RELEASE NOTES

JN-SW-4065-JN516x-JenNet-IP-SDK

Related Products and Versions

This document relates to:

JN-SW-4065-JN516x-JenNet-IP-SDK

CONTENTS

CONTENTS 2

RELEASE SUMMARY 3

1. SDK Software Components 3

2. Supported Products 3

3. Software Installation 4

4. Components and Version Numbers 4

5. Release Details 4

5.1 New Features and Changes 5

5.1.1 Build Environment 5

5.1.2 Peripheral API 5

5.1.3 802.15.4 MAC 5

5.1.4 JenNet/JIP 5

5.1.5 Border Router and Linux components 5

5.1.6 JenNet-IP Smart Home Application Note 6

5.1.7 Tools 6

5.2 Modifications Required 6

5.2.1 Porting Notes 6

5.2.2 Application changes 6

5.3 Known Issues 6

6. Memory Usage 7

7. Bootloader Updater 8

7.1 Confirming Version 8

7.2 Performing Update 8

 © NXP B.V. 2013

RELEASE SUMMARY

1. SDK Software Components

The supplied software (JN-SW-4065) comprises a Software Developer' Kit (SDK) to provide support for the JN516x microcontroller, specifically for the JenNet-IP stack. This SDK must be installed on top of JN-SW-4041 (SDK Toolchain) that is currently available from http://www.nxp.com/techzones/wireless-connectivity/jennet-ip. Please also check for newer versions of JN-SW-4007, the JN51xx Flash programmer.

This is version v940 of the JenNet-IP SDK installer for JN516x and includes:

- IEEE802.15.4 MAC (802.15.4-2003/2006)
- JenNet
- 6LoWPAN (6LP)
- JenNet-IP (JIP)
- Chip and board support for JN516x devices
- JenNet-IP Browser tool
- Recalibration library
- Additional files to enable standalone installation of libraries (JN-SW-4051 installation not required)
- Persistent Data Manager (PDM) for the JN516x, using EEPROM instead of serial Flash memory
- Micro-MAC to build compact limited-functionality 802.15.4 applications for energy harvesting use
- A bootloader update program to update early sample devices

2. Supported Products

The SDK update supports the following NXP products.

Chips	Protocols	Evaluation Kits
JN5168 JN5164	802.15.4 JenNet-IP	JenNet-IP EK001
JN5161	802.15.4	

Please note that this SDK does not include support for the JN514x family of microcontrollers.

 © NXP B.V. 2013

3. Software Installation

The JenNet-IP SDK for the JN516x device can be installed directly from the supplied installer. To choose the installation directory, follow the instructions within the installer.

JN-SW-4065 replaces JN-SW-4056, removing the requirement to install JN-SW-4051 (JenNet-IP for JN514x). If updating a previous installation of JN-SW-4056, please uninstall first, then install JN-SW-4065.

If installing this SDK to update an existing JN5148 SDK on your machine, first back up your Applications development directory and any user-modified files within the SDK directory.

This SDK, JN-SW-4065, is installed on top of JN-SW-4041 (SDK Toolchain), currently available from http://www.nxp.com/techzones/wireless-connectivity/jennet-ip.

Therefore, before installing JN-SW-4065, you must install the SDK Toolchain:

JN-SW-4041-SDK-Toolchain-v1.1.exe

You can then install the JenNet-IP SDK:

JN-SW-4065-JenNet-IP-SDK-Installer-v940.exe

The MD5 checksum for the JN-SW-4065 installer is:

9a873adc3a21890b44feed38c15f1c01

For programming devices, the JN516x Flash Programmer version v1.8.6 (or greater) should be used. This is also available from http://www.nxp.com/techzones/wireless-connectivity/jennet-ip, as JN-SW-4007.

Full SDK installation instructions are provided in the *SDK Installation and User Guide (JN-UG-3064)*, available from http://www.nxp.com/techzones/wireless-connectivity/jennet-ip.

4. Components and Version Numbers

This release includes the following component versions:

Component	Version
IEEE 802.15.4-2006 network layer	55198
JenNet-IP network layers	55462
Bootloader	V6/45532
Production test libraries	1v48/51950

5. Release Details

The following sections provide further details on the features and changes within each software component in the SDK. Changes are relative to the previous v1.0 JN516x release (v857).

 © NXP B.V. 2013

5.1 New Features and Changes

5.1.1 Build Environment

New features are:

- Now includes the Maths library in the build by default, which creates smaller code than using the standard libraries

5.1.2 Peripheral API

New features are:

- Functions for direct access to the JN516x EEPROM have been added. These are to be documented in JN-UG-3087, the JN516x Integrated Peripherals API User Guide

5.1.3 802.15.4 MAC

New features are:

- Time-of-flight library added. If not used, the extra code is automatically discarded from the build
- Encryption and decryption have been made faster

5.1.4 JenNet/JIP

New features are:

- Sleeping end device support added, including:
  - Several new API calls (see User Guide JN-UG-3080)
  - Over network download (OND)
  - Ability for a remote node to request a sleeping device stays awake
  - JIP C and Java libraries now apply longer timeouts if a device ID indicates the device can sleep
- Dynamic profile support
  - Network parameters can be set to be changed automatically on a running network without disrupting it, as the network grows or shrinks
- Fast commissioning in Standalone mode: enables factory pre-commissioning
- 'Low Energy' support for energy harvesting devices with limited power capability

5.1.5 Border Router and Linux components

Border Router and Linux components have been updated in the Application Note JN-AN-1110 (JenNet-IP Border-Router), including:

- Support for dynamic profiles
- Ability to register Low Energy devices
- Support for Raspberry Pi build

 © NXP B.V. 2013

5.1.6 JenNet-IP Smart Home Application Note

The Smart Home application note has been split into a template (JN-AN-1190) and the applications (JN-AN-1162). A low energy application for energy harvesting devices has been added, plus several sensor applications.

5.1.7 Tools

The OND tool has been updated.

5.2 Modifications Required

5.2.1 Porting Notes

Moving from a previous JN516x release to this one requires no special porting instructions, though section 5.2.2 should be consulted for any application changes required.

5.2.2 Application changes

The following application changes are required:

- The term “Gateway started” is now known as “Network announce”. This affects the following:
  - Use of E_STACK_GATEWAY_STARTED should be replaced by E_STACK_NETWORK_ANNOUNCE
  - The function eApi_SendGatewayStarted() should be replaced by eApi_SendNetworkAnnounce()
- The function v6LP_Tick() has been deprecated. It still works but should be replaced by vJIP_Tick() for future compatibility

5.3 Known Issues

The following known issues are present in this release

Severity	Description
Medium	**Issue**: Occasional multicast/groupcast transmission failure **Workaround**: None – system recovers by itself
High	**Issue**: Large network takes long time to reform, after reset via OND or quick Co-ordinator reset **Workaround**: None – network does reform, but time taken is outside desired target
Medium	**Issue**: Traps on MIB variables are not supported **Workaround**: None – traps should not be used, and references to them in the user documentation should be ignored
High	**Issue**: When using Linksys WRT160NL Border Router, wi-fi drops out after 12 hours uptime (known issue with OpenWRT on this device) **Workaround**: Reset device, or use one of the other supported hardware platforms

 © NXP B.V. 2013

Medium	**Issue**: Low Energy frame causing routers to reset in Standalone mode **Workaround**: Low Energy devices cannot be added to a network in Standalone mode. Operation in Gateway mode is not affected
High	**Issue**: libJIP discovers sleeping end device with zero MIBs **Workaround**: Use UDP/IP rather than TCP/IP when communicating with sleeping end devices. Use short sleep durations on end devices. If problem is observed, reset discovering device
High	**Issue**: libJIP fails to filter out-of-order responses. When communicating with sleeping end devices and a request is timed out before a response is received, subsequent request may accept response from original request **Workaround**: Use UDP/IP rather than TCP/IP when communicating with end devices. Use short sleep durations on end devices
Medium	**Issue**: OND server should send out newer revision of requested image if available **Workaround**: Only a problem for end devices. Ensure that revision number of new image is only 1 higher than revision number of currently running image

6. Memory Usage

Memory requirements are estimated for a JenNet-IP node with a typical configuration of:

- 100 routing table entries (each routing table entry requires 16 bytes)
- 2 OND images
- 2 IPv6 buffers
- 2 UDP sockets
- 4 traps
- 2Kbyte processor stack

On a JN5168, the stack (MAC, JenNet, 6LP, JIP, OND) for this configuration requires:

- Flash: 106200 bytes approximately
- RAM: 18700 bytes approximately

This was calculated to include space taken from the heap and the processor stack. It can be seen that for a JN5168 this would leave approximately 24872 bytes of Flash and 14068 bytes of RAM for an application, assuming that the application has to fit into 128 Kbytes to enable a second OND image to fit. For a JN5164, there will be an extra 560 bytes of RAM available due to OND not having to claim some workspace.

For EEPROM usage, it is difficult to provide a figure, as the stack on the JN5168 uses 11 sectors for OND (out of 63), but it is necessary to leave some ‘headroom’ so that updates can be written into new sectors before the old data is deleted. A rough ‘rule of thumb’ might be to have as much space again - so that would be 22 sectors. This leaves 41 sectors for the application, so about 20 in use at any one time. The current application uses 14 sectors. Each sector is 64 bytes, though PDM takes 16 of those.

© NXP B.V. 2013

7. Bootloader Updater

Early engineering samples of the JN516x included an older bootloader that must be updated to the latest version.

7.1 Confirming Version

To determine if an update is required, connect the JN516x for programming with the JN51xx Flash Programmer v1.8.6 (or newer) and press the 'Refresh' button in the Flash Programmer GUI.

If it reports:

Device: JN5168, BL: 0x128F0000

then the bootloader needs to be updated.

If it reports:

Device: JN5168, BL: 0x00080003 or **Device: JN5168, BL: 0x00080006**

then the bootloader has already been updated, and a further update is not required.

7.2 Performing Update

To update the bootloader follow the steps below:

1. Using the JN51xx Flash Programmer v1.8.6 (or newer), program the **BootloaderUpdate_JN5168.bin** installed by this SDK at **C:\Jennic\Tools\BootloaderUpdater\BootLoaderUpdater_JN5168.bin** to the device. The Flash Programmer will report:

   "The firmware file was built for another processor version and might not run on the connected device. Are you sure you want to continue?"

   Select 'Yes' and then wait for the programming operation to complete.
2. Close the Flash Programmer application
3. Start a terminal emulator application (for example, HyperTerminal) and configure it to use the serial port that the JN516x device is connected to, with port settings of 115200 baud, 8 bits, no parity, 1 stop bit
4. Reset the JN516x device. Do not press reset a second time or disconnect power during the update process. Wait until the serial output on the terminal emulator indicates the update has completed. If there is no serial output (e.g. due to mis-configuration of the terminal emulator), do not try to reset a second time but wait for at least 10 seconds before re-trying
5. Once the bootloader has been updated, restart the Flash Programmer, press the 'Refresh' button and confirm that it now reports:

   Device: JN5168, BL: 0x00080006

Note: The bootloader update is only required for engineering sample parts and the update will only be required once per part.

 © NXP B.V. 2013