AN11676

ZigBee RF4CE OM15012 Application Template

Rev. 1.0 — 3 June 2015 **Application note**

Document information

Info	Content
Keywords	ZigBee, RF4CE, JN516x, OM15012
Abstract	This application note describes the supplied software template for developing ZigBee RF4CE applications for the NXP JN516x family of wireless microcontrollers. The template is based on an example application designed for hardware components from the JN516x-EK003 Evaluation Kit, including the OM15012 remote control unit.

Revision history

Rev	Date	Description
1.0	20150603	Initial version

Contact information

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

1. Introduction

This document accompanies the ZigBee RF4CE application template for the NXP JN516x series of wireless microcontrollers. The C source files for the template are supplied in the ZIP package for this application note. They contain skeleton code which can be used as a starting point for application development.

The template is based on a simple example application which incorporates the essential operations that are required in a ZigBee RF4CE system. The necessary hardware for the application is provided in the JN516x-EK003 ZigBee RF4CE Evaluation Kit.

2. Application overview

The ZigBee RF4CE application template can be used as a basis for developing ZigBee RF4CE applications which are to be run on the NXP JN516x wireless microcontrollers. The template is based on an example ZigBee RF4CE system containing one or more target nodes and one or more controller nodes (a target node can be paired with up to eight controller nodes; a controller node can be paired with up to eight target nodes).

Note: Before starting your ZigBee RF4CE application development using this template, you are advised to familiarize yourself with ZigBee RF4CE concepts and resources by referring to the ZigBee RF4CE Stack User Guide (JN-UG-3074), which is available from the NXP Wireless Connectivity TechZone.

2.1 Hardware

2.1.1 Remote control and RF4CE receiver

The example code is designed to work with the remote control unit and a USB dongle target from the JN516x-EK003 ZigBee RF4CE Evaluation Kit, as follows:

- OM15012 remote control unit
- DR1198 USB dongle

These components are shown below in Fig 1 and Fig 2. The evaluation kit is fully described in the JN516x-EK003 Evaluation Kit User Guide (JN-UG-3106).

Fig 1. NXP remote control - OM15012 remote control and key names

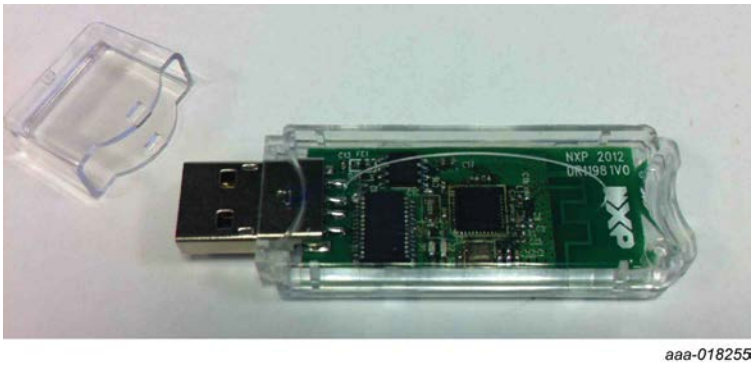

Fig 2. NXP receiver – DR1198 USB dongle

Two dongles are provided in the evaluation kit. One is to be used as an RF4CE receiver, while the other one can be used as either a second RF4CE receiver or a sniffer to spy on RF4CE packets exchanged over the network (for information on sniffer usage, refer to the JN516x-EK003 Evaluation Kit User Guide (JN-UG-3106)).

2.1.2 Programming dongle for remote control unit

A programming dongle (DR1128) is also provided in the evaluation kit. This dongle allows a wired connection between the remote control unit and a PC. The dongle can be used to load an application binary file from the PC into Flash memory in the remote control unit. It also allows information and debug messages from the remote control unit to be displayed on the PC.

The PC is connected to the dongle via the supplied USB-A-to-Mini-B cable. The dongle is connected to the remote control unit via six pins on the dongle that go into a connector at the bottom of the remote control unit via an adaptor.

Before connecting the programming dongle to the remote control unit, the jumper located near the relevant connector on the remote control unit must be removed. The jumper must be in place only when the programming dongle is not connected.

The connection of the programming dongle is shown in Fig 3.

Fig 3. DR1128 programming dongle connection

2.2 Software

The application template comprises two applications – one for a controller node and one for a target node. The controller node is referred to as the remote node and the target node is referred to as the receiver node. Source files and binary files for these applications are provided in the ZIP package of this application note.

- Controller (remote) application:
  - The source file is called remote.c
  - A binary file is provided in one build version for the JN5168 device: Remote_JN5168_OM15012.bin
- Target (receiver) application:
  - The source file is called receiver.c
  - A binary file is provided for the JN5168 device: Receiver_JN5168_EK001.bin

Note: This document assumes that only hardware components supplied in the JN516x-EK003 ZigBee RF4CE Evaluation Kit will be used. In this case, the JN5168 versions of the above binary files will be required.

Each of the above two applications has its own folder in the application note directory structure. In each case, the source files are contained in the Source sub-folder and the binary files are contained in the Build sub-folder. Makefiles are also provided in the Build folders. Eclipse project files are provided in the top level of the directory structure.

In order to develop and build your own ZigBee RF4CE applications, you will need the following NXP Software Developer's Kit (SDK) components:

- JN-SW-4041: JN51xx SDK toolchain
- JN-SW-4060: JN516x ZigBee RF4CE SDK libraries

These SDK components are available from the NXP Wireless Connectivity TechZone. Installation instructions are provided in the SDK Installation and User Guide (JN-UG-3064).

2.3 Compatibility

The software provided with this application note has been designed for use with the following evaluation kits and SDK (Software Developer's Kit) versions:

Table 1. Compatibility

Product Type	Part Number	Version
Evaluation kit	JN516x-EK003	-
SDK libraries	JN-SW-4060	v1014
SDK toolchain	JN-SW-4041	v1.1

2.4 OM15012 remote key mapping

Key descriptions/functions for the remote control unit can be found in file remote.c (tsButton array). These can be adapted (if needed) to match the keypad layout and symbols printed on the physical keys of another remote control.

3. Application functionality

The example application, on which the template is based, provides the following ZigBee RF4CE functionality on the two node types.

3.1 Target (receiver) node

The target node application supports the following features:

- Pairing and unpairing
- Channel agility (always enabled)
- Security (always enabled)
- Periodic power-saving sleep mode (wakes periodically to receive commands)
- Application state and operational information displayed in a terminal emulator on a PC

Further information about these and other features is provided in the following sub-sections.

3.1.1 Pairing (auto-discovery) mode

The target node can be paired with up to eight controller nodes. Pairing information is held in a pairing table on the target node. In order to be paired with a controller node, the target node must be in Pairing (or auto-discovery) mode, which is initiated automatically at power-up.

In this mode, the target node is open to pairing requests from potential partner controller nodes. On receiving such a request, the target node checks the device type and supported profile of the requesting device against internal lists of valid device types and profiles. If matches are found, the target node sends a response and adds an entry for the device to the local pairing table.

By default, the target node is configured to be always open for pairing via the build-time option ALWAYS_OPEN_FOR_PAIRING. In this case, if a ‘discovery request’ is received while the target node is in the running state then (provided that the requesting device is of an appropriate device type and supports an appropriate profile) the target node will automatically send a ‘discovery response’, enter pairing mode and add an entry for the device to the local pairing table.

Alternatively, the time during which the target node will remain in pairing mode can be set to a limited duration (5 seconds by default, but can be configured at build-time via the macro AUTO_DISCOVERY_DURATION) if the macro ALWAYS_OPEN_FOR_PAIRING is defined as FALSE.

In addition, the target node can be configured to reject pairing/discovery requests (from controller nodes) with LQI values lower than a certain threshold. This threshold can be defined at build-time via the macro NIB_ATTR_NWK_DISC_LQI_THRESHOLD.

The above build-time options are described further in section 8.1.

3.1.2 ZigBee RF4CE power saving

ZigBee RF4CE network-layer power-saving mode is implemented on the target node. In this mode, the JN516x receiver is disabled to save power but is periodically activated for a limited period in order to receive commands. This feature is not enabled by default but can be enabled at build-time - the periodicity and the time for which the receiver is active are also configurable at build-time (see section 8.2).

Note: If power saving is enabled on the target node, data packets from the controller node sent during the inactive period of the receiver will not be received. However, since the controller node will retry sending the packets (core stack functionality), the packets will eventually be received.

3.1.3 Displayed information

The target node displays certain information in a terminal emulator (such as Tera Term) on the connected PC. The data is output via one of the UARTs on the JN516x device. The information that can be displayed in the terminal emulator is detailed in section 6.

3.2 Controller (remote) node

The controller node application supports the following features:

- Pairing
- Channel agility (always enabled)
- Security (always enabled)
- Power-saving sleep mode (resulting from inactivity)

Further information about these and other features is provided in the following sub-sections.

3.2.1 Pairing

When a controller node is paired with a target node:

- An entry for the target node is added to the pairing table on the controller node
- An entry for the controller node is added to the pairing table on the target node

In order to pair a controller node with a target node, the target node must be in pairing mode (see section 3.1.1). Then, provided that the controller node is in the running state, a pairing operation can be performed. The controller node can be paired with the target node by holding down the pairing button of the remote control (key K21; see the PAIR function in the tsButton array in remote.c). Once the pairing has successfully completed:

- A paired message, including the pairing reference, will be displayed in the terminal emulator on the PC connected to the target node
- The pairing reference will also be displayed in the top-right corner of the LCD screen (if present) on the controller node
- The LED on the remote control (in the hole near the Power/K34 key) is illuminated for one second if the pairing succeeds or flashes five times if the pairing fails

3.2.2 Power-saving

A power-saving sleep mode is implemented on the controller node, since this node is expected to be self-powered (e.g. through batteries) and go through long periods of inactivity (no button-presses). The JN516x device on the node will automatically enter sleep mode after a certain period of inactivity, which is 60 seconds by default but is configurable at build-time (see section 8.3). During sleep, data held in on-chip RAM is preserved but most internal chip functions are shut down, including the CPU and the majority of on-chip peripherals. The JN516x device will remain in sleep mode until it is woken by user activity. Pressing any key will wake the device and will also perform the function associated with the button. This feature is always enabled.

3.2.3 Reset

A factory-new reset can be performed on the controller node by holding down the DESTROY button (K25; see the DESTROY function in the tsButton array in remote.c) during power-on of the remote (keep button pressed at least 3 seconds after power-on). This deletes all persistently stored context data (application and stack) from non-volatile memory and restores the factory-new settings. It therefore performs a cold start. The LED on the remote control flashes four times to confirm the parameter reset.

4. Installing the application

The contents of the ZIP package of this application note should be extracted to

<SDK_ROOT>\Application

where <SDK_ROOT> is the path into which the JN516x ZigBee RF4CE SDK has been installed.

In order to use the example application with the JN516x-EK003 ZigBee RF4CE Evaluation Kit, you must load the supplied binary files into the relevant kit components as follows:

- Remote_JN5168_OM15012.bin into the OM15012 remote control unit
- Receiver_JN5168_EK001.bin into the DR1198 USB dongle

Instructions for re-programming the firmware in the above evaluation kit components are provided in an appendix of the JN516x-EK003 Evaluation Kit User Guide (JN-UG-3106). You will need to use the JN51xx Flash Programmer tool.

Note: Before loading the application binaries for the first time, you are strongly advised to erase the contents of the EEPROM of the target JN516x device. You can do this using the JN51xx Flash programmer.

5. Using the application

This section describes how to use the example application. It assumes that the application has been loaded into the relevant evaluation kit components, as described in section 4, and then all boards have been powered OFF.

The example application simply allows pairing between the controller and target nodes to take place, and then commands to be sent from the controller node to the target node. The commands are displayed in the terminal emulator on the PC.

To use the example application, follow the instructions below:

1. Connect the target node to a PC by simply plugging the dongle into a USB port of the PC.
2. Determine which COM port on the PC has been used for this connection (it will be labelled “USB Serial Port”) and make a note of its number.
3. Run a terminal emulator program (such as Tera Term) on the PC and configure it to use the COM port for the target node (port number as determined above, bits per second: 115200, data bits: 8, parity: none, stop bits: 1, flow control: none).
4. Power-on the controller node by connecting a power supply (e.g. batteries), but do not press any buttons yet (however, if you want to erase previous pairing information already present in the remote control, press and hold DESTROY button, K25, while connecting the power supply).
5. Press the PAIR button on the controller node (remote control). This will initiate pairing on the controller node. When the target and controller nodes have paired, this will be indicated by a ‘Paired’ message in the terminal emulator on the PC.
6. Send commands from the controller node to the target node by pressing the buttons of the remote. The commands received by the target node will be displayed in the terminal emulator on the PC. The information that can be displayed in the terminal emulator is listed and described in section 6.

Alternatively, you can plug the DR1128 programming dongle into the remote control to display information and debug messages in a terminal emulator (same settings as for the target node connection). In this case, do not use any other power supply (so remove the batteries, if present) as the remote control is powered by the programming dongle.

6. Information displayed on PC

In the example application, the terminal emulator on the PC is used to display certain system information. The displayed information includes:

- ColdStart after a power-up, if no previous pairing or ZigBee RF4CE network membership is found or after a factory-new reset on the target node
- WarmStart after a power-up, if a previous pairing ZigBee RF4CE network membership is found, or after a normal reset on the target node
- <Running> when the target node enters the running mode
- <AutoDiscovery> when the target node is in pairing mode (within the next 5 seconds, by default, a pairing operation can be performed – see below)
- <Pairing> during a pairing operation
- <Unpairing> during an unpairing operation
- Paired when a pairing has been successfully completed (a pairing reference and the IEEE/MAC address of the paired controller node are also displayed)
- <NetworkInPowerSaveMode> when network layer power-saving mode is enabled on the target node
- UserControlPressed on receiving a UserControlPressed command (the pairing reference of the sender and the received command code are also displayed)

On receiving a UserControlPressed command with the DISPLAY_INFO command code, the target node's pairing table is displayed (within the next 5 seconds, by default, a pairing operation can be performed – see above).

An example of information displayed in the terminal emulator is shown below.

```
<AutoDiscovery> <Pairing> <Running-1>
Paired: PairRef 0; MAC 0xAD02000000000000
UserControlPressed: PairRef 0; CmdCode [0x20]
UserControlPressed: PairRef 0; CmdCode [0x21]
<AutoDiscovery> <Pairing> <Running-1>
Paired: PairRef 1; MAC 0xAD01000000000000
UserControlPressed: PairRef 1; CmdCode [0x20]
UserControlPressed: PairRef 1; CmdCode [0x21]


|--------PairingTable----------------|
| PairRef | MAC Address           |
|---------------------------------------|
|    0    | 0xAD02000000000000|
|    1    | 0xAD01000000000000|
|---------------------------------------|
Press '0' to select next PairRef for removal within next
5 seconds:
```

```
Press '1' to remove PairRef 0
Press '1' to remove PairRef 1
Press '1' to remove PairRef 0
Press '1' to remove PairRef 1
<Unpairing-1> <Running-4>
UserControlPressed: PairRef 0; CmdCode [0x20]

|--------PairingTable----------------|
| PairRef | MAC Address           |
|---------------------------------------|
|    0    | 0xAD02000000000000|
|---------------------------------------|
Press '0' to select next PairRef for removal within next
5 seconds:
Press '1' to remove PairRef 0
<Running-5>
```

7. Application structure

This section outlines the basic logic of the application code for the two node types in the example application. This logic is presented in terms of state machine diagrams for the target and controller nodes.

- The target node application is illustrated in section 7.1
- The controller node application is illustrated in section 7.2

7.1 Target node application

The logic of the target node application (receiver.c) is illustrated in the Fig 4.

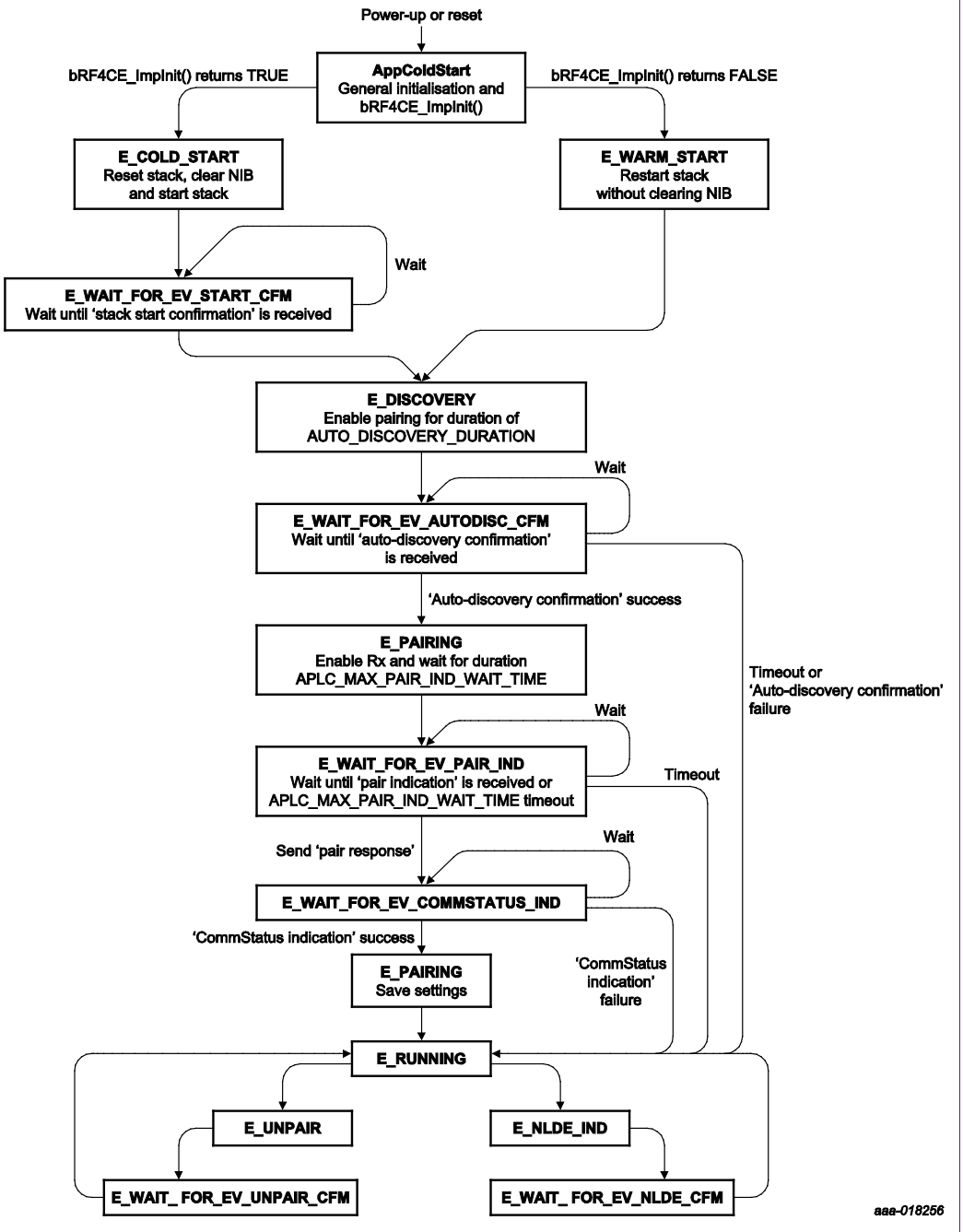

Fig 4. Target node application logic

7.2 Controller node application

The logic of the controller node application (remote.c) is illustrated in the Fig 5.

Fig 5. Controller node application logic

8. Build-time configuration

Certain parameters of the example application can be configured at build-time in the header file config.h. This file is shared by the controller (remote control) and target (receiver) applications, and is located in the Common/Source folder of the application note's directory structure. The following options can be configured in this file.

8.1 Pairing mode on target node

The target node will enter pairing (Auto-discovery) mode immediately after power-on.

By default, the target node is configured to be always open for pairing (see section 3.1.1) by defining the following macro as TRUE:

ALWAYS_OPEN_FOR_PAIRING

Alternatively, by defining the above macro as FALSE, the node can be programmed to remain in pairing mode for a fixed duration. This is 5 seconds by default, but can be configured in config.h by setting the following macro to the desired number of seconds:

AUTO_DISCOVERY_DURATION

When the target node receives a pairing/discovery request from a potential partner controller node, it can use the LQI value of the received packet to decide whether to reject the request – request packets with LQI values below a certain threshold will be ignored. The following macro can be used to enable this feature and define the minimum valid LQI value (below which a request will be ignored):

NIB_ATTR_NWK_DISC_LQI_THRESHOLD

8.2 Power-saving on target node

The ZigBee RF4CE network-layer power-saving feature can be used on the target node. This allows the JN516x receiver to be disabled (as soon as the device enters the running state) and to be periodically activated in order to receive commands from the controller node (see section 3.1.2). This feature is not enabled by default but can be enabled in config.h by setting the following macro to TRUE:

ENABLE_NWK_POWER_SAVING_MODE

The period of a power-saving cycle (the time between consecutively enabling the receiver) can be defined by setting the following macro to a number of 'symbols':

NWK_DUTY_CYCLE

(In the ZigBee RF4CE specification, the term 'duty cycle' is used to mean 'cycle period')

The duration for which the receiver is active within a power-saving cycle can be defined by setting the following macro to a number of 'symbols':

NWK_ACTIVE_PERIOD

The parameters of a power-saving cycle are illustrated in the Fig 6.

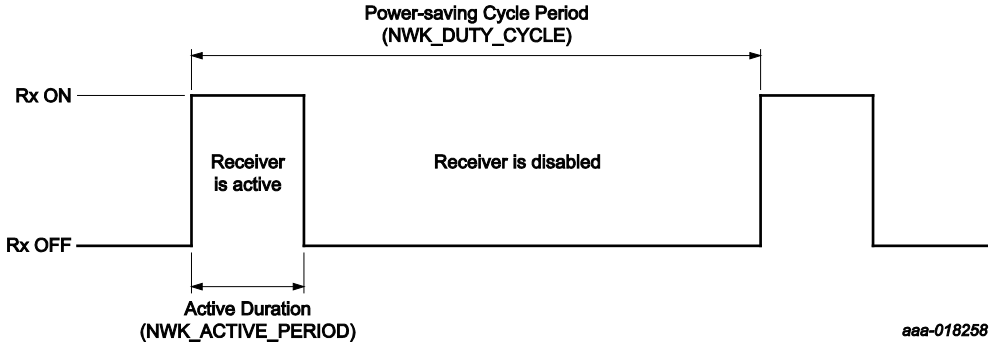

Fig 6. Power-saving parameters

8.3 Power-saving on controller node

The JN516x device on the controller node automatically enters power-saving sleep mode after a certain period of inactivity (see section 3.2.2). This period of inactivity is 60 seconds by default but can be configured in config.h by setting the following macro to the desired number of seconds:

TIME_BEFORE_SLEEP

9. Building the application

This section describes how to build the application provided in this application note. For example, you will need to re-build the application once you have adapted it for your own use. The software can be built only for the JN5168 wireless microcontroller.

The application can be built using the Eclipse IDE or makefiles. These two build methods are described separately in section 9.1 and section 9.2 respectively.

9.1 Using Eclipse

This section describes how to build the application using the Eclipse IDE.

To build the application, follow the instructions below:

1. Ensure that the project directory is located in

   <SDK_ROOT>\Application

   where <SDK_ROOT> is the path into which the JN516x ZigBee RF4CE SDK was installed.
2. Start the Eclipse platform and import the relevant project files (.project and .cproject) as follows:
   a. In Eclipse, follow the menu path File>Import to display the Import dialogue box
   b. In the dialogue box, expand General, select Existing Projects into Workspace and click Next
   c. Enable Select root directory, browse to the Application directory and click OK
   d. In the Projects box, select the project to be imported and click Finish
3. Build an application. To do this, ensure that the project is highlighted in the left panel of Eclipse and use the drop-down list associated with the hammer icon in the Eclipse toolbar to select the relevant build configuration – once selected, the application will automatically build. Repeat this to build the other application.

   The binary files will be created in the relevant Build directories for the applications.

Once an application has been built, you will need to load the resulting binary file into the relevant device. Instructions for re-programming the firmware in the JN516x-EK003 ZigBee RF4CE Evaluation Kit components are provided in an appendix of the JN516x-EK003 Evaluation Kit User Guide (JN-UG-3106).

9.2 Using makefiles

This section describes how to build the application using the supplied makefiles.

The application for each node type has its own Build directory, which contains the makefile for the application.

To build the application, follow the instructions below:

1. Ensure that the project directory is located in

   <SDK_ROOT>\Application

   where <SDK_ROOT> is the path into which the JN516x ZigBee RF4CE SDK was installed.
2. Navigate to the Build directory for the application to be built and follow the instructions below for your target:

Remote control (e.g. OM15012)

At the command prompt, enter:

```
make -f Makefile JENNIC_CHIP=JN5168 EK=EKx TRACE=1 OM=OM15012
```

You can alternatively enter the above command from the top level of the project directory, which will build the binaries for both the applications.

Receiver (dongle)

```
make -f Makefile JENNIC_CHIP=JN5168 EK=EK001
```

In both of the above cases, the binary file will be created in the Build directory.

Once an application has been built, you will need to load the resulting binary file into the relevant device. Instructions for re-programming the firmware in the JN516x-EK003 ZigBee RF4CE Evaluation Kit components are provided in an appendix of the JN516x-EK003 Evaluation Kit User Guide (JN-UG-3106).

10. Legal information

10.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

10.2 Disclaimers

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.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

10.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

11. List of figures

12. List of tables

13. Contents

For more information, visit: http://www.nxp.com

Date of release: 3 June 2015
Document identifier: AN11676