



RELEASE NOTES

JN517x ZigBee 3.0 SDK

JN-SW-4270

Build 1841

NXP Semiconductors

For the contact details of your local NXP office or distributor, refer to:

www.nxp.com

CONTENTS

RELEASE SUMMARY (v1841)	3
1. Software Components	3
2. Supported Hardware Products	4
3. Installation	4
3.1 LPCXpresso Installation	4
3.2 SDK Plug-in Installation	5
4. Release Details	5
4.1 New Features and Changes	5
4.2 Bug Fixes	6
4.3 Modifications Required	9
4.3.1 Porting to R22 stack:	11
4.4 Known Issues	14
5. Application Porting	14
6. Related Documentation	14
RELEASE HISTORY (v1520)	15
7. Release Details	15
7.1 New Features and Changes	15
7.2 Bug Fixes	15
7.3 Modifications Required	15
7.4 Known Issues	15
RELEASE HISTORY (v1483)	16
8. Release Details	16
8.1 New Features and Changes	16
8.2 Bug Fixes	16
8.3 Modifications Required	17
8.4 Known Issues	17
RELEASE HISTORY (v1446)	18
9. Release Details	18
9.1 New Features and Changes	18
9.2 Bug Fixes	18
9.3 Modifications Required	18
9.4 Known Issues	18
RELEASE HISTORY (v1433)	19
10. Release Details	19
10.1 New Features and Changes	19
10.2 Bug Fixes	19
10.3 Modifications Required	19
10.4 Known Issues	19

RELEASE SUMMARY (v1841)

The JN517x ZigBee 3.0 Software Developer's Kit (JN-SW-4270) contains software resources needed to develop ZigBee 3.0 applications for the NXP JN5179, JN5178 and JN5174 wireless microcontrollers on Windows and Linux platforms. This SDK must be installed on top of the LPCXpresso toolchain (see Section 3).

1. Software Components

This release of the JN-SW-4270 software includes the components listed in the SW_Contents_Register_JN-SW-4270.txt

2. Supported Hardware Products

This software release supports the following hardware products:

Chips	Modules	Evaluation Kits
JN5179-001	JN5179-M10 JN5179-M13 JN5179-M16	JN517x-DK005
JN5178-001		
JN5174-001		

3. Installation

This SDK (JN-SW-4270) must be installed on top of the Eclipse-based LPCXpresso. The SDK is distributed as an Eclipse plug-in archive for LPCXpresso.

3.1 LPCXpresso Installation

LPCXpresso can be obtained from the following NXP web page:

www.nxp.com/lpcxpresso

In order to develop JN517x applications without limitation, we recommend that you purchase the Pro edition of LPCXpresso.

The required version of LPCXpresso for this SDK release is:

LPCXpresso IDE v7.9.2 build 493 (LPCXpresso_7.9.2_493)

Important: This is the version with which the libraries within the SDK were compiled and verified. Other versions of LPCXpresso may not be compatible with the contents of the SDK and cannot be guaranteed to work or be supported with the JN51xx devices.

To obtain LPCXpresso and install it on your development machine:

1. If you do not already have a web account with NXP, navigate to www.nxp.com and create an account.
2. Sign in to your NXP web account.
3. Navigate to the page www.nxp.com/lpcxpresso.
4. Select the **Downloads** tab and then click the **Download** button.
5. Check whether the displayed version is the recommended version indicated above:
 - If it is the recommended version, download it.
 - If it is not the recommended version, click **Previous** and then select the recommended version and download it.
6. Launch the LPCXpresso installer and follow the on-screen instructions. On Linux platforms, this MUST be done as root. Full installation details are provided in the *LPCXpresso IDE Installation and Licensing Guide*, available on the **Documentation** tab of the above web page.

3.2 SDK Plug-in Installation

Once LPCXpresso is installed, the SDK plug-in archive can be installed as follows:

1. Start LPCXpresso (on Linux platforms, this MUST be done as root).
2. Navigate to the drop-down menu option **Help**, which can be found at the top of the IDE options.
3. In the drop-down menu, select **Help > Install New Software**.
4. In the pop-up dialog box, click the **Add** button.
5. In the resulting **Add Repository** pop-up dialog box, within the **Name** field enter "NXP ZigBee 3.0 v1xxx SDK". Then click the **Archive** button.
6. In the resulting **Repository Archive** pop-up window, navigate to the location where the **JN-SW-4270-ZigBee 3.0-v1841.zip** is present, then click the **OK** button.
7. This will populate the plug-ins list as follows:
 - NXP RFCS JN-SW-4263 Software Development Kit**
 - NXP RFCS JN-SW-4270 Software Development Kit**
 - NXP RFCS JN51xx Flash Programmer**
 - NXP RFCS Terminal**
8. Select all of the above by clicking the **Select All** button. Once all the options are selected, click the **Finish** button.
9. Follow all the subsequent instructions, and accept the terms and conditions and disclaimers.

4. Release Details

4.1 New Features and Changes

The following new features are included in this release:

Internal ID	Description
lpsw8035	Support for install codes to be read from OTP autonomously from the stack.
lpsw8430	Allow the application to disable route discovery when sending data packets

4.2 Bug Fixes

The following issues have been fixed in this release:

**JN517x ZigBee 3.0 SDK
Release Notes**

Internal ID	Description
MCUZIGBEE-1609	GPP Functionality Attribute doesn't have the right bitmap
artf571504	NXP ZCL code has not implemented transitiontime parameter as part of Recall Scene command
MCUZIGBEE-1610	GP frame counter is wrong
MCUZIGBEE-1611	GP doesn't handle reserved and manufacturer specific commands
MCUZIGBEE-1612	Zigbee stack doesn't default back to not using install codes
artf571502	[ZCL] ZCL certification failing due to OO-TC-05S as NXP ZCL code does not handle special case where minimum reporting interval is set to FFFF and maximum reporting interval is set to 0
artf571504	[ZCL] ZB 3.0 certification S-TC-04S failing as NXP ZCL code has not implemented transitiontime parameter as part of Recall Scene command which has been introduced in ZCLr7
MCUZIGBEE-1579	Wrong definition of various macros for BDB Commissioning Mode bit field in bdb_api.h
artf566994	Leave indication should check for device type before sending an APSME update out
artf566995	Run time fragmentation APDU allocation is done in the non default cluster implementation
artf553167	The stack clears the link key on synchronous rejoin failure
artf552149	End Device join results in the Coordinator sending route request to a magic address of 0x00CF
artf550364	R22 Coordinator ignores association request
artf552096	Critical sections unsafe when using high priority interrupts
artf552095	MultiMac support needs to handle HW not present
artf552089	Purge Join resource on authentication timer
artf551817	Suppress Transport key and authentication after Orphan Notification/Re-alignment
artf547808	Applications built using OTA don't successfully validate the image and switch to the new image.
artf545795	Rx On When Idle End device address conflict resolution parent to send leave
artf545793	The API to transmit a transport key message sends the key only APS encrypted
artf545516	Reset make neighbour table and MAC table misaligned
artf545515	Leave sent from a parent whose address conflicts with the local node
artf545514	User defined default key is not persisted
lpsw8712	The APDU is not getting freed in BDB source code if ZPS function returns non-success value
lpsw8724	Route doesn't recover from a many to one route failure

lpsw8090	The stack set depth function does no copy the new depth to the beacon payload
lpsw8091	Stack form distributed network does not set depth or capacity flags in beacon payload
lpsw8407	Stack network Restore functionality doesn't restore the complete data
lpsw8408	binding table does not align to network address/neighbour table changes
lpsw8409	Rejoin and data poll request are sent to invalid address
lpsw8458	PDUM_vInit function does not set the FreeHead List index to zero
lpsw8492	NPDU leak happens with fragmentation
lpsw8498	Beacon notification on an active scan on the coordinator can result in trampling of memory
lpsw8503	GP alias doesn't get reset after sending GP packet
lpsw8504	NWK address response and IEEE address response for RxOnWhenIdle End device for extended address request is wrong
lpsw7654	BDB_bOutOfBandCommissionGetKey() returns incorrect results
lpsw8667	E_ZCL_CBET_REPORT_REQUEST will be missing if two attributes are reported together
lpsw8668	Add a function to increment PollControl timing parameters when an end device wake up from sleep
lpsw8664	The number of endpoints supported by a device should not be equal to or more than maximum endpoint ID supported
lpsw8665	Endpoint not getting registered if it contains any attribute of datatype Int40, Int48, Int56 , Int64
lpsw8666	Poll Control Cluster: Internal Variable Initialization
lpsw8681	Router and Coordinator do a bus error exception when Network leave request is received
lpsw8682	Coordinator sends a beacon response of PAN at capacity after it receives a unicast management permit join req
lpsw8692	The event ZPS_EVENT_APS_ZGP_DATA_INDICATION is not getting generated at AT_GP application and ZCL for DUT_ZGPD
lpsw8697	The function BDB_vNsStateMachine would set a wrong bdbNodeJoinLinkKeyType attribute in bdb_ns.c file
lpsw8712	The APDU is not getting freed in BDB source code if ZPS function returns non-success value
lpsw8719	A new device can not join even when an old device has left the network when using Flash TCLK feature
lpsw8742	Route doesn't recover from a many to one route failure

4.3 Modifications Required

These modifications must be made otherwise the stack will not function properly.

1. The new stack has support for better throughput and automatic buffering of data packets during route discovery. This requires the addition of a new queue in the application.

The size of the queue is defined as:

```
#define MCPS_DCFM_QUEUE_SIZE    5
```

This queue should be tied to the stack definition:

```
extern PUBLIC tszQueue zps_msgMcpsDcfm;
```

The storage of the queue must be defined:

```
PRIVATE MAC_tsMcpsVsCfmData asMacMcpsDcfm[MCPS_DCFM_QUEUE_SIZE];
```

In the APP_vInitResources function an additional queue must be added:

```
ZQ_vQueueCreate(&zps_msgMcpsDcfm,      MCPS_DCFM_QUEUE_SIZE,  
sizeof(MAC_tsMcpsVsCfmData),(uint8*)asMacMcpsDcfm);
```

2. All references to BDB_vSetKeys should be replaced with ZPS_vSetKeys.
3. The beacon filter functionality has been extended to support filtering on depth. This addition means that the filter mask field in the beacon filter structure has been changed from a u8FilterMap to u16FilterMap. The bit mask doesn't change.
4. The binding table functionality has been optimised to not store the extended address inside the binding table structure. This now has a look up field.

If the destination address mode in the binding table holds the value of 0x3 (IEEE address mode) the destination address can be got using the following reference:

```
ZPS_tsAplAib * tsAplAib = ZPS_psAplAibGetAib();  
u64Addr = ZPS_u64NwkNibGetMappedIeeeAddr(  
ZPS_pvAplZdoGetNwkHandle(), tsAplAib->psAplApsmeAibBindingTable-  
>psAplApsmeBindingTable[0].pvAplApsmeBindingTableEntryForSpSrcAddr[j].u16  
AddrOrLkUp);
```

where j is the index of the binding table entry.

If the destination address mode is (0x1) which refers to the group address then this is stored as a 16 bit value in the tsAplAib->psAplApsmeAibBindingTable-

>psAplApsmeBindingTable[0].pvAplApsmeBindingTableEntryForSpSrcAddr[j].u16
AddrOrLkUp

5. There is a change in the stack behaviour during rejoins.

When a node does a rejoin with the allocate address bit set, In the previous revisions of the stack it would get a new address assigned.

The behaviour of the stack is now to only issue a new address if the address the device has allocated to itself is conflicting with something the parent has.

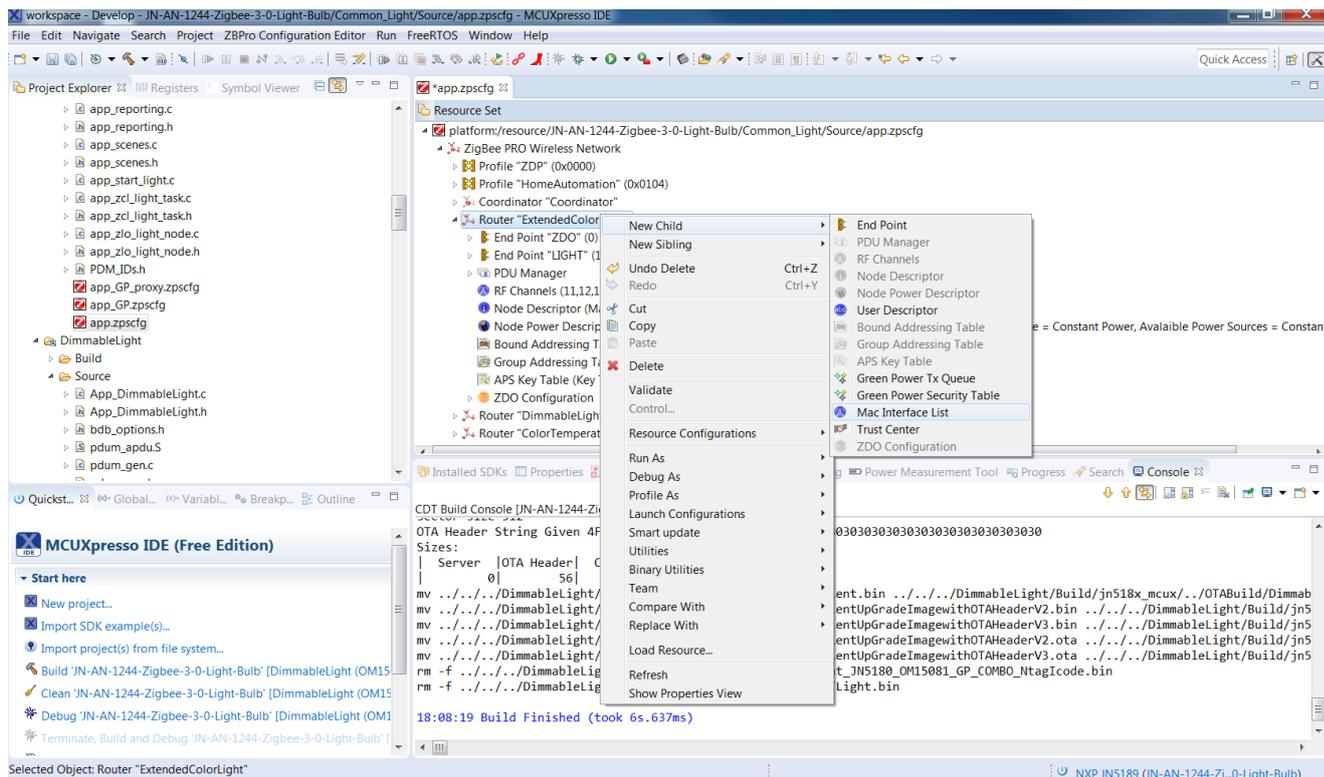
The device would be issued the same address it selected for itself if it wasn't an illegal address i.e. 0 or greater than 0xffff7 or conflicting with current addresses stored by the parent.

6. The [void ZPS_vSetOrphanUpdateDisable (bool_t bEnableOverride)] is no longer supported. This is because the stack by default treats the orphan join as a secured rejoin onto the network and hence doesn't send APSME updates or transport key to the orphaned device.

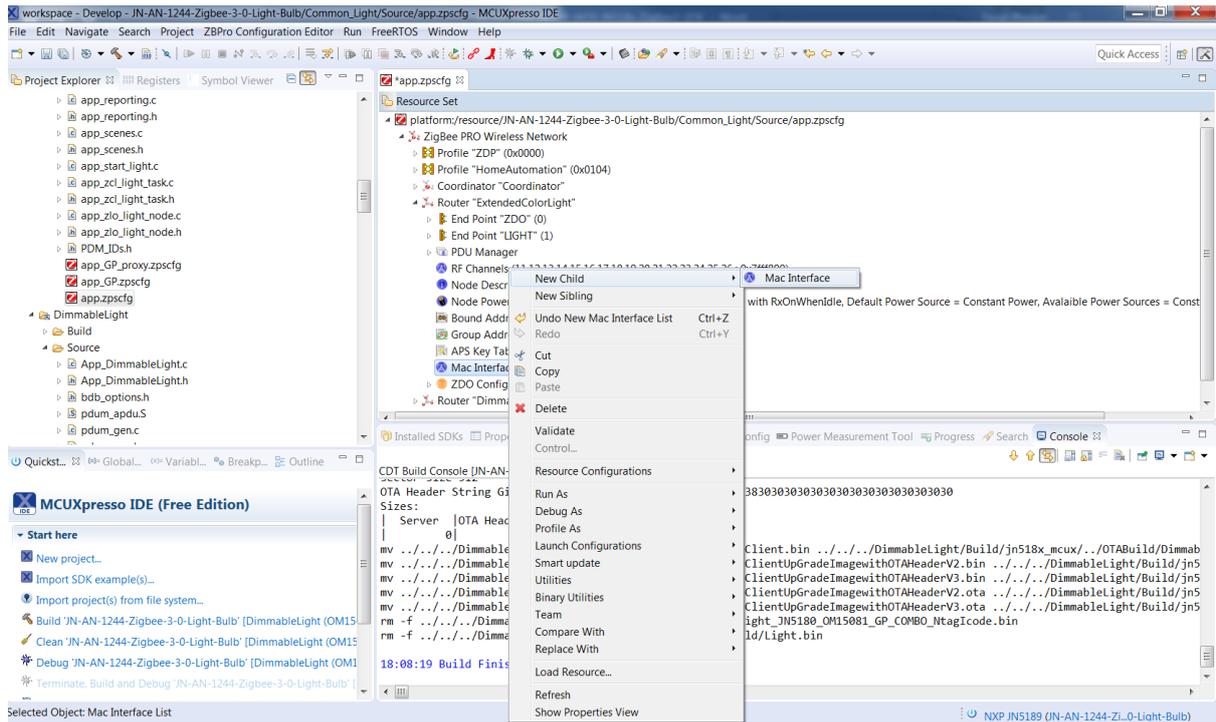
4.3.1 Porting to R22 stack:

The Zigbee PRO R22 version of the stack allows for multiple MAC interfaces to be present. This is to support both 2.4G and 868 MHz frequency bands using the single Zigbee stack. To address this a MAC interface table needs to be configured in the ZPS Config diagram.

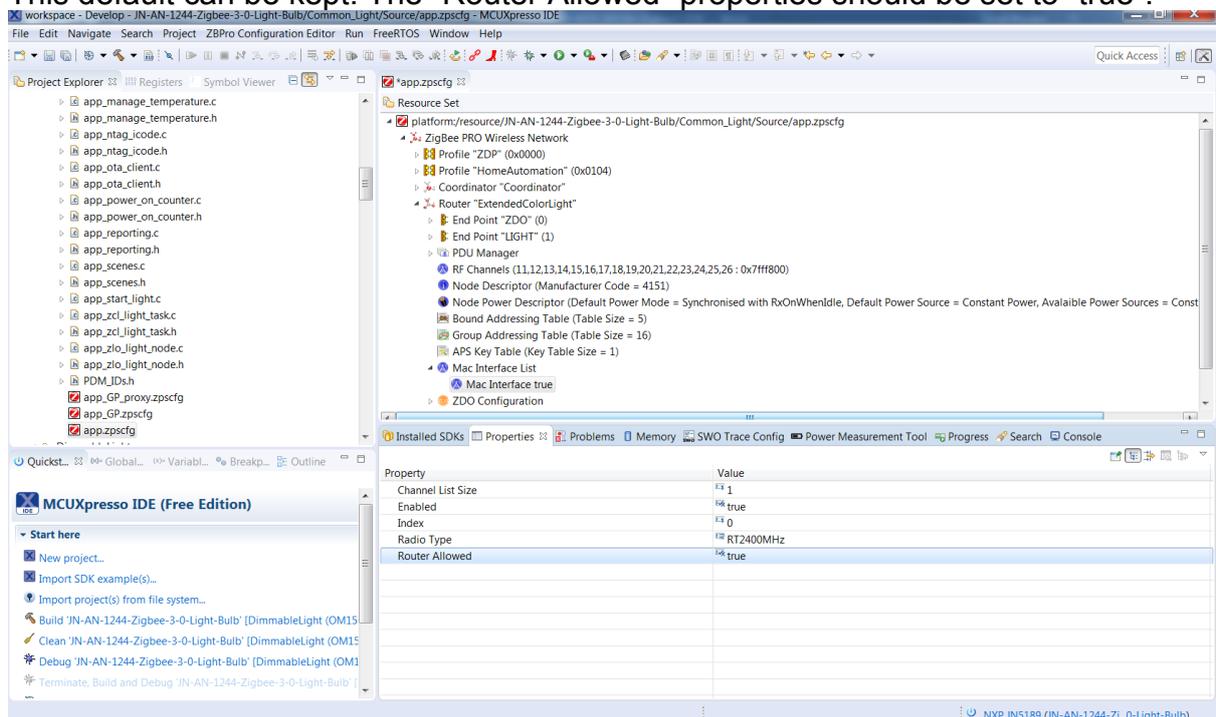
The Mac Interface list can be found as an option for the node. For e.g. if you have Zigbee network with a router node. You can select the router node and press the right mouse button to provide the options. The Mac Interface list can be found under New Child -> Mac Interface List.



After adding the Mac Interface List, select the Mac Interface list and press the right mouse button to provide the options. The Mac interface can be found under New Child -> Mac Interface.



After adding the Mac Interface, the properties can be updated. The default is 2.4G. This default can be kept. The “Router Allowed” properties should be set to “true”.



In the config diagram remove any reference to ChannelStructure
e.g.

```
<ChannelStructure FrequencyBand="2.4GHz"  
PageChannelBitmask="0x7FFFF80"/>
```

Add add

```
<MacInterfaceList>  
  <MacInterface ChannelListSize="1" index="0" RadioType="RT2400MHz"  
  Enabled="true"/>  
</MacInterfaceList>
```

before the ZDOservers

Any reference to `psActvNtEntry->uAncAttrs.bfBitfields.u3Age` should be changed to
`psActvNtEntry->u8Age`

The channel mask in the AIB is an array so this needs to change from
`psAib->u32ApsChannelMask`

to

```
psAib->pau32ApsChannelMask[0]
```

The channel mask is no longer persisted by the stack. This needs to be persisted by
the application.

local address no longer stored in the binding table

```
psAplApsmeBindingTable->u64SourceAddress
```

use

```
ZPS_u64AplZdoGetleeeAddr()
```

Any reference to `MAC_vHandleMcpsVsReqRsp` should now be changed to

```
PUBLIC void ZPS_vMacHandleMcpsVsReqRsp(void* pvMac,
```

```
MAC_tsMcpsVsReqRsp
```

```
*psMcpsVsReqRsp,
```

```
MAC_tsMcpsVsSyncCfm
```

```
*psMcpsVsSyncCfm)
```

The first element should be `NULL`.

The JPT library should be linked into each project build. The Makefile should be updated
with `LDLIBS += JPT_$(JENNIC_CHIP)`

The `PUBLIC uint32 zps_eSocMacSetTxBuffers (uint8 u8MaxTxBuffers)` has changed to
`PUBLIC uint32 ZPS_u32MacSetTxBuffers (uint8 u8MaxTxBuffers)`

The TC callback prototype has changed to
`bool vNfTcCallback (uint16 u16ShortAddress,
uint64 u64DeviceAddress,
uint64 u64ParentAddress,
uint8 u8Status,
uint16 u16MaclD)`

4.4 Known Issues

The following issues are known in this release:

Internal ID	Severity	Description
lpsw7828	Low	During OTA upgrade on large networks (>150 nodes), it has been observed that some of the nodes reset with a bus exception. Devices recover and will automatically rejoin the network after reset, with only minimal disruption. If an OTA upgrade is in progress on the node, this will restart after the last valid block received.
lpsw7994	High	The vAHI_TimerDioControl() function prototype still exists in the JN516x Integrated Peripherals API but the function is obsolete. Therefore, do not use this function.

5. Application Porting

Applications developed using the NXP JN516x ZigBee Home Automation/Light Link SDK (JN-SW-4168) can be migrated to ZigBee 3.0 on JN517x. Guidance on the migration of applications to ZigBee 3.0 on JN517x is provided in the Application Note *ZigBee 3.0 Migration Guidelines for JN517x (JN-AN-1236)*.

ZigBee 3.0 applications no longer require the NXP proprietary operating system, JenOS. However, the non-OS resources that were included in JenOS (e.g. Persistent Data Manager) are included in the JN51xx Core Utilities (JCU), supplied in the JN517x ZigBee 3.0 SDK.

6. Related Documentation

The following user documentation supports this software release:

- ZigBee 3.0 Stack User Guide [JN-UG-3113]
- ZigBee 3.0 Devices User Guide [JN-UG-3114]
- ZigBee Cluster Library (for ZigBee 3.0) User Guide [JN-UG-3115]
- ZigBee Green Power (for ZigBee 3.0) User Guide [JN-UG-3119]
- JN51xx Core Utilities User Guide [JN-UG-3116]
- JN51xx Production Flash Programmer User Guide [JN-UG-3099]
- ZigBee 3.0 Migration Guidelines for JN517x Application Note [JN-AN-1236]

All the above manuals are available as PDF documents from the [Wireless Connectivity](#) area of the NXP web site.

RELEASE HISTORY (v1520)

7. Release Details

7.1 New Features and Changes

The following new features are included in this release:

Internal ID	Description
lpsw7627	Added mandatory ClusterRevision for ZLL Commissioning / ZLL Utility and any other remaining clusters

7.2 Bug Fixes

The following issues have been fixed in this release:

Internal ID	Description
lpsw7872	Reboot after OTA upgrade time of 0xffffffff causes bus error
lpsw7927	The Touchlink commands that can be received as broadcasts are now restricted

7.3 Modifications Required

No modifications are required in this release.

7.4 Known Issues

The following issues are known in this release:

Internal ID	Severity	Description
lpsw7828	Low	During OTA upgrade on large networks (>150 nodes), it has been observed that some of the nodes reset with a bus exception. Devices recover and will automatically rejoin the network after reset, with only minimal disruption. If an OTA upgrade is in progress on the node, this will restart after the last valid block received.
lpsw7994	High	The vAHL_TimerDioControl() function prototype still exists in the JN516x Integrated Peripherals API but the function is obsolete. Therefore, do not use this function.
lpsw8014	-	When a scene recall with a bad group ID or scene ID is processed, a default response with status Success is sent, when it should be Not Found (0x8b).

RELEASE HISTORY (v1483)

8. Release Details

8.1 New Features and Changes

This is a first release and therefore there are no changes from a previous release.

8.2 Bug Fixes

The following issues have been fixed in this release:

Internal ID	Severity	Description
lpsw7423	Low	When an intermediate node tries to discover a route to a destination within the route discovery window, the route discovery can fail, but recovers on multiple retries.
lpsw7738	Medium	When an End Device sends data directly to a routing device, if the End Device address does not match the address held by the receiving routing device then the End Device should be asked to leave the network. This is not happening.
lpsw7678	Medium	The bit fields sent over-the-air for a power descriptor do not match what is configured on the device.
lpsw7629	High	Device can go to sleep prematurely after a fragmented transfer.
lpsw7630	Low	After an address conflict, duplicate entries are present in the neighbour table
lpsw7657	Medium	The wrong archive tool was used for building some of the libraries. This results in inconsistent symbols.
lpsw7662	High	When an address conflict for an End Device child occurs and is resolved by the parent, the parent loses knowledge of the new resolved address of the End Device and removes it from its tables.
lpsw7748	Medium	The parent of an 'Rx On When Idle=TRUE' End Device child does not issue a leave with re-join when the End Device is aged out. This leaves the End Device orphaned.
lpsw7804	Low	When the first child of a Router parent is an End Device and it gets aged out, the parent sends responses indicating a neighbour table size of 0 to management LQI requests.
lpsw7790	Medium	In a distributed network, the node which forms the network allocates its first child Router device as a parent. This leads to a problem of obeying a leave from the child and leaving the network, and having the 'remove children' flag set in the leave indication.
lpsw7791	Medium	A management leave always results in the target issuing a leave with the 'remove children' flag set.
lpsw7631	High	The stack does not assign a valid address if the allocate address bit is 0 and the device has an invalid short address
lpsw7805	Critical	When an End Device changes its parent and it has Binding table entries for the old parent, it does not keep the address-map pair of the old parent and this results in bound data messages failing to be sent to the previous parent.

lpsw7663	Low	A parent forwards a broadcast packet originating from an End Device child back to the child as part of the passive ack.
lpsw7385	Low	Indirect buffered frame gets transmitted to a sleepy child outside the required timeout interval. This results in the data being missed by the sleepy child as the receiver is off.
lpsw7506	High	The Green Power cluster was not endian-safe and would fail security, and hence prevent commissioning of Green Power devices into the network.
lpsw7314	Low	IEEE 802.15.4 MAC switches off receiver when it receives a broadcast ahead of intended unicast.
lpsw7369 lpsw7395	Medium	OTA upgrade to external Flash memory does not work. Bootloader causes image corruption when copying image from external Flash. This results in the image not booting up. This has been resolved with bootloader update to Version 8.

8.3 Modifications Required

No modifications are required in this release.

8.4 Known Issues

The following issues are known in this release:

Internal ID	Severity	Description
lpsw7828	Low	During OTA upgrade on large networks (>150 nodes), it has been observed that some of the nodes reset with a bus exception. Devices recover and will automatically rejoin the network after reset, with only minimal disruption. If an OTA upgrade is in progress on the node, this will restart after the last valid block received.

RELEASE HISTORY (v1446)

9. Release Details

9.1 New Features and Changes

This is a first release and therefore there are no changes from a previous release.

9.2 Bug Fixes

The following issues have been fixed in this release:

Internal ID	Severity	Description
lpsw7506	High	The Green Power cluster was not endian-safe and would fail security, and hence prevent commissioning of Green Power devices into the network.

9.3 Modifications Required

No modifications are required in this release.

9.4 Known Issues

The following issues are known in this release:

Internal ID	Severity	Description
lpsw7385	Low	Indirect buffered frame gets transmitted to a sleepy child outside the required timeout interval. This results in the data being missed by the sleepy child as the receiver is turned off.
lpsw7369 lpsw7395	Medium	OTA upgrade to external Flash memory does not work. Bootloader causes image corruption when copying image from external Flash. This results in the image not booting up.
lpsw7314	Low	IEEE 802.15.4 MAC switches off receiver when it receives a broadcast ahead of intended unicast. A MAC fix is in place but not in this release. Would recover on subsequent poll.
lpsw7423	Low	When an intermediate node tries to discover a route to a destination within the route discovery window, the route discovery can fail, but recovers on multiple retries.

RELEASE HISTORY (v1433)

10. Release Details

10.1 New Features and Changes

This is a first release and therefore there are no changes from a previous release.

10.2 Bug Fixes

This is a first release and therefore there are no bug fixes from a previous release.

10.3 Modifications Required

No modifications are required in this release.

10.4 Known Issues

The following issues are known in this release:

Internal ID	Severity	Description
lpsw7385	Low	Indirect buffered frame gets transmitted to a sleepy child outside the required timeout interval. This results in the data being missed by the sleepy child as the receiver is turned off.
lpsw7369 lpsw7395	Medium	OTA upgrade to external Flash memory does not work. Bootloader causes image corruption when copying image from external Flash. This results in the image not booting up.
lpsw7314	Low	IEEE 802.15.4 MAC switches off receiver when it receives a broadcast ahead of intended unicast. A MAC fix is in place but not in this release. Would recover on subsequent poll.
lpsw7423	Low	When an intermediate node tries to discover a route to a destination within the route discovery window, the route discovery can fail, but recovers on multiple retries.