Application Note: JN-AN-1200

ZigBee-RF4CE Application Template

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 NXP JN516x-EK001 Evaluation Kit.

1 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 familiarise 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.

1.1 Hardware

The example code is designed to work with node hardware from the NXP JN516x-EK001 Evaluation Kit, as follows:

- Controller node is a DR1174 Carrier Board with a DR1215 LCD Expansion Board
- Target node is a DR1174 Carrier Board (connected to a PC)

The buttons on the Controller node are used to interact with the Target node and system information is displayed in a terminal emulator (e.g. Tera Term) on the connected PC.

However, note the following:

- The DR1174 Carrier Board used as the Target node does not require an expansion board (but can have any one mounted)
- Alternatively, a DR1198 USB Dongle from the evaluation kit can be used as a Target node
- Alternatively, a DR1174 Carrier Board with a DR1199 Generic Expansion Board can be used as a Controller node, but an LCD screen will not be available to aid user navigation

A ZigBee-RF4CE system which corresponds to the example application is illustrated in the figure below.

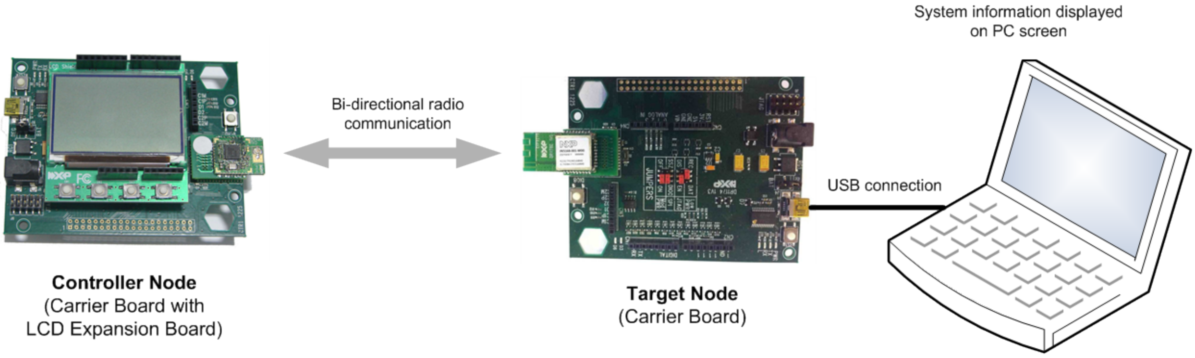
System information displayed on PC screen
Bi-directional radio communication
USB connection
Controller Node
(Carrier Board with LCD Expansion Board)
Target Node
(Carrier Board)

Figure 1: ZigBee-RF4CE System Based on Example Application

1.2 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**
    - Binary files are provided in two build versions for two JN516x device types:
      Remote_JN5168_EK001.bin for the JN5168 device
      Remote_JN5161_EK001.bin for the JN5161 device
- **Target (Receiver) application:**
    - The source file is called **Receiver.c**
    - Binary files are provided in two build versions for two JN516x device types:
      Receiver_JN5168_EK001.bin for the JN5168 device
      Receiver_JN5161_EK001.bin for the JN5161 device

Note: This document assumes that only hardware components supplied in the JN516x-EK001 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).*

1.3 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:

Product Type	Part Number	Version
Evaluation Kit	JN516x-EK001	-
SDK Libraries	JN-SW-4060	v1014
SDK Toolchain	JN-SW-4041	v1.1

2 Application Functionality

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

2.1 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 sub-sections below.

2.1.1 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 and reset
- manually by pressing the DIO8 button on the Carrier Board of the Target node

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.

The time during which the Target node will remain in Pairing mode is 5 seconds by default, but can be configured at build-time via the macro AUTO_DISCOVERY_DURATION.

Alternatively, the Target node can be 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 adds an entry for the device to the local Pairing table.

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 7.1.

2.1.2 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 7.2).

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.

2.1.3 Displayed Information

The Target node displays certain information in a terminal emulator (such as Tera Term) on the connected PC. The connection to the PC is made from the USB Mini B port on the Carrier Board to a USB port on the 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 5.

2.1.4 Resets

Two types of reset can be performed on the Target node, as follows:

- **Factory-new reset:** This reset deletes all context data (e.g. network attributes and pairing data) from memory and restores the factory-new settings. It therefore performs a ‘cold start’. Immediately after this reset, the Target node will enter pairing mode for a limited time (see Section 2.1.1). A factory-new reset is initiated by holding down the DIO8 button and pressing the RST button on the Carrier Board of the Target node.
- **Normal reset:** This reset restores all context data (e.g. network attributes and pairing data) from non-volatile memory. It therefore performs a ‘warm start’. Immediately after this reset, the Target node will enter pairing mode for a limited time (see Section 2.1.1). A normal reset is initiated by pressing the RST button on the Carrier Board of the Target node.

2.2 Controller (Remote) Node

The Controller node application supports the following features:

- Pairing and unpairing
- 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 sub-sections below.

2.2.1 LCD and Buttons

The DR1215 LCD Expansion Board used on the Controller node features an LCD screen and a row of four buttons (SW1-SW4) below the screen. The function of each button is indicated above the button on the LCD screen.

Note: Alternatively, a DR1199 Generic Expansion Board from the JN516x-EK001 Evaluation Kit can be used on the Controller node, but an LCD screen will not be available to aid user navigation.

2.2.2 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 2.1.1). 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 pressing the button SW1. 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
- LEDs D3 and D6 on the Carrier Board of the Controller node will illuminate as follows:
    - D3 flashes and is then extinguished
    - D6 stays illuminated (while the node is paired with at least one Target node)

2.2.3 Unpairing

When a Controller node is unpaired with a Target node, the relevant entries are removed from the Pairing tables on both nodes.

A Controller node can delete its own pairing or any other pairing with the Target node. The unpairing of a (paired) Controller node is performed in three stages:

1. Instruct the Target node to display the list of pairing references of the paired Controller nodes on the PC – this is done by pressing the button SW2 to send a UserControlPressed command with the DISPLAY_INFO command code.
2. Scroll through the list of pairing references using the button SW3 – each button-press results in the transmission of a UserControlPressed command with the ZERO_DIGIT command code, which moves to the next pairing reference in the list.
3. Instruct the Target node to remove the pairing corresponding to the currently displayed pairing reference – this is done by pressing button SW4 to send a UserControlPressed command with the ONE_DIGIT command code. After the pairing has been removed:
    - if there is still at least one pairing entry in the Pairing table on the Controller node, the next pairing reference will be displayed on the LCD panel (if present)
    - if there is no pairing entry remaining in the Pairing table on the Controller node, the pairing screen will be displayed on the LCD panel (if present)

2.2.4 Selecting a Target Node to Control

A Controller node can be paired with up to eight Target nodes. In the case of multiple paired Target nodes, the correct Target node must be selected on the Controller node before sending commands to it. This selection is performed using button DIO8 on the Carrier Board of the Controller node – pressing this button moves the selected Target node to the next one in the Pairing table (on the Controller node). The currently selected Target node is indicated using its pairing reference which is displayed in the top-right corner of the LCD panel (if present) on the Controller node.

2.2.5 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 7.3). 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 of the buttons SW1-SW4 on the LCD Expansion Board will wake the device and will also perform the function associated with the button. This feature is always enabled.

2.2.6 Reset

A factory-new reset can be performed on the Controller node by holding down the DIO8 button and pressing the RST button on the Carrier Board of the node. 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’.

3 Installing the Application

The ZIP package contents 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 (see Section 1.2).

In order to use the example application with the JN516x-EK001 Evaluation Kit, you must load the supplied binary files into the relevant kit components as follows:

- **Remote_JN5168_EK001.bin** into the JN5168 module on the DR1174 Carrier Board with the DR1215 LCD Expansion Board
- **Receiver_JN5168_EK001.bin** into the JN5168 module on another DR1174 Carrier Board or, alternatively, into the JN5168 device on a DR1198 USB Dongle

Instructions for re-programming the firmware in the above evaluation kit components are provided in an appendix of the *JN516x-EK001 Evaluation Kit User Guide (JN-UG-3093)*. 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.

4 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 3, and then all boards have been powered off.

Note: The instructions assume that a DR1174 Carrier Board is used as the Target node, *but alternative instructions for the DR1198 USB Dongle are shown in italics*.

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 connecting the USB Mini B port on the Carrier Board to a USB port on the PC, using a cable provided in the evaluation kit (this step may involve installing the device driver for the cable). *If using the USB Dongle, simply plug the dongle into a USB port of the PC.*

2. Determine which Comms 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 Comms 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) to its Carrier Board, but do not press any buttons yet.

5. Reset the Target node by pressing the RST button on its Carrier Board to put the node into Pairing mode for a limited time (5 seconds, by default). *If using the USB Dongle, reset this device by unplugging it and plugging it back into the PC.*

Note: The pairing duration for the Target node is configurable at build-time in the **config.h** file by setting AUTO_DISCOVERY_DURATION to the desired number of seconds (see Section 7.1).

6. Within the pairing period of the Target node (5 seconds, by default), press the button SW1 on the Expansion Board of the Controller node. 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.

7. Send commands from the Controller node to the Target node by pressing the buttons SW2-SW4 on the Expansion Board of the Controller node. 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 5.

Note: The Target node can be paired with up to eight Controller nodes. To add another pairing, repeat Steps 4 to 6 above.

5 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>
```

6 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 6.1
- The Controller node application is illustrated in Section 6.2

6.1 Target Node Application

The logic of the Target node application (**Receiver.c**) is illustrated in the diagram below.

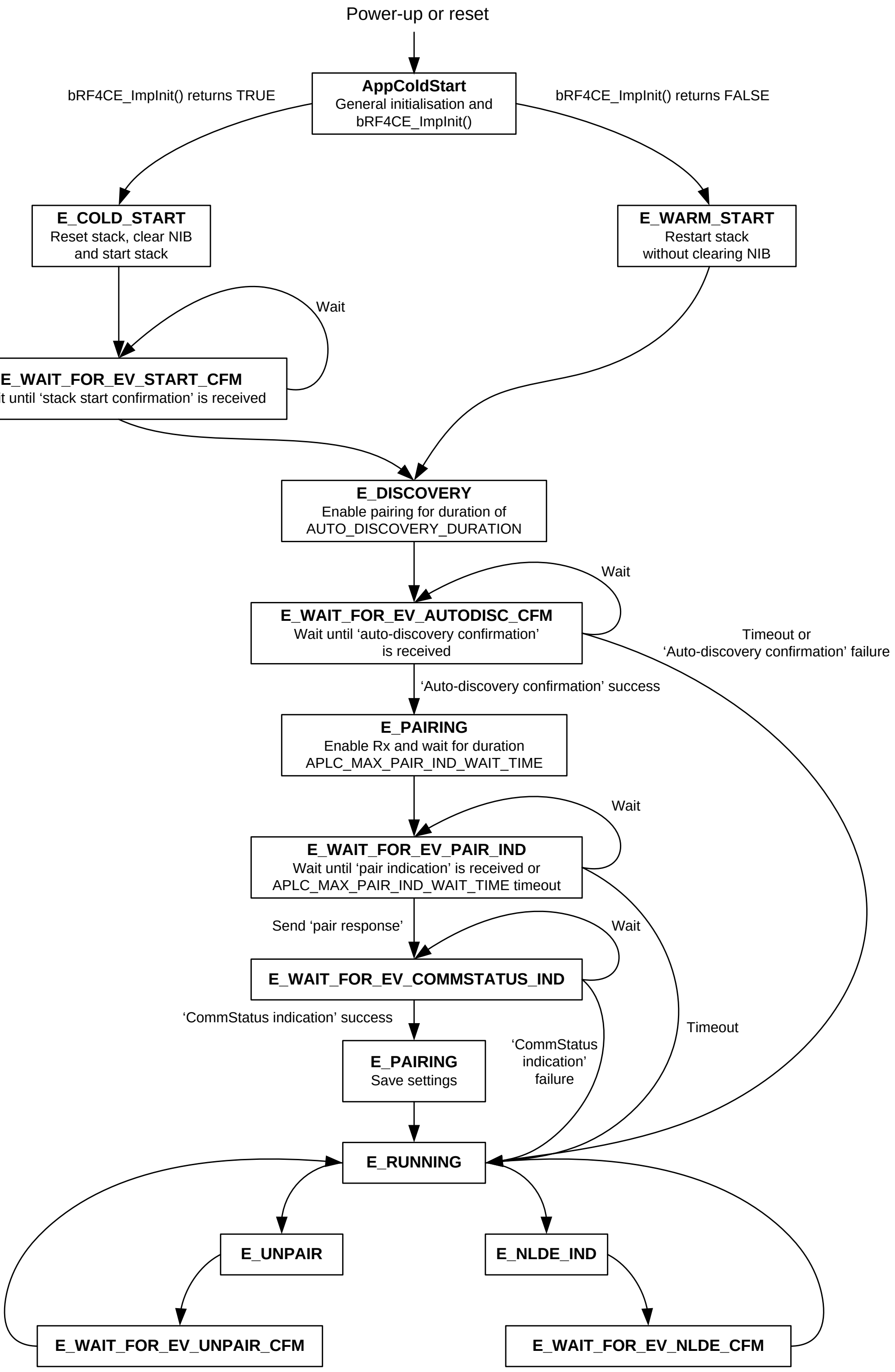
Power-up or reset
AppColdStart
General initialisation and bRF4CE_ImpInit()
bRF4CE_ImpInit() returns TRUE
bRF4CE_ImpInit() returns FALSE
E_COLD_START
Reset stack, clear NIB and start stack
E_WARM_START
Restart stack without clearing NIB
Wait
E_WAIT_FOR_EV_START_CFM
Wait until 'stack start confirmation' is received
E_DISCOVERY
Enable pairing for duration of AUTO_DISCOVERY_DURATION
Wait
E_WAIT_FOR_EV_AUTODISC_CFM
Wait until 'auto-discovery confirmation' is received
Timeout or 'Auto-discovery confirmation' failure
'Auto-discovery confirmation' success
E_PAIRING
Enable Rx and wait for duration APLC_MAX_PAIR_IND_WAIT_TIME
Wait
E_WAIT_FOR_EV_PAIR_IND
Wait until 'pair indication' is received or APLC_MAX_PAIR_IND_WAIT_TIME timeout
Send 'pair response'
Wait
E_WAIT_FOR_EV_COMMSTATUS_IND
'CommStatus indication' success
Timeout
E_PAIRING
Save settings
'CommStatus indication' failure
E_RUNNING
E_UNPAIR
E_NLDE_IND
E_WAIT_FOR_EV_UNPAIR_CFM
E_WAIT_FOR_EV_NLDE_CFM

Figure 2: Target Node Application Logic

6.2 Controller Node Application

The logic of the Controller node application (**Remote.c**) is illustrated in the diagram below.

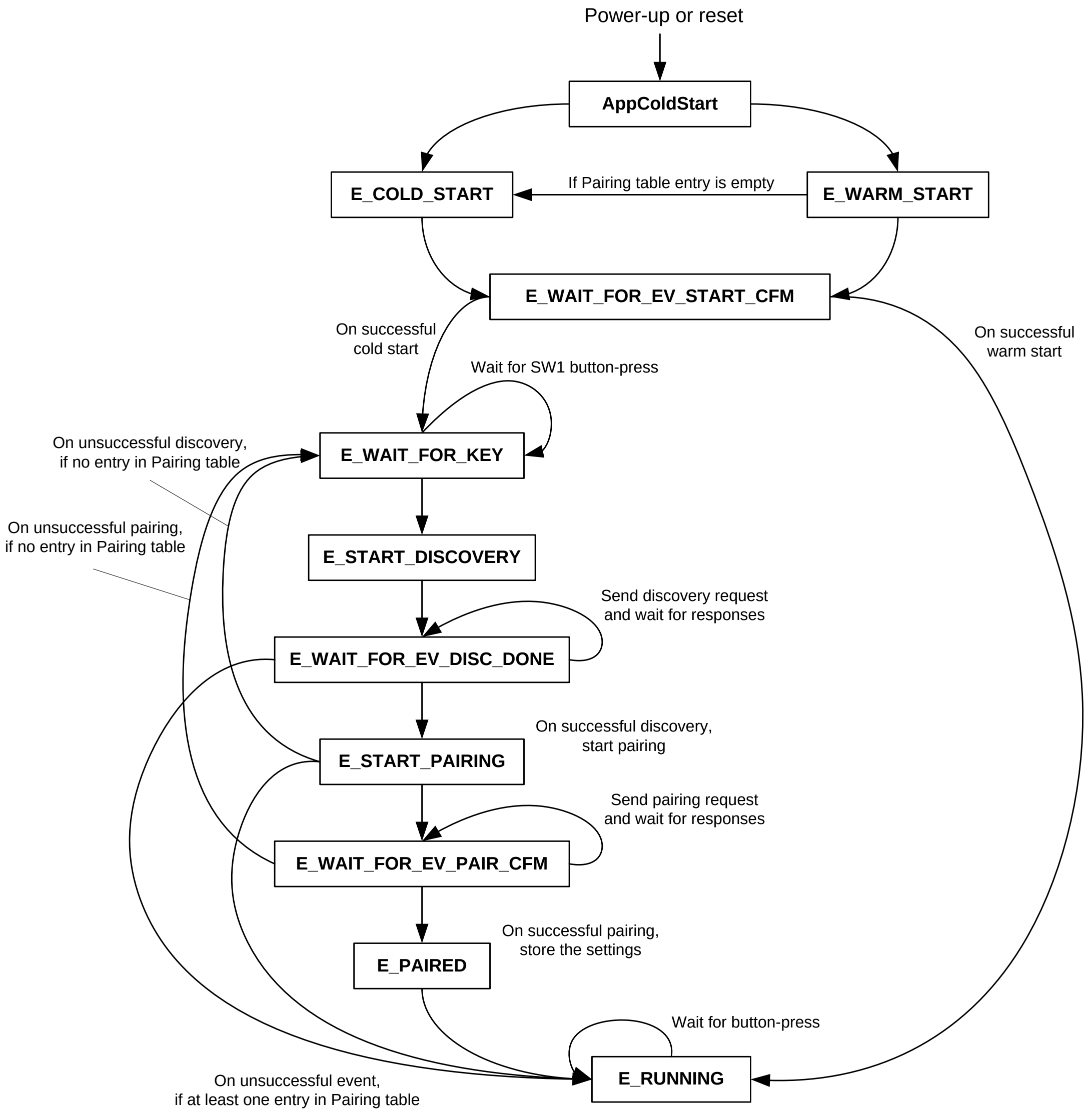
Power-up or reset
AppColdStart
E_COLD_START
If Pairing table entry is empty
E_WARM_START
E_WAIT_FOR_EV_START_CFM
On successful cold start
On successful warm start
Wait for SW1 button-press
On unsuccessful discovery, if no entry in Pairing table
E_WAIT_FOR_KEY
On unsuccessful pairing, if no entry in Pairing table
E_START_DISCOVERY
Send discovery request and wait for responses
E_WAIT_FOR_EV_DISC_DONE
On successful discovery, start pairing
E_START_PAIRING
Send pairing request and wait for responses
E_WAIT_FOR_EV_PAIR_CFM
On successful pairing, store the settings
E_PAIRED
Wait for button-press
On unsuccessful event, if at least one entry in Pairing table
E_RUNNING

Figure 3: Controller Node Application Logic

7 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) 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.

7.1 Pairing Mode on Target Node

The Target node will enter Pairing (Auto-discovery) mode in any of the following situations:

- Immediately after power-on
- Immediately after a reset (normal or factory-new)
- On pressing the DIO8 button on the DR1174 Carrier Board

The node will 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

Alternatively, the Target node can be configured to be always open for pairing (see Section 2.1.1) by defining the following macro as TRUE:

ALWAYS_OPEN_FOR_PAIRING

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

7.2 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 2.1.2). 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 figure below.

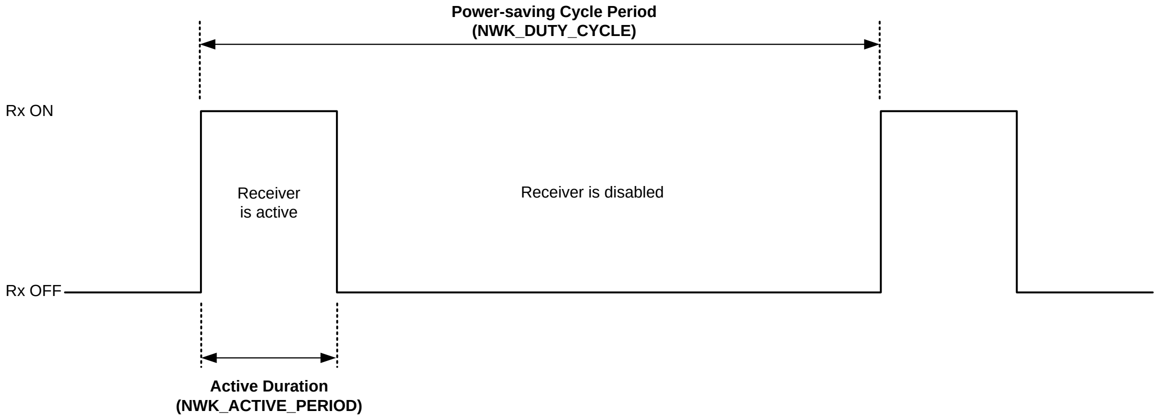

Figure 4: Power-Saving Parameters

7.3 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 2.2.4). 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

8 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 for the JN5168, JN5164 or JN5161 wireless microcontroller.

The application can be built using the Eclipse IDE or makefiles. These two build methods are described separately in the sub-sections below.

8.1 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-EK001 Evaluation Kit components are provided in an appendix of the *JN516x-EK001 Evaluation Kit User Guide (JN-UG-3093)*.

8.2 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 chip type:

   For JN5168

   At the command prompt, enter:

   ```
   make clean all
   ```

   Note that for the JN5168, you can alternatively enter the above command from the top level of the project directory, which will build the binaries for both the applications.

   For JN5164 or JN5161

   At the command prompt, enter:

   ```
   make JENNIC_CHIP=JN5164 clean all
   ```

   or

   ```
   make JENNIC_CHIP=JN5161 clean all
   ```

   In all 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-EK001 Evaluation Kit components are provided in an appendix of the *JN516x-EK001 Evaluation Kit User Guide (JN-UG-3093)*.

Revision History

Version	Notes
1.0	First release

Important Notice

NXP Laboratories UK Ltd
(Formerly Jennic Ltd)
Furnival Street
Sheffield
S1 4QT
United Kingdom

Tel: +44 (0)114 281 2655
Fax: +44 (0)114 281 2951

www.nxp.com