123456789ABCDEF
```

Unused entry with MAC address, AES key and words 0 and 1 of user data

```
0, 0x00158d0012345678, 0x0123456789ABCDEF0123456789ABCDEF,  
0x0123456789ABCDEF0123456789ABCDEF,  
0x0123456789ABCDEF0123456789ABCDEF
```

Unused entry with MAC address, AES key and words 0, 1 and 2 of user data

```
0, 0x00158d0012345678, 0x0123456789ABCDEF0123456789ABCDEF,  
0x0123456789ABCDEF0123456789ABCDEF,  
0x0123456789ABCDEF0123456789ABCDEF,  
0x0123456789ABCDEF0123456789ABCDEF
```

Used entry with just MAC address

```
1, 0x00158d0012345678
```

For an example of using the `-L` option, refer to [Section 3.13](#).

Chapter 2
Command Reference

3. Example Commands

This chapter contains a number of example commands that can be issued using the JN51xx 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: It is assumed that a target JN51xx-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:

```
$ JN51xxProgrammer.exe -l
```

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

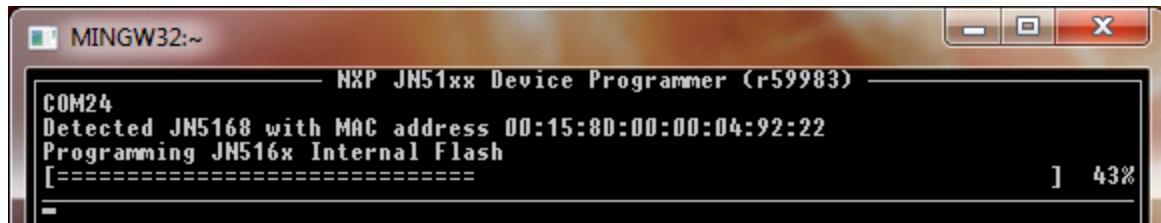
```
Available connections:  
COM24  
COM41
```

3.2 Load Binary File into Flash Memory

The following command uses the `-f` option to load the specified binary file into the Flash memory of the JN51xx device on the serial connection specified using the `-s` option:

```
$ JN51xxProgrammer.exe -s COM24  
-f AN1180_154_HomeSensorCoord_JN5168.bin
```

This results in the following output in the command window:



3.3 Erase Complete Contents of EEPROM

The following command uses the --eepromerase option to erase the complete contents of the EEPROM in the JN51xx device on the serial connection specified using the -s option:

```
$ JN51xxProgrammer.exe -s COM24 --eraseeprom=full
```

This results in the following output in the command window:

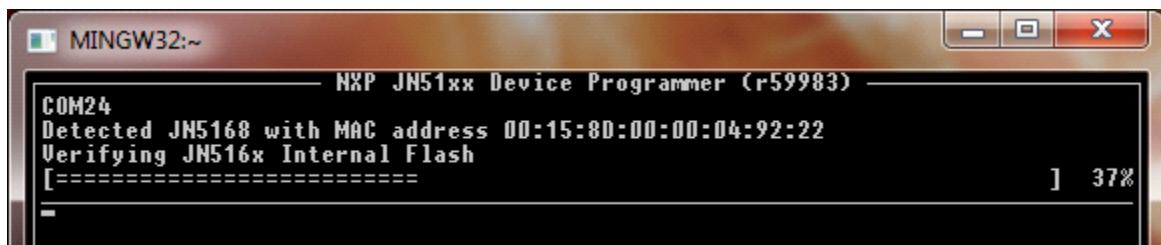


3.4 Load Binary File, Verify Write and Erase EEPROM

The following command uses the -f option to load the specified binary file into the Flash memory of the JN51xx device on the serial connection specified using the -s option, verify this write using the -v option and erase the entire contents of EEPROM using the --eepromerase option:

```
$ JN51xxProgrammer.exe -s COM24  
-f AN1180_154_HomeSensorCoord_JN5168.bin -v --eraseeprom=full
```

This results in the following output in the command window:



3.5 Set MAC Address of JN51xx Device

The following command uses the `--mac` option to set the MAC address of the JN51xx device on the serial connection specified using the `-s` option:

```
$ JN51xxProgrammer.exe -s COM24 --mac 0x00158d0012345678
```

This results in the following output in the command window:



Press Y to confirm or N to abort the operation. Once the operation has been confirmed, the one-time programmable MAC address is written.

3.6 Display the Device Configuration

The following command uses the `--deviceconfig` option to display the configuration of the JN51xx device on the serial connection specified using the `-s` option (the verbosity of the displayed data is set to the lowest level using the `-V` option):

```
$ JN51xxProgrammer.exe -s COM24 --deviceconfig -V 0
```

This will yield results in the following format:

```
COM24: Detected JN5168 with MAC address 00:15:8D:00:00:32:DB:4E
COM24: Reading device configuration
COM24: Device configuration:
JTAG_ENABLE,VBO_200,CRP_LEVEL0,EXTERNAL_FLASH_NOT_ENCRYPTED,
EXTERNAL_FLASH_LOAD_ENABLE
```

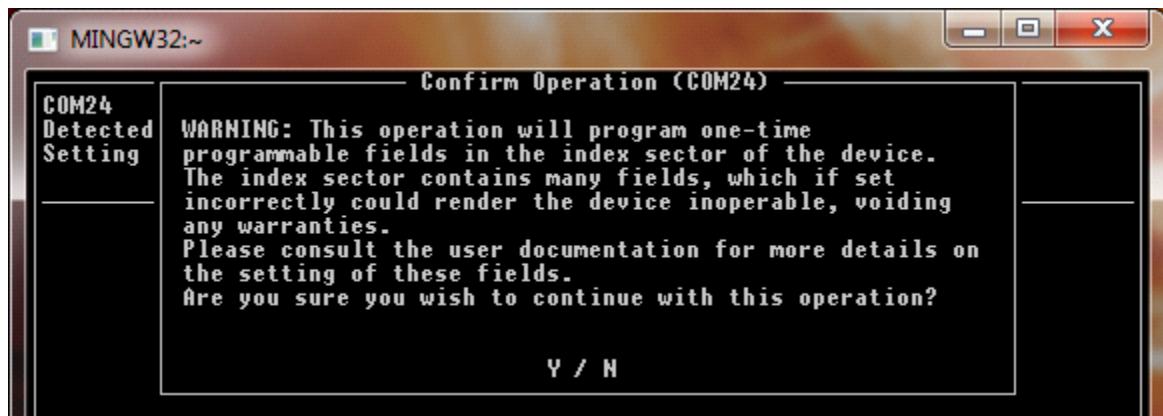
3.7 Set the Device Configuration and AES Key

The following command uses the `--deviceconfig=` option to set the device configuration and the `--key` option to set the AES key of the JN51xx device on the serial connection specified using the `-s` option:

```
$ JN51xxProgrammer.exe -s COM24  
    --key 0x0123456789ABCDEF0123456789ABCDEF  
    --deviceconfig=VBO_195,CRP_LEVEL2
```

The above device configuration sets the brown-out voltage to 1.95V and the bootloader code protection to level 2 (bootloader access to the device is disabled). All other device configuration parameters are set to their default values.

This results in the following output in the command window:



Press Y to confirm or N to abort the operation. Once the operation has been confirmed, the one-time programmable device configuration and AES key are written.

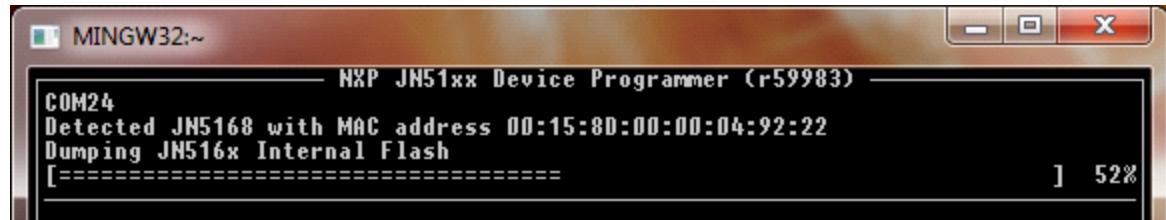
3.8 Dump the Contents of Flash Memory

The following command uses the **-F** option to set save the contents of the Flash memory of the JN51xx device on the serial connection specified using the **-s** option:

```
$ JN51xxProgrammer.exe -s COM24 -F flash.bin
```

The data is saved to the file **flash_00-15-8D-00-00-32-DB-4E.bin** - the filename specified in the command is appended with the MAC address of the JN51xx device.

This results in the following output in the command window:



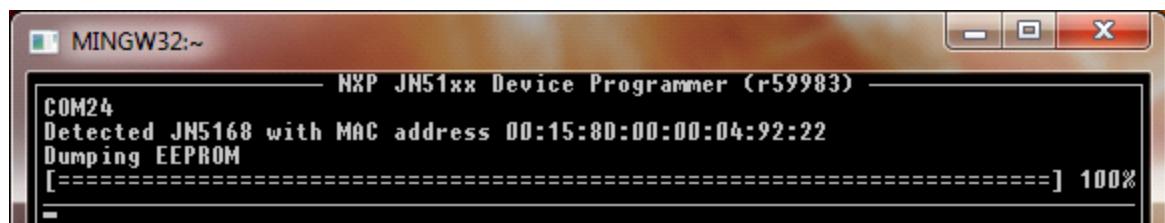
3.9 Dump the Contents of EEPROM

The following command uses the **-E** option to set save the contents of the EEPROM of the JN51xx device on the serial connection specified using the **-s** option:

```
$ JN51xxProgrammer.exe -s COM24 -E eeprom.bin
```

The data is saved to the file **eeprom_00-15-8D-00-00-32-DB-4E.bin** - the filename specified in the command is appended with the MAC address of the JN51xx device.

This results in the following output in the command window.



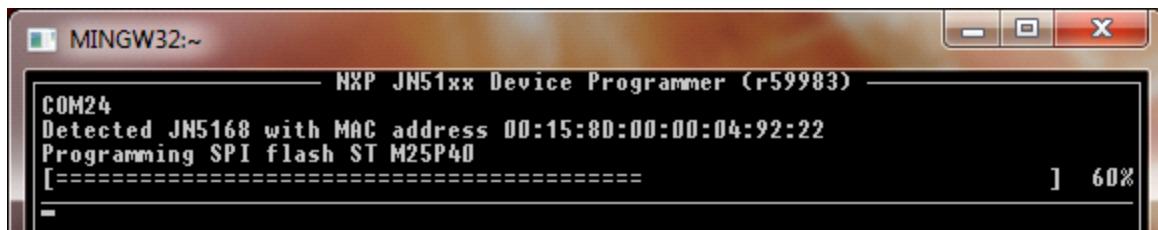
3.10 Switch to External Flash Memory

The JN51xx device contains internal Flash memory but can also have access to an external (SPI-connected) Flash memory device. Either Flash memory can be programmed with this tool and a command option is provided to select which memory is to be accessed. For the JN51xx device, the default is internal Flash memory so this option must be used if access to external Flash memory is required.

The following command uses the --selectflash option to switch access from internal Flash memory to external Flash memory of the JN51xx device on the serial connection specified using the -s option and load the binary file specified using the -f option:

```
$ JN51xxProgrammer.exe -s COM24  
--selectflash=external -f AN1180_154_HomeSensorCoord_JN5168.bin
```

This results in the following output in the command window.



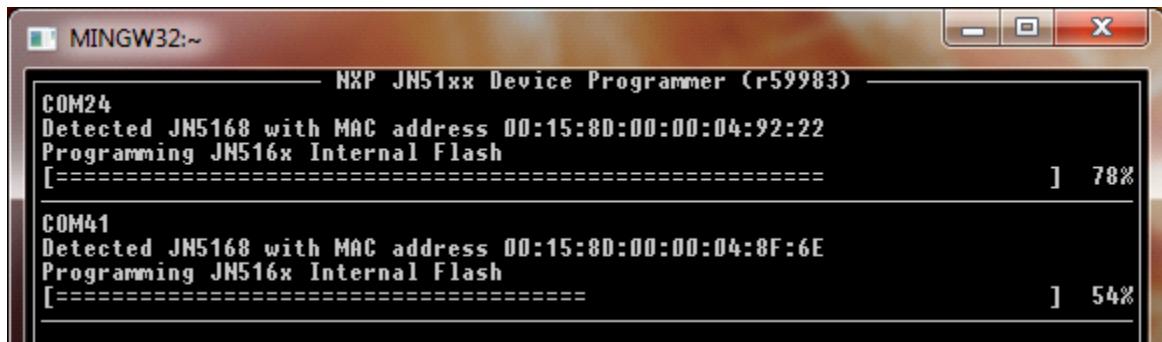
3.11 Program Multiple Devices Simultaneously

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 JN51xx devices. In this case, the `-s` option is used more than once in a single command.

The following command uses the `-f` option to load the specified binary file into the Flash memory of the JN51xx devices on the (two) serial connections specified using the `-s` options:

```
$ JN51xxProgrammer.exe -s COM24 -s COM41  
-f AN1180_154_HomeSensorCoord_JN5168.bin
```

This results in the following output in the command window.

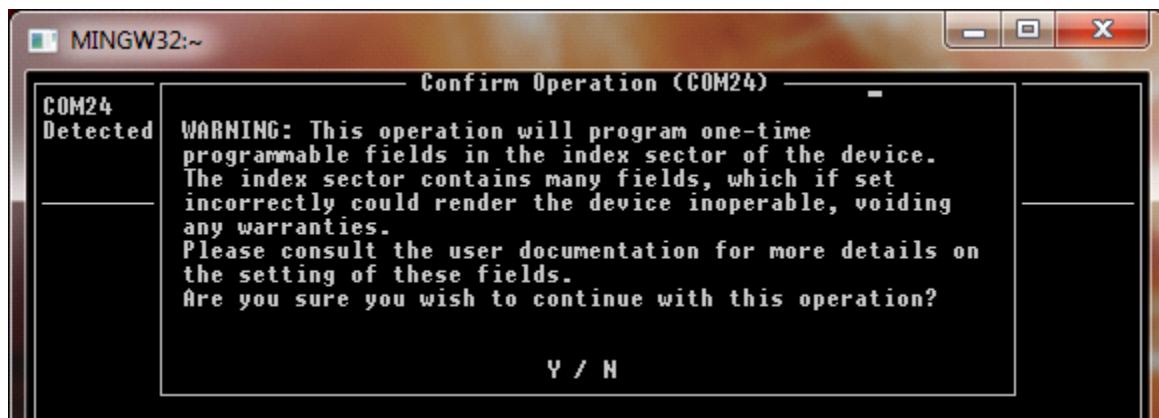


3.12 Program Permanent User Data

The following command uses the `--userdata` option to load the specified 128-bit data string into one word reserved for user data (word 0) in the Flash memory of the JN51xx device on the serial connection specified using the `-s` option:

```
$ JN51xxProgrammer.exe -s COM24  
--userdata 0=0x0123456789ABCDEF0123456789ABCDEF
```

This results in the following output in the command window.



Press Y to confirm or N to abort the operation. Once the operation has been confirmed, the one-time programmable user data is written.

3.13 Specify a License File

The following command uses the `--license` option to specify a license file containing a MAC address (and optionally an AES key and/or user data) to be programmed into the Flash memory of the JN51xx device on the serial connection specified using the `-s` option and load the binary file specified using the `-f` option:

```
$ JN51xxProgrammer.exe -s COM24  
--license license.txt -f AN1180_154_HomeSensorCoord_JN5168.bin
```

For further details on license files and example entries, refer to [Section 2.2.3](#).

3.14 Load and Execute an Application in JN51xx RAM

The following command uses the `-r` option to load the specified binary file directly into RAM of the JN51xx device on the serial connection specified using the `-s` option and then execute the application:

```
$ JN51xxProgrammer.exe -s COM24  
                         -r FlashProgrammerExtension_JN5168.bin
```

This results in the following output in the command window.



Note 1: If used, this option is performed last (irrespective of where it is specified in the option list), since communications with the bootloader will be lost once the application has been loaded and executed.

Note 2: An application loaded using this option is not written to Flash memory.

Note 3: Once the application has been written to RAM, it will be automatically executed immediately.

3.15 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):

```
$ JN51xxProgrammer.exe -s COM10
```

This results in the following output in the command window.



3.16 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):

```
$ JN51xxProgrammer.exe -s COM24
```

This results in the following output in the command window.



Appendices

A. Identifying PC Communications Port Used

In order to use the JN51xx 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 JN51xx device - this is described below (for Windows 7 and 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 JN51xx Flash Programmer CLI (see [Section 2.1](#)).

Appendices

Revision History

Version	Date	Comments
1.0	13-June-2014	First release
1.1	22-Dec-2015	Updated for JN517x

JN51xx Production Flash Programmer

User Guide

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