



ZigBee IoT Gateway User Guide

JN-UG-3117

Revision 1.5

24-Mar-2017

Contents

Contents	2
Preface	3
Organisation	3
Acronyms and Abbreviations	3
Related Documents	3
Support Resources	4
Trademarks	4
1 Introduction	5
2 NFC-enabled IoT Application	6
2.1 Raspberry Pi IoT Gateway Hardware Platform	6
2.1.1 Hardware Components	6
2.1.2 Activating the Hardware	7
2.2 Gaining Web Access to Raspberry Pi System	7
3 NFC Commissioning	8
3.1 Commission Node into Network via NFC Reader	8
3.2 Commission Nodes into Network with a Phone	9
3.3 Secure Commissioning	10
3.4 De-commissioning with Phone or NFC Reader	10
4 Web Server CGI Scripts	11
4.1 Index.html	11
4.2 Control Page	11
4.3 Database Viewer	13
4.4 Lamps Page	14
4.5 Plugs Page	15
4.6 Logging Page	17
4.7 Mobile Site	18

Preface

This User Guide describes the fundamentals of the ZigBee IoT Gateway which is included in the JN516x-EK004 Evaluation Kit and JN517x-DK005 Development Kit. The IoT Gateway software is supplied and detailed in the Application Note *ZigBee IoT Gateway for NFC (JN-UG-1222)*.



Note: The manual is included in the Application Note *ZigBee IoT Gateway for NFC (JN-UG-1222)* and should be read in conjunction with the Application Note document.

Organisation

This manual consists of 4 chapters, as follows:

- [Chapter 1](#) gives a high level overview of the use of NFC-enabled gateways
- [Chapter 2](#) outlines the hardware requirements for the IoT gateway
- [Chapter 3](#) describes the NFC commissioning and de-commissioning mechanisms in the gateway
- [Chapter 4](#) details the web interface and the various available windows

Acronyms and Abbreviations

NFC	Near Field Communication
NDEF	NFC Data Exchange Format
NTAG	NFC Tag
API	Application Programming Interface
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
HA	Home Automation
RPI	Raspberry Pi
CGI	Computer Generated Images
ZCB	ZigBee Control Bridge
ZLO	ZigBee Lighting & Occupancy

Related Documents

JN-UG-3112	NFC Commissioning User Guide
JN-AN-1218	ZigBee 3.0 Light Bulbs Application Note
JN-AN-1221	ZigBee HA Lighting with NFC Application Note
JN-AN-1222	ZigBee IoT Gateway with NFC Application Note

Support Resources

To access JN516x and JN517x online support resources (such as SDKs, Application Notes and User Guides), visit the Wireless Connectivity area of the NXP web site:

www.nxp.com/products/interface-and-connectivity/wireless-connectivity

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

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

This manual describes the 'Internet of Things' (IoT) Gateway which runs on the Raspberry Pi hardware platform and is supplied in the NXP JN516x-EK004 and JN517x-DK005 hardware kits. The IoT Gateway is designed for use in ZigBee PRO wireless network and supports the NXP PN7120 Near Field Communication (NFC) reader for commissioning purposes.

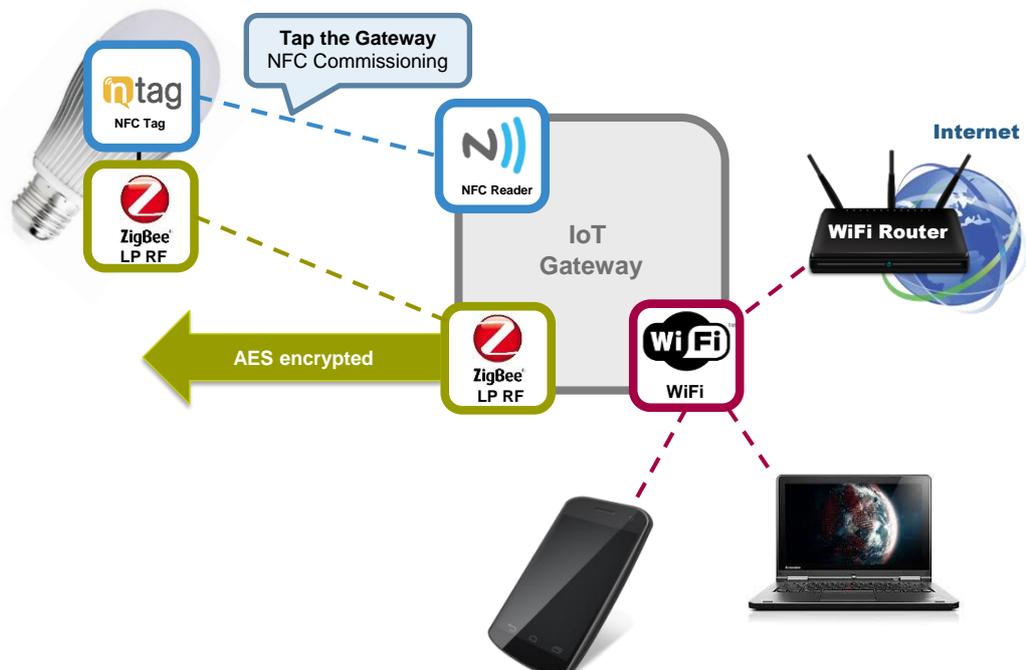
The IoT Gateway software is supplied and detailed in the Application Note *ZigBee IoT Gateway for NFC (JN-AN-1222)*. This software supports the following ZigBee variants:

- ZigBee Home Automation profile (for ZigBee 2.0)
- ZigBee Lighting & Occupancy devices (for ZigBee 3.0)

This User Guide is included in the above Application Note package and describes the:

- gateway software and hardware architecture
- underlying communication and data mechanism
- external interfaces, e.g. the web interface and the socket interface

The related Application Notes *ZigBee HA Lighting with NFC (JN-AN-1221)* and *ZigBee 3.0 Light Bulbs (JN-AN-1218)* illustrate the control of dimmable lights, colour dimmable lights and switches via the IoT Gateway.



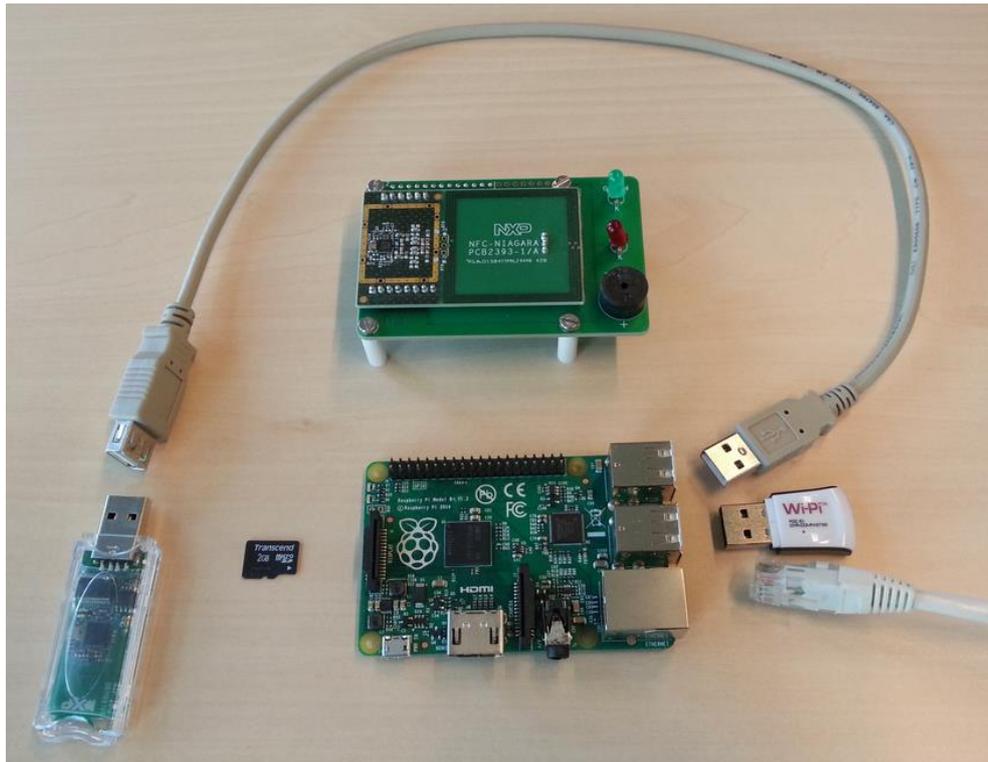
The wireless inter-device communication is done via the ZigBee network. The Coordinator of this ZigBee network resides in the IoT Gateway. This IoT Gateway forms a bridge between the wired IP network and the wireless ZigBee network.

2 NFC-enabled IoT Application

2.1 Raspberry Pi IoT Gateway Hardware Platform

2.1.1 Hardware Components

The Raspberry Pi (RPI) IoT Gateway requires the following building blocks:



Building Block	Description
Raspberry Pi	General purpose application board, powered via micro-USB cable
Micro-SD card	At least 2GB, contains the OpenWRT Linux image
PN7120-based NFC reader extension board	With buzzer and LEDs, needed for commissioning
JN5169 USB dongle	Containing ZigBee Control Bridge firmware (see JN-AN-1223 and JN-AN-1216)
WiPi USB dongle	For Wi-Fi connectivity
Ethernet cable	For software updates via the Cloud

2.1.2 Activating the Hardware

In the directory `Build`, you will find several binary files compressed in ZIP format. You will need to determine which version of the Raspberry Pi board is included in your JN516x/7x hardware kit. The version is printed on the PCB of the Raspberry Pi.

The table below show which zipped disc image to use for each board version.

Board Version	Image to use
Raspberry Pi Model B+ V1.2 (obsolete)	openwrt-brcm2708-bcm2708-sdcard-vfat-ext4.zip
Raspberry Pi 2 Model B V1.1	openwrt-brcm2708-bcm2709-sdcard-vfat-ext4.zip

Get the image that fits your Raspberry Pi board, and unzip it.

The procedure for writing to the SD-card is described in the Raspberry Pi Installation Guide, found at:

www.raspberrypi.org/documentation/installation/installing-images

Windows users need to:

1. Install the “Win32DiskImager” program on a PC.
2. Insert the Raspberry Pi’s SD-card into the PC.
3. Write the image to the SD-card.
4. Insert the SD-card back into the Raspberry Pi.
5. Power up the Raspberry Pi.

2.2 Gaining Web Access to Raspberry Pi System

Web access can be arranged either via the IoT Gateway’s access point or via your own access point.

Via RPI’s access point

Connect the PC (running a web browser) to the wireless network (**SSID: lot_GW_NXP**). The Gateway DHCP server will provide an IP address in the 192.168.2.0 network. The IoT web pages can be found at address **192.168.2.1** (or at <http://iot-gw.nxp>).

Via your own access point

This requires that the Gateway is connected to a LAN via the Ethernet port.

Go to the DHCP page of the access point and note which new device has been added once the Gateway has booted. Assuming the access point allocated the Gateway to address **192.168.1.123**, the web pages will be found at address **192.168.1.123**.

3 NFC Commissioning

The Secure Joiner software plays a key role in the NFC commissioning process. There are two potential clients for the Secure Joiner:

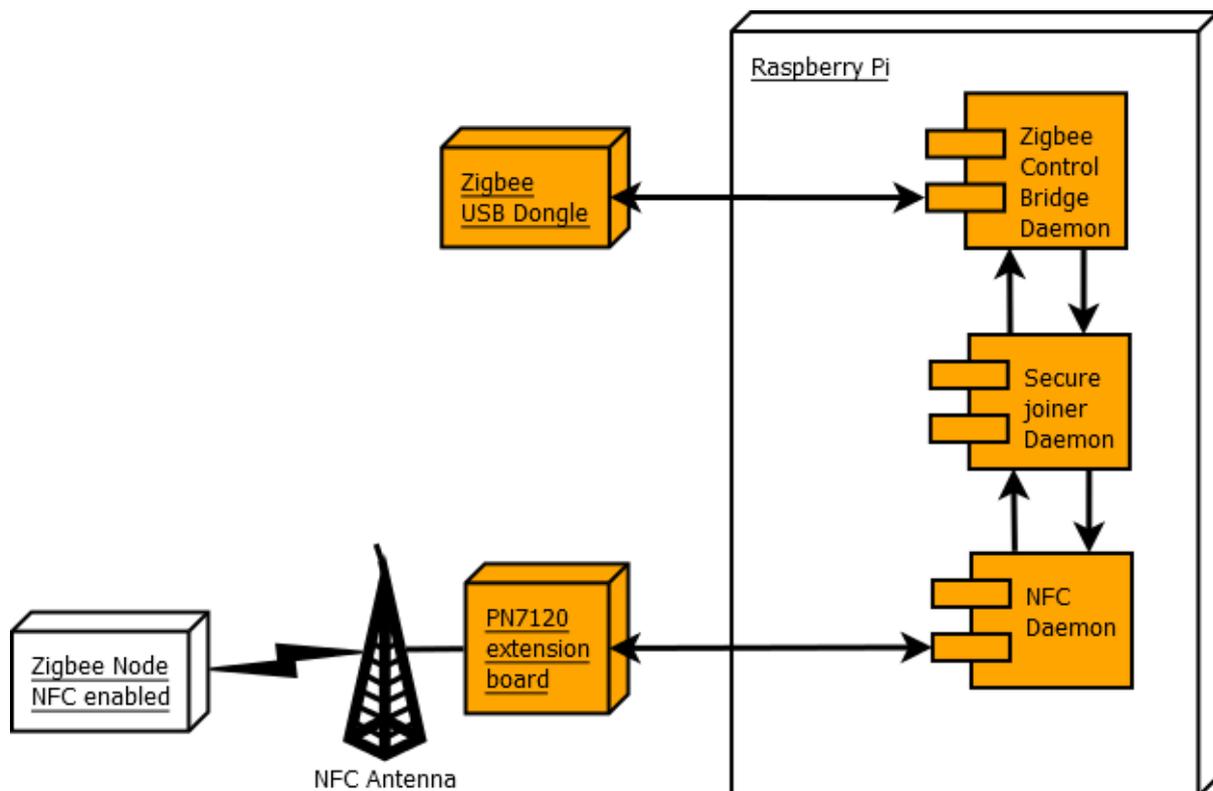
- an NFC-enabled mobile phone running a commissioning application
- the Gateway's NFC reader



Note: The JN516x-EK004 and JN517x-DK005 kits do not come with a commissioning application for smart phones, although the software implementation accommodates NFC-enabled phones.

3.1 Commission Node into Network via NFC Reader

When using the Gateway's NFC reader, commissioning/decommissioning mode can be set on the "mobile site" web page, prior to touching a device. Let us assume that the NFC reader daemon is in commissioning mode.



Once the IoT-compatible device touches the Gateway's NFC reader for single-touch commissioning (and remains touched for a short while), the flow through the system is as follows (note that ZCB refers to ZigBee Control Bridge):

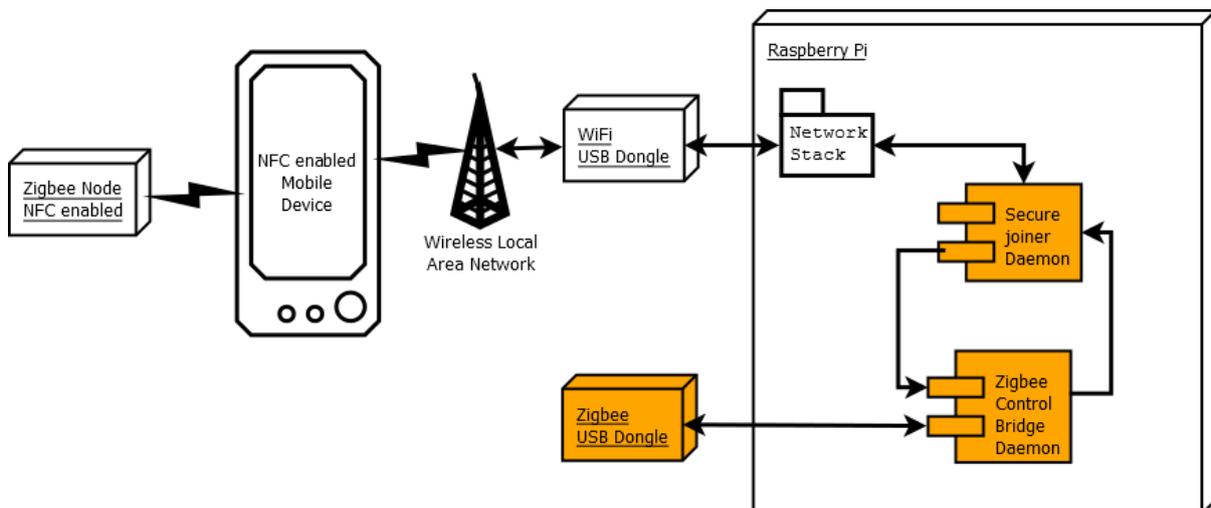
1. The NFC daemon registers at the secure join socket (port 2000).
2. The NFC daemon sends an *'linkinfo'* structure that it gets from the device that is currently being single-touch commissioned.
3. The Secure Joiner reads this socket message, translates it into an *'AuthorizationRequest'* CMD message and sends it to the ZigBee Coordinator via the input queue of the ZCB-Linux.
4. The ZigBee Coordinator processes the data and responds with a ZCB *'authorize'* response that is picked up by the ZCB-Linux and transferred to the Secure Joiner via its message queue.
5. The Secure Joiner translates this ZCB *'authorize'* message into a *'secjoin'* message and sends this to the NFC reader over the socket.
6. The NFC daemon writes the result back into the device which can then be commissioned to the wireless network, as soon as it is powered-on.

In case the system cannot handle the *'linkinfo'* structure and the response is never received, a timeout occurs and a corresponding error message is sent to the NFC reader. During the (de-)commissioning action, the NFC reader gives audible and LED feedback to the user.

Note: The Installation Code NFC commissioning has also been implemented to take benefit from the ZigBee 3.0 out of band commissioning. If the ZigBee end node supports the installation code, the gateway replaces the *linkinfo* structure with *oob_request_info* structure to match the NTAG data type.

3.2 Commission Nodes into Network with a Phone

An NFC-enabled phone can be put in two modes: commissioning and de-commissioning. The mode must be set prior to touching a node.



For commissioning, once the phone touches an IoT-compatible device for single-touch commissioning (and remains touched for a short while), the flow through the system is as follows (note that ZCB refers to ZigBee Control Bridge):

1. The phone registers at the secure join socket (port 2000).
2. Phone sends '*linkinfo*' structure that it gets from the device that is currently being single-touch commissioned.
3. Secure Joiner reads this socket message, translates it into an 'AuthorizationRequest' CMD message and sends it to the ZigBee Coordinator via the input queue of the ZCB-Linux.
4. The ZigBee Coordinator processes the data and responds with a ZCB 'authorize' response that is picked up by the ZCB-Linux and transferred to the Secure Joiner via its message queue.
5. The Secure Joiner translates this ZCB 'authorize' message into a 'secjoin' message and sends this to the phone over the socket.
6. The phone writes the result back into the device which can then be commissioned to the wireless network.

In case the system cannot handle the '*linkinfo*' structure and the response is never received, a timeout occurs and a corresponding error message is sent to the phone. During the (de-)commissioning action, the phone gives audible feedback to the user.

3.3 Secure Commissioning

To prevent the exposure of critical ZigBee information in the NTAG-I2C 'NFC connected tag', a public key exchange can be applied (ECDH scheme) to derive an AES session key and encrypt the ZigBee information stored in the NTAG-I2C. This and the commissioning of unpowered devices is explained in detail in [2].



Note: Currently, the use of (ECDH) secure commissioning is disabled in the Gateway. For more information, please refer to the *NFC Commissioning User Guide (JN-UG-3112)*.

3.4 De-commissioning with Phone or NFC Reader

The data flow for de-commissioning is much easier. In both cases (phone touches device or device touches NFC reader), the phone/reader just writes the de-commission command into the NTAG-I2C tag, after which the node can process it and leave the network. De-commissioning also affects the database as it is directly updated in the web interface, even when the device is unpowered.

4 Web Server CGI Scripts

4.1 Index.html

The IoT Gateway pages use a frameset with a quick-navigation bar at the top.

4.2 Control Page

The CGI script ('iot_control.cgi') that serves this page collects all the controls of the Gateway.

Zigbee Network

Normal Reset Create Network Permit join: Start

Channel mask: 0x4000

11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>										

(Green = preferable HA channels for co-existence with WIFI. Changes only take effect after clicking Create Network)

NFC Mode

Commission Device Decommission Device

Version

Refresh

Topology

Upload

Time

Sync with browser Fri Dec 2 16:46:41 2016

Restart

All Zcb Secure Joiner NTAG Daemon

Plug Emulator

Start Stop

System table

id	name	intval	stval	lastupdate
0	sysstart	1480679634		Fri Dec 2 12:53:54 2016
1	nfcmode	0		Fri Dec 2 13:20:59 2016
2	zcb_version	66061758		Fri Dec 2 12:54:09 2016
3	zcb_extpanid	0	00158D0000A11F53	Fri Dec 2 13:20:10 2016
4	chanmask	16384		Fri Dec 2 13:19:58 2016
5	zcb_channel	14		Fri Dec 2 13:20:10 2016
6	zcb_permit	0		Fri Dec 2 13:18:11 2016

Queue Overview

name	num
control interface	0
secure joiner	0
zcb-in	0
dbp	0

The fields of the above page are as follows:

- **ZigBee Network:** Used for resetting the ZigBee Control Bridge (ZCB), setting the ZigBee channel mask, starting the network and opening the permit-join window for EZ-mode commissioning.

- **NFC Mode:** Used to select commissioning or decommissioning of devices via NFC.
- **Version/Refresh:** Displays the version string of the ZCB so that the user can check if the latest version is installed.
- **Time/Sync with browser:** Sets the current time of the Gateway clock.
- **Restart:** (Re-)starts all IoT daemons or one at a time.
- **Plug Emulator:** Enables/disables a periodic plug meter generator (for testing purposes).
- **System Table:** Shows the current contents of the system table.
- **Queue Overview:** For all IoT message queues, shows how many messages are in a queue. Should be zero (0) when all processes run fine.

The Control Page CGI script executes some of the commands indirectly via the Control Interface program, as CGI-scripts cannot modify the IoT database.

4.3 Database Viewer

The 'iot_database.cgi' CGI script displays the contents of all IoT tables in the database.

IoT Database - Thu Apr 14 8:00:34 2016 (11122)

Device Tables overview

Clear Lamps Clear Plugs Clear Plughistory Clear

iot_devs

id	mac	dev	ty	par	nm	heat	cool	tmp	hum	prs	co2	bat	batt	als	xdoc	yloc	zloc	sid	cmd	lvl	rgb	kelvin	act	sum	flags	lastupdate
0	00158D0000F4D4BF	switch	dim	0		0	0	0	0	0	0	0	0	0	0	0	0	0	Off	0	000000	0	0	0	9	Thu Apr 14 7:56:30 2016
1	00158D0000A11D5B	lamp	col	0		0	0	0	0	0	0	0	0	0	0	0	0	0	on	99	000000	0	0	0	9	Thu Apr 14 7:59:58 2016

iot_plughistory

id	mac	sum	lastupdate

System Table

Clear

iot_system

id	name	intval	strval	lastupdate
0	sysstart	1460568780		Wed Apr 13 19:33:00 2016
1	zcb_version	310378497		Thu Apr 14 7:31:12 2016
2	chanmask	2048		Wed Apr 13 19:33:45 2016
3	zcb_channel	11		Thu Apr 14 7:56:12 2016
4	zcb_extpanid	0	00158D0000A11F53	Thu Apr 14 7:56:12 2016
5	permit	1460613564		Thu Apr 14 7:56:24 2016
6	zcb_permit	1		Thu Apr 14 7:56:24 2016

zcb

Clear ZCB

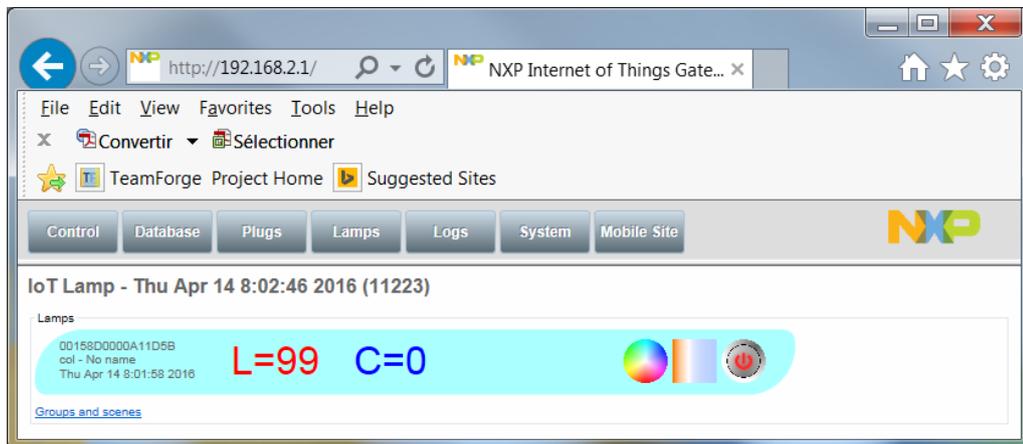
iot_zcb

id	mac	status	saddr	type	lastupdate
0	00158D0000F4D4BF	2	8976	104	Thu Apr 14 7:56:30 2016
1	00158D0000A11F53	2	0	2	Thu Apr 14 7:56:12 2016
2	00158D0000A11D5B	2	cc65	102	Thu Apr 14 7:56:55 2016

When needed, the web page contents are automatically (partially) refreshed by AJAX technology. For test purposes, the entire database can be cleared, all tables at once or in subsets.

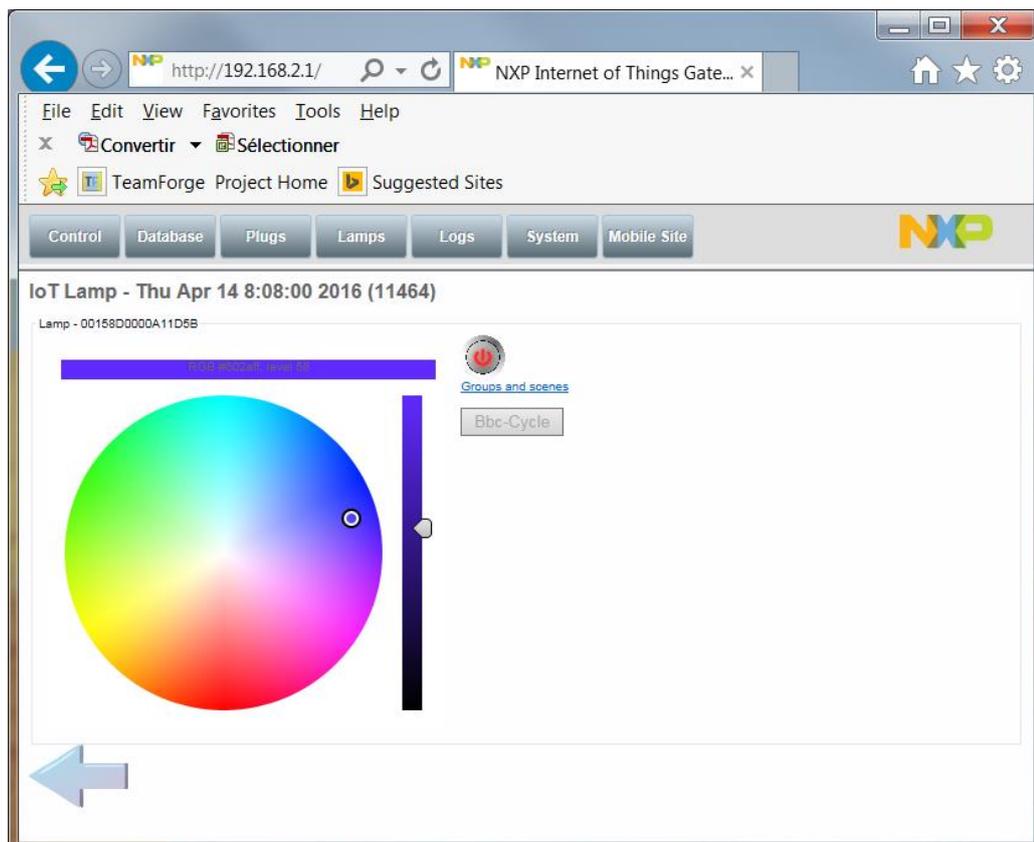
4.4 Lamps Page

The 'iot_lamp.cgi' CGI script displays all the lamps in the system:



Lamps can be controlled directly from this view by pressing the on/off button on the right.

To the left of the on/off button, there is a colour wheel image and a colour gradient image. Selecting either of these opens the lamp control view. The screenshot below shows the view after the colour wheel has been selected.

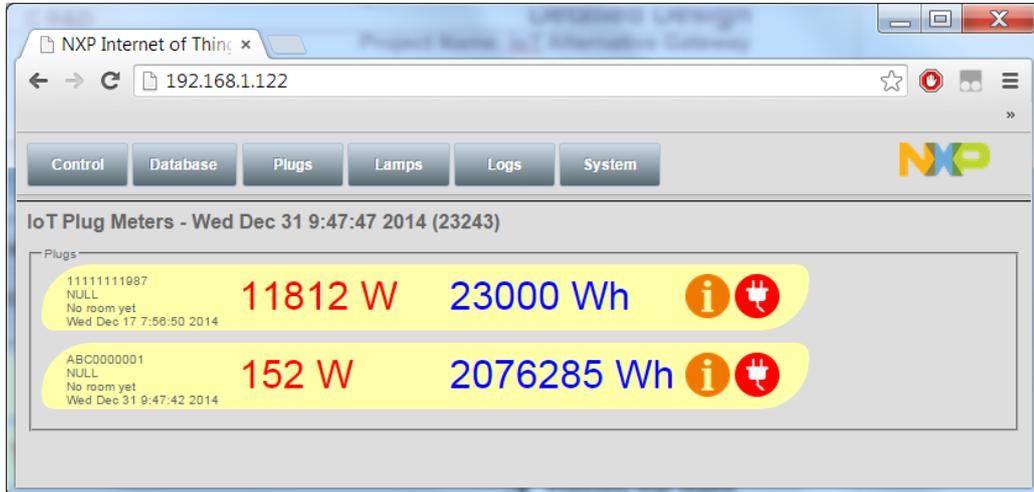


A new colour can be picked by pressing and dragging the mouse pointer on the colour wheel. Simple on/off lamps, dimmable lamps and tuneable-white lamps are also supported.

4.5 Plugs Page

Plugs are not supported in the JN516x-EK004 and JN517x-DK005 kits. However, the Gateway supports these devices and Reference Designs for them can be obtained from NXP.

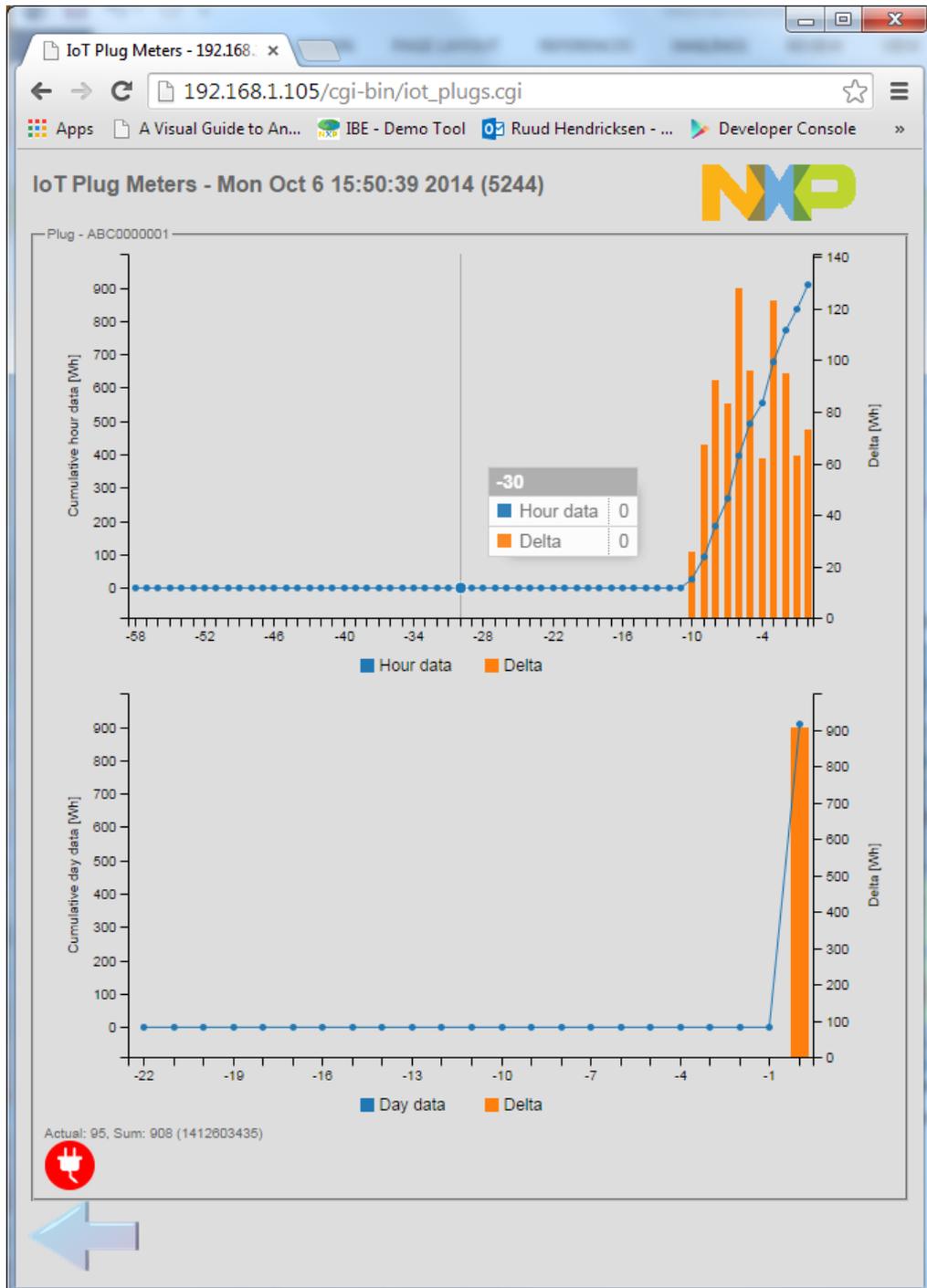
The 'iot_plugs.cgi' CGI script displays all plug meters in the system.



Plugs can be controlled directly from this view by pressing the right on/off button (showing the outline of a plug). To the left of the on/off button, there is an information button (showing an 'i') - clicking this opens the Plug History view, which is illustrated in the screenshot below.

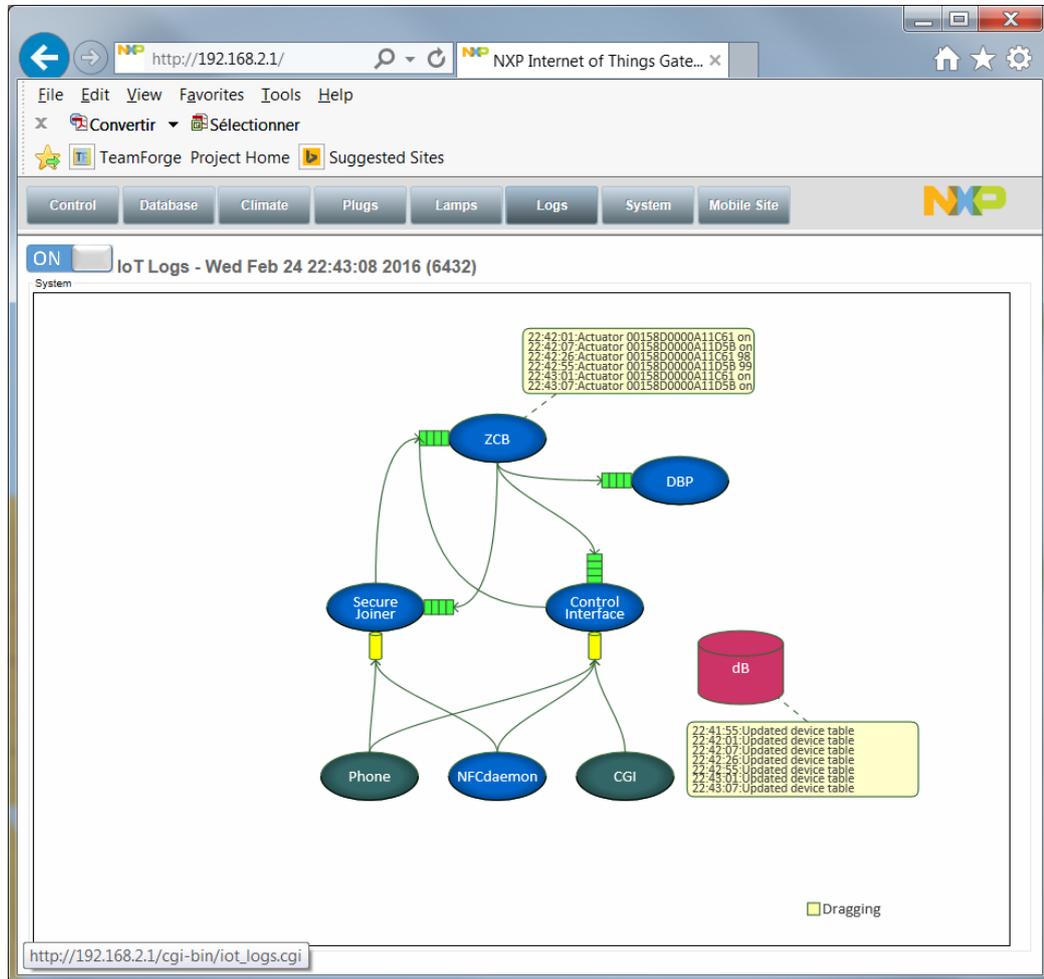
The top half of the view shows the energy consumption over the last 60 minutes (60 readings) and the bottom half shows the consumption over the last 24 hours (24 readings). Note that these are sliding windows from right to left.

Hovering the mouse over the graph gives detailed consumption for that point in time.



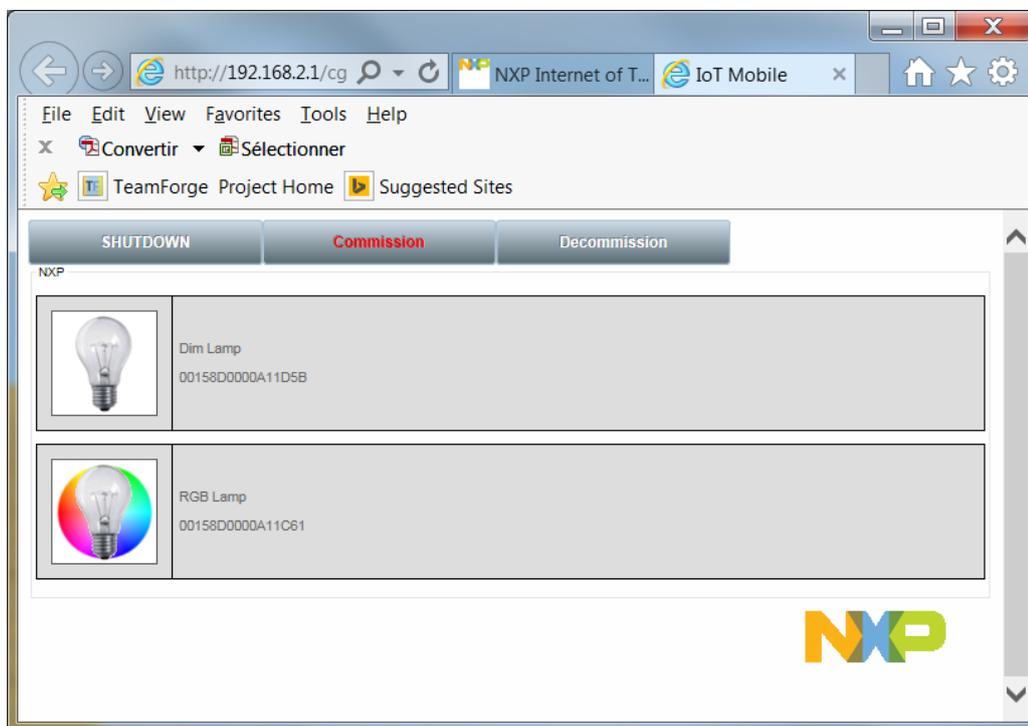
4.6 Logging Page

The 'iot_logs.cgi' CGI script shows a graphical status logging all IoT processes: Balloon textboxes show new activity in the processes. Pressing the on/off button toggles between this graphical view and a text-oriented view.



4.7 Mobile Site

Due to the inability to use NFC (for commissioning purposes) on some phones, a “Mobile Site” web page is available for use with such smart phones, to be used in combination with the Gateway’s NFC reader for commissioning.



The header layout of this simplified page contains three buttons, as follows:

Button	Function
Shutdown	Shut down the Gateway
Commission	Put the NFC reader daemon in the commissioning state
Decommission	Put the NFC reader daemon in the de-commissioning state

Below the header, there is a single row per device, which can be clicked to open the device’s control window. Note, in this view, only lamps and plug meter devices are shown. The other devices can still be accessed via the normal IoT web pages.

Revision History

Version	Date	Description
1.0	19-Nov-2015	First release
1.1	15-Apr-2016	Add support for Raspberry Pi 2, fix typo errors, and update screen shots.
1.2	20-Jul-2016	Add support for JN517x-DK005 (JN5179 Control Bridge)
1.3	3-Oct-2016	Editorial changes made
1.4	2-Dec-2016	Mark Raspberry Pi Model B+ as obsolete (unsupported) Add reference to JN-AN-1216 (ZCB ZigBee 3.0) Remove 3 rd NFC mode (Factory Reset)
1.5	24-Mar-2017	Add note about NFC commissioning with Installation Code

Important Notice

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

NXP Semiconductors

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

www.nxp.com