ZigBee Gateway User Guide and Setup

Introduction

The Purpose of this document is to illustrate how to start-up and begin using the ZigBee Wireless Gateway. The document also serves as basic guide to controlling and managing ZigBee sensors using the web interface.

It is important to note that in the context of this document, the term ZigBee refers to specifically XBee Series 2.5 ZigBee modules from Digi. Throughout the document, ZigBee, XBee, Sensor and Node are used interchangeably to mean an XBee Module running the ZNet API firmware (with API mode 2) which have sensors/actuators attached to any of their internal ADC/Digital IO, and are capable of joining a ZigBee Wireless Mesh Network on the created PAN.

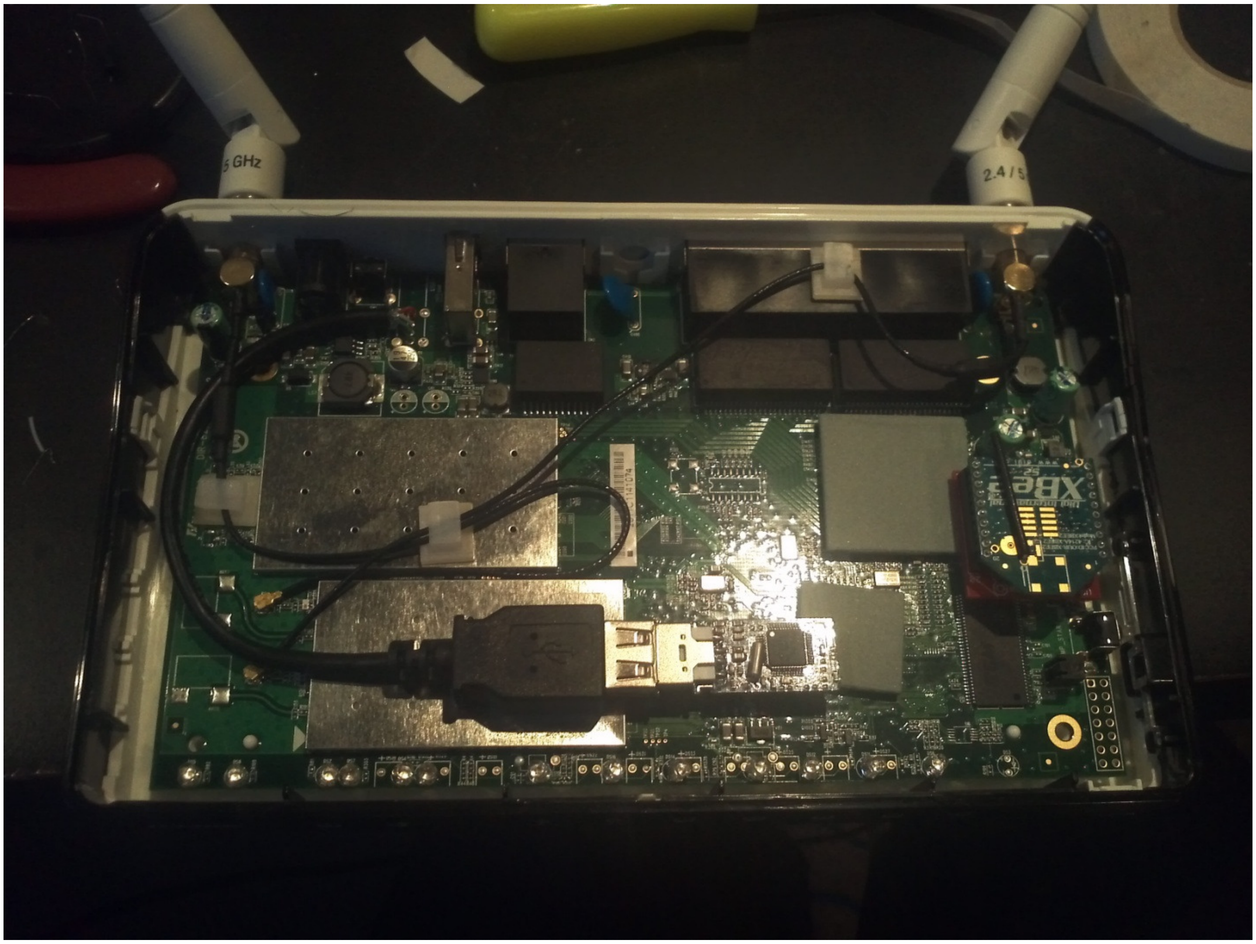
The Gateway

The ZigBee Gateway is an application that runs on a Wireless Router (Dlink DIR-825 Dual Band Wireless N Router) with a custom built OpenWrt image along with customized hardware. OpenWrt is an open source Linux Platform for wireless routers. More information about OpenWrt can be found at openwrt.org. Since The ZigBee Gateway is an application that runs in the background, the device operates as a regular OpenWrt based wireless router simultaneously.

The custom built image is built on the OpenWrt Attitude Adjustment Release. The image has features baked into it such as the LUCI user interface and some kernel mods to allow storing and running applications off a USB stick, which is necessary for the system to work.

The Hardware

The Hardware has been customized to include an internal 1GB USB flash drive that uses one of the un-utilized USB channels if the DIR-825. This frees the External USB port for other applications in the future. The router also contains an unutilized TTL serial port. The Router has been modified to use the Serial port to communicate with an *XBee Series 2.5* 2.4GHz Radio Module from Digi, which has been mounted and secured internally. It is important to note that both the USB Flash Drive and XBee Module are removable if needed.



The internal XBee module is running the XBee ZNet Coordinator API Firmware (API mode 2) which allows it to create and control a ZigBee/XBee Sensor Network. All XBee Nodes on the PAN must be running the same firmware to work properly. More information about the different firmware settings can be found at digi.com. Also Refer to Appendix A.

The USB Flash Drive is formatted as an EXT4 disk and has been partitioned to use 700 MB for Data and Applications and uses the remainder of the disk as SWAP space. The ZigBee Gateway Application runs in Python 2.7.3. Since Python itself is relatively large compared to the Routers small Flash memory, it must be installed on the USB stick. If the USB stick is removed or isn’t mounted properly, the Application will not work.

The Python Application makes use of the CherryPy Web Framework to enable all the Web Server functionality. The XBee/ ZigBee functionality is controlled by a multi-threaded shared library, written in STD C++, built from the ground up and has been extended to be useable from a python application. The Underlying XBee/ZigBee Network operations are abstracted away to run independent of the python application. The Library however allows the Python application to configure, start/stop and query the XBee Network.

Getting Started

The application does not run automatically when the router Boots up, it must be started manually by the user. This could be changed in the future with a simple startup script for OpenWrt.

You will need couple things to get started

- A Terminal Program on your computer that can connect to a networked device via SSH. Installing PuTTy is recommended if you are using Windows. Terminal works fine if you are using Mac OS.
- An Ethernet Cable to connect to your router initially. After you set up the Wireless Router on your network, you can connect to it wirelessly if you wish.
- A Second Ethernet Cable with a connection to the internet (could connect to a network switch that is connected to the internet for example)

Connecting to the Router

First you will want to connect an Ethernet cable from your PC to one of the 4 LAN connections of the Router to configure it. Ensure your network settings allow for DCHP client connections. If you wish to have the router connect to the internet, connect a second Ethernet Cable from the WAN port to any internet modem/switch.

Power on the Router. The Status LED will begin flashing while booting up. This can take up to a 60 seconds so be patient. Do not disconnect you Power Supply during boot up. When the status LED stops blinking, and turns blue, the router is finished booting up. At this point you can begin your SSH session with the router

The IP address of the Gateway is set to default ***192.168.1.1***. Launch your terminal program of choice and SSH to ***192.168.1.1***

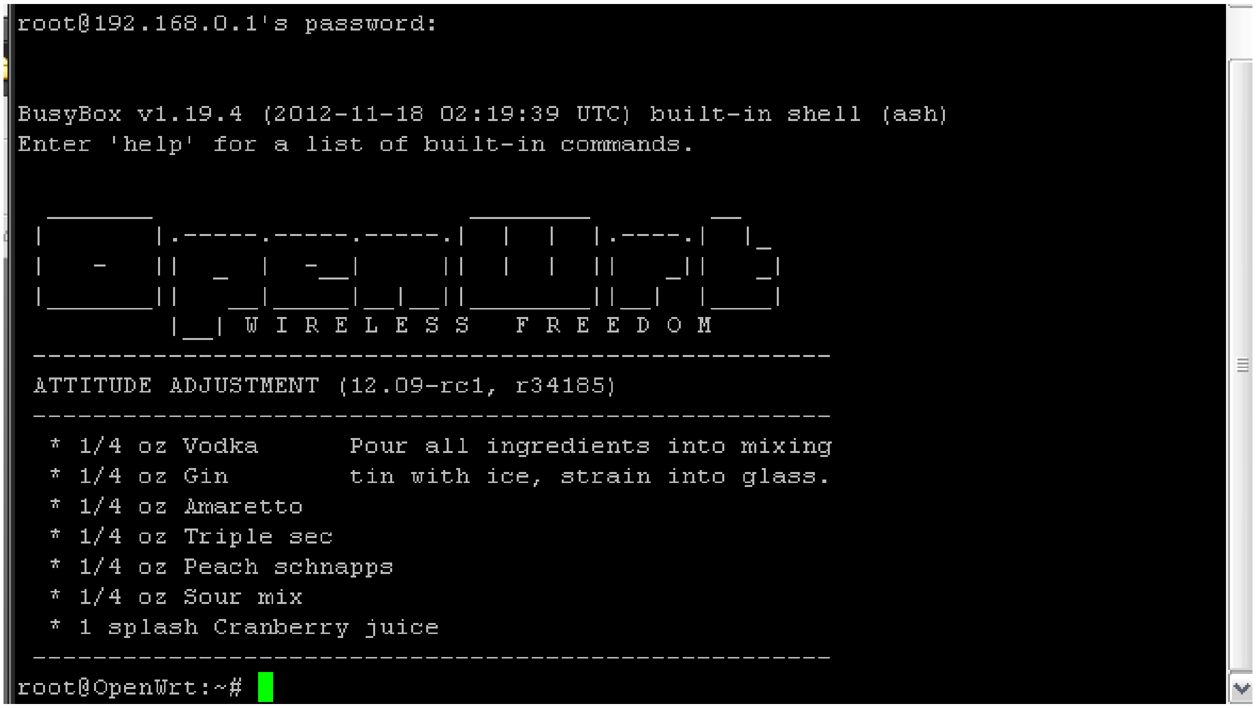
You will first be prompted for a login

Type in ***root*** and press enter

Next you will be prompted for a password

The default Password to the router is ***magic***

You should now see the Open WRT Splash Screen and shell prompt. You are now ready to run Application from this point, but you may want to setup/check your network connection first if you wish to have the application communicate with Sense Tecnic / WotKit.



```
root@192.168.0.1's password:


BusyBox v1.19.4 (2012-11-18 02:19:39 UTC) built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 ATTITUDE ADJUSTMENT (12.09-rc1, r34185)
 -----------------------------------------------------
  * 1/4 oz Vodka      Pour all ingredients into mixing
  * 1/4 oz Gin        tin with ice, strain into glass.
  * 1/4 oz Amaretto
  * 1/4 oz Triple sec
  * 1/4 oz Peach schnapps
  * 1/4 oz Sour mix
  * 1 splash Cranberry juice
 -----------------------------------------------------
root@OpenWrt:~#
```

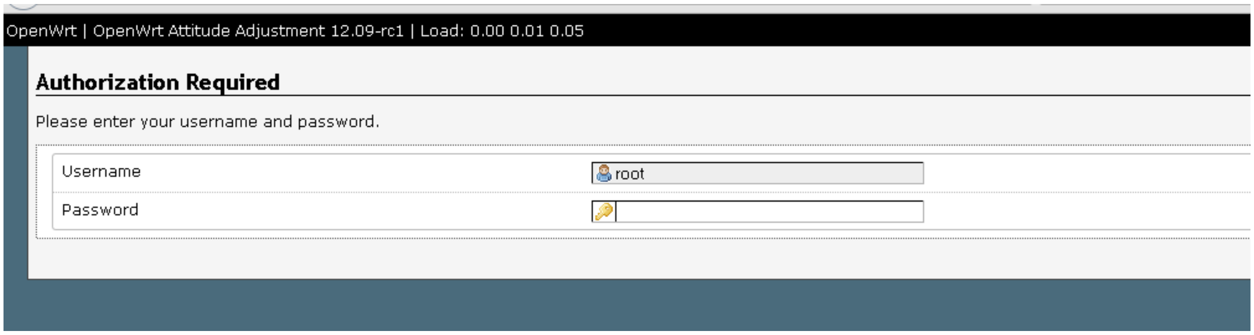
Note: password can be changed using the LUCI UI password utility. You can also type passwd to run the password utility in command shell once logged into the router.

Setting-Up the Network

The Application needs to be able to connect to the internet to post sensor data to the WotKit Web Application. If you have the router wired to a net connection on the WAN port, likely no further settings are needed. You may want to configure the router to use a wireless network instead.

To set up a Wireless network you can use the installed LUCI UI to configure your connection.

Open up a browser window and type the IP address of the router in the address bar. You should be redirected to a login screen for the LUCI UI. The user name is ***root***. The default password is ***magic***.



OpenWrt | OpenWrt Attitude Adjustment 12.09-rc1 | Load: 0.00 0.01 0.05

Authorization Required

Please enter your username and password.

Username	root
Password	

Note: if no LUCI comes up or a connection times out, you may need to enable it manually or restart your router. To enable it manually, go to the active SSH command shell window and type

```
/etc/init.d/uhttpd enable
/etc/init.d/uhttpd start
```

LUCI should now be enabled.
Visit openwrt.org for more information on troubleshooting

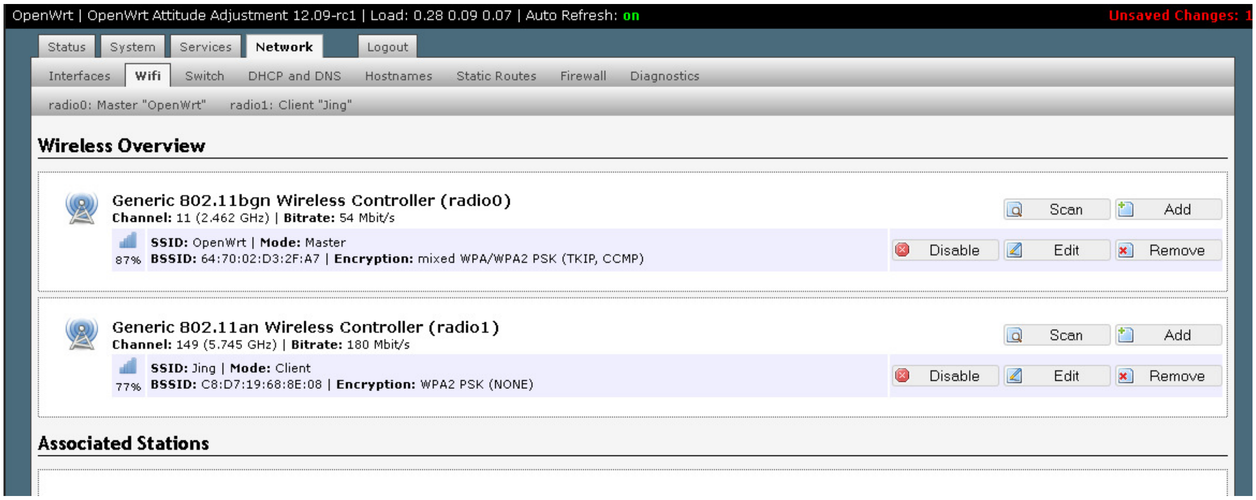
Navigate to the Network Tab on the Top, and Click on Wifi.

The DIR 825 has 2 Wireless channels, radio 0 and radio 1. Radio 0 is for standard 2.4GHz b/g/n channels, and radio 1 is for 5 GHz channels. Most cases you will use radio 0 unless a 5GHz signal is available.

If radio 0 is disabled, enable it by clicking enable.

Click scan to perform a network scan to find a wireless access point.

Find the network you want to connect to and click join network. You can now enter in the passphrase for the network as well as configure additional advanced settings. You must click Save and Apply to finish and connect to the network.



OpenWrt | OpenWrt Attitude Adjustment 12.09-rc1 | Load: 0.28 0.09 0.07 | Auto Refresh: on

Unsaved Changes: 1

Wireless Overview

Generic 802.11bgn Wireless Controller (radio0)
Channel: 11 (2.462 GHz) | **Bitrate:** 54 Mbit/s

87% **SSID:** OpenWrt | **Mode:** Master
BSSID: 64:70:02:D3:2F:A7 | **Encryption:** mixed WPA/WPA2 PSK (TKIP, CCMP)

Generic 802.11an Wireless Controller (radio1)
Channel: 149 (5.745 GHz) | **Bitrate:** 180 Mbit/s

77% **SSID:** Jing | **Mode:** Client
BSSID: C8:D7:19:68:8E:08 | **Encryption:** WPA2 PSK (NONE)

Associated Stations

If everything went fine, you should now be connecting to the network. Test the status of your Internet Connection by going to the Diagnostics Tab under Network.

You should be able to ping openwrt.org. If you are unable to ping, try setting up your network again. Note that networks which require a Certificate such as UBC Secure will not work.

Starting the Application

The Application is started manually via the SSH session. In your terminal program, type

```
python “/mnt/XBeeApp.py”
```

/mnt is where the XBeeApp.py resides. (memory stick is mounted at /mnt)

The application should now be launched. You will see debug messages printed out into the terminal window. These messages are just for debugging the system and may be disabled in the future.

The Application will have started a new server that will be listening on Port 8080. Access it by typing ***192.168.1.1:8080*** in your browser address bar

Troubleshooting Starting the Application

If you get error messages saying that the python cannot be found or the file “XBeeApp.py” cannot be found, try following these directions.

Change into the /mnt directory (type `cd /mnt`). Do a quick list of the directory (type `ls`) to ensure the disk is actually mounted. Sometimes OpenWrt fails to mount the disk at startup. If you don’t see XBeeApp.py or anything listed, you may have to manually mount the disk.

To mount the disk type in the command:

```
mount –t ext4 /dev/sda1 /mnt –o rw,sync
```

This mounts the first partition of the internal memory stick at the mount point /mnt

Type `ls` again. You should now be able see the file XBeeApp.py

Type in the terminal window

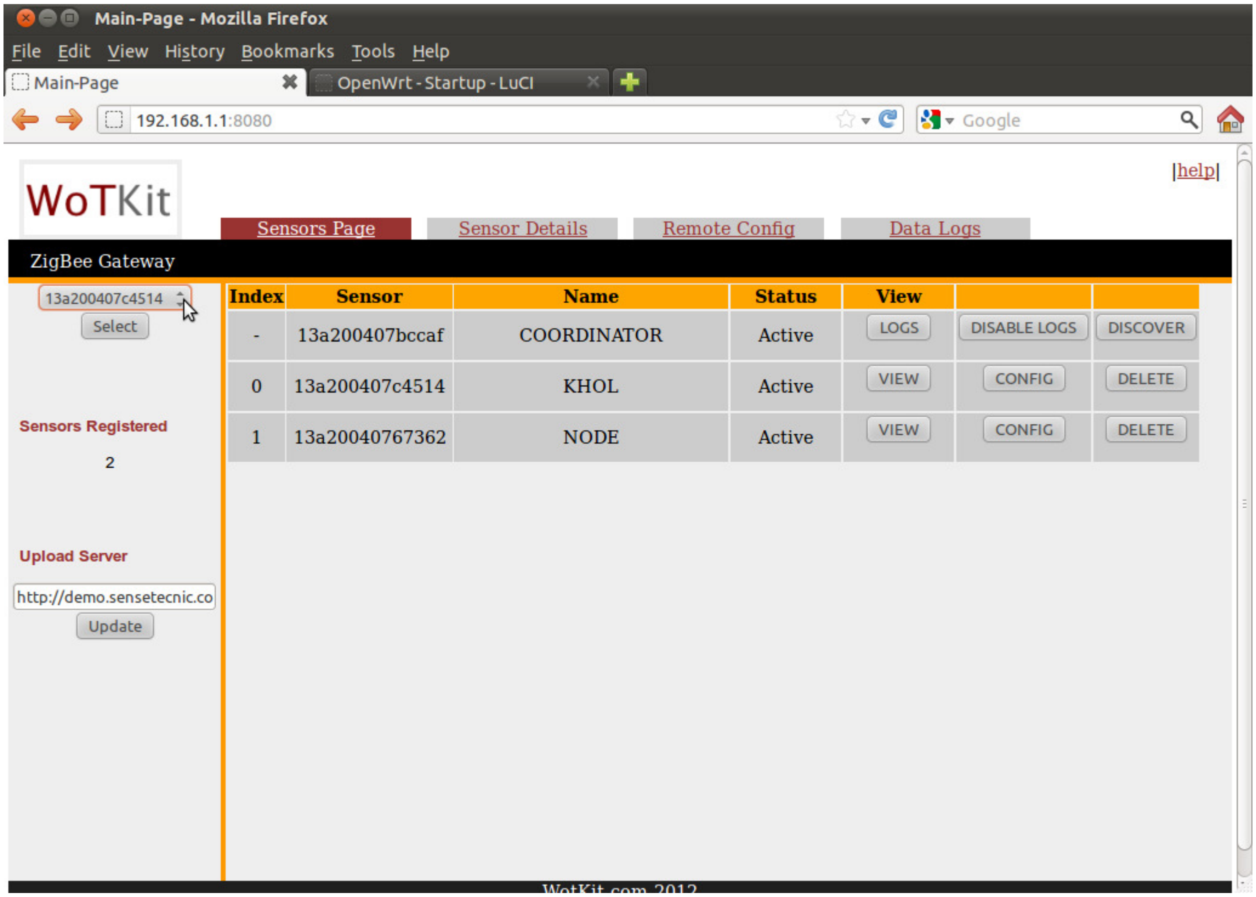
```
python “XBeeApp.py”
```

Now you should see the application start up and print debug messages in the window.

The XBee Application

The web Application is always served on port 8080 of the router. To View the Web Application, Open up a browser Window and type ***192.168.1.1:8080*** into the address Bar. You should be redirected to the ZigBee Server/ Gateway main Page.

The Main Page / Sensors Page



13a200407c4514

Sensors Registered

2

Upload Server

http://demo.sensetecnic.co

Index	Sensor	Name	Status	View		
-	13a200407bccaf	COORDINATOR	Active	LOGS	DISABLE LOGS	DISCOVER
0	13a200407c4514	KHOL	Active	VIEW	CONFIG	DELETE
1	13a20040767362	NODE	Active	VIEW	CONFIG	DELETE

The Coordinator starts a PAN on PAN ID 0x1234. Any XBee Node that connects to this PAN will automatically be registered on the Gateway and displayed on this page. The Newly Discovered Node will also be registered on WotKit simultaneously. If the gateway cannot connect to the WotKit server, the newly discovered nodes will be queued for registry until a connection can be established. Sensor data is logged by default and takes place immediately, regardless of server connection to ensure data will not be lost. These logs can be retrieved later through your browser.

The Coordinator

The Coordinator and its Serial Number is always displayed in the first row of the list. The Coordinator possesses a Button to Disable/Enable Logs and to Do Node Discovery

Node Discovery

Nodes are discovered dynamically during run time however the user may want to trigger a node discovery for troubleshooting purposes. Do Node Discovery attempts to uncover as many connected nodes as possible by issuing a broadcast beacon, and waiting any for devices to respond. Each Device responds to the Beacon upon receiving it, notifying the Coordinator to add it to the system.

Connected XBee/ZigBee Sensors

Displays the unique 64 bit Serial Number of the XBee Node as it is written on the device.

Displays the Node's Name or Node ID (if it has been set). Useful for Identifying types of sensors (i.e.: AMBIENT TEMP SENSOR)

There are a few different Options on the right portion of the Sensor List, View, Config, and Delete

View – Lets you view Specific Information regarding that Sensor

Config – Lets you configure the specific Sensor

Delete – Removes that specified sensor from the PAN.
The Node is automatically deleted from WotKit at the same time. Note that this is only useful for Nodes that are no longer active. Active Sensors are still be able to contact the Coordinator, and are re-added to the system. This prevents the user from accidently removing the wrong sensor.

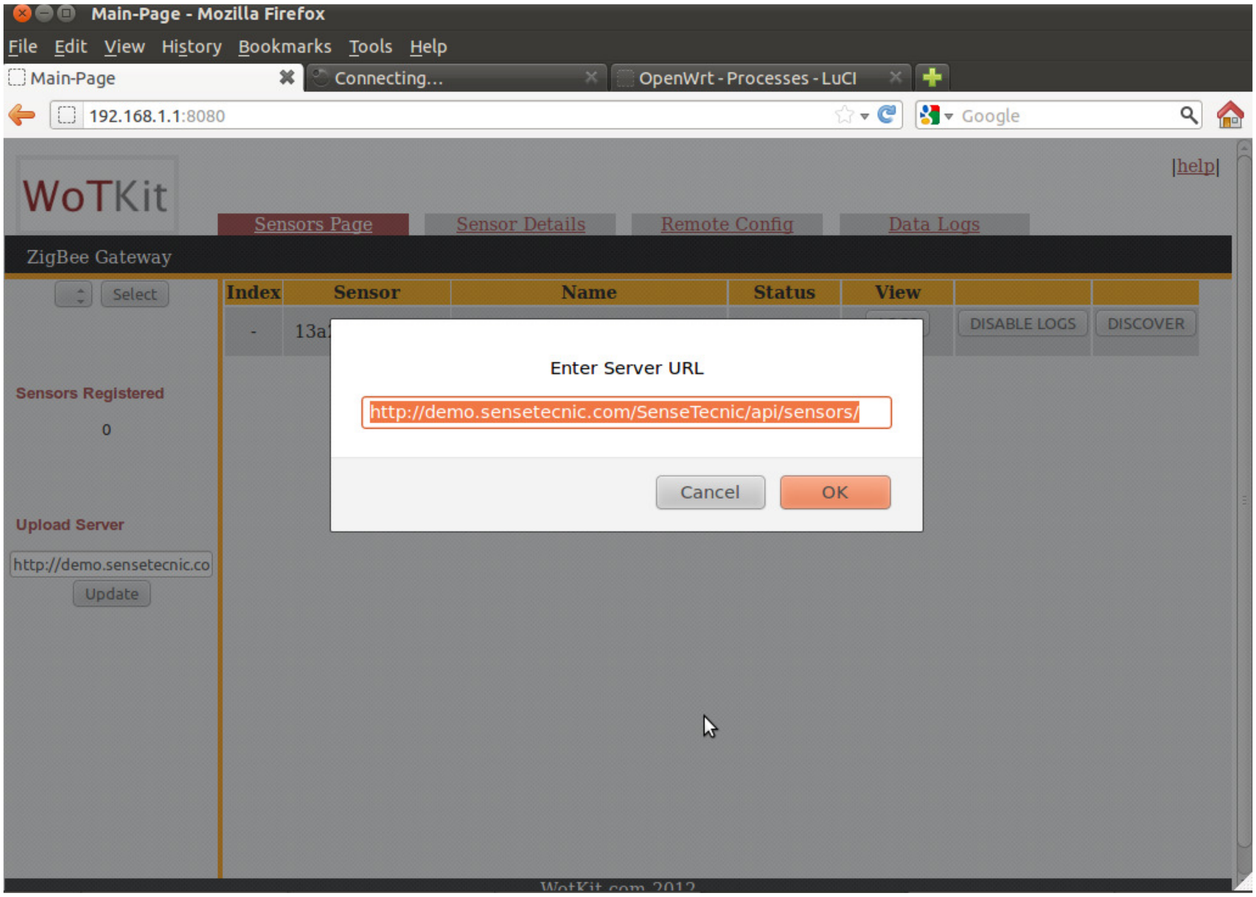
You can alternatively select a sensor for Viewing by using the dropdown menu in the left column. The Sensors will be listed by their Serial Number in the order they were discovered.

Sensor Activity

If the Node has sent data within the last 60 seconds, it is considered active, otherwise considered inactive. Sensors registered in the left Column displays the number of sensors that joined the network.

Changing the Upload Server

You can Change or Update the upload Server by clicking update under Upload Server. Just type in the new server address in the window that pops up. Useful if the location or address of the WotKit server ever moves or changes URLs.



Sensors Registered

0

Upload Server

http://demo.sensetecnic.co

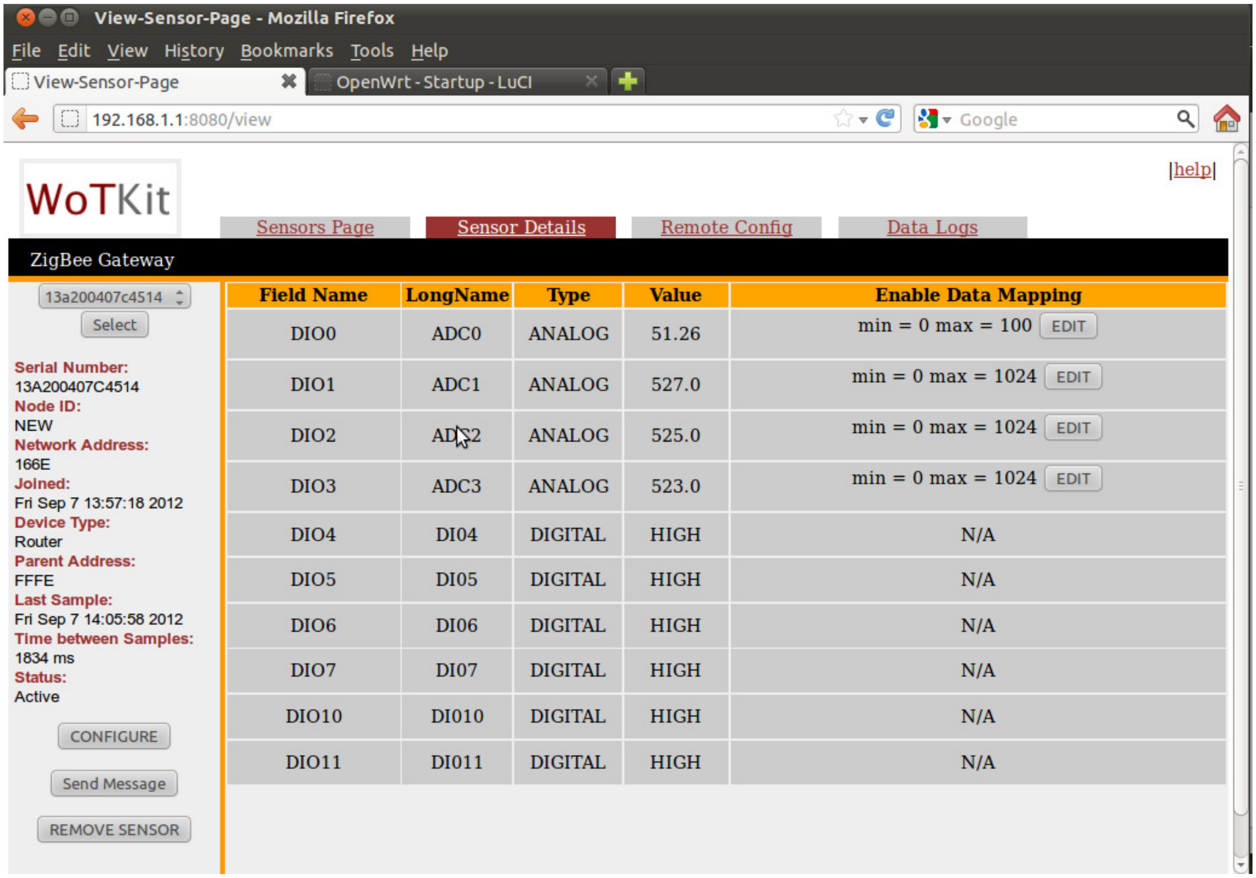
Enter Server URL

http://demo.sensetecnic.com/SenseTecnic/api/sensors/

*Note: The Default Server being used is the Sense Tecnic Demo Server and may be outdated. The Sensors data is pushed to the Site using the username **khol** and the password **action12***

Viewing Sensor Details Page

This Page lets you view important details about the sensor node as well as reports last Sensor readings/DIO values on each active channel. It also lets you change the sensor reading range from a raw ADC value to real world units/values. You can select the Sensor/Node to view from the dropdown list in the left column.



Serial Number: 13A200407C4514
Node ID: NEW
Network Address: 166E
Joined: Fri Sep 7 13:57:18 2012
Device Type: Router
Parent Address: FFFE
Last Sample: Fri Sep 7 14:05:58 2012
Time between Samples: 1834 ms
Status: Active

Field Name	LongName	Type	Value	Enable Data Mapping
DIO0	ADC0	ANALOG	51.26	min = 0 max = 100 EDIT
DIO1	ADC1	ANALOG	527.0	min = 0 max = 1024 EDIT
DIO2	ADC2	ANALOG	525.0	min = 0 max = 1024 EDIT
DIO3	ADC3	ANALOG	523.0	min = 0 max = 1024 EDIT
DIO4	DI04	DIGITAL	HIGH	N/A
DIO5	DI05	DIGITAL	HIGH	N/A
DIO6	DI06	DIGITAL	HIGH	N/A
DIO7	DI07	DIGITAL	HIGH	N/A
DIO10	DI010	DIGITAL	HIGH	N/A
DIO11	DI011	DIGITAL	HIGH	N/A

The Sensor Details

The following details about the sensor are displayed in the column on the left
Serial Number – 64 bit Serial Number of the Device.

Node ID –The Name of the sensor (if Set)

Network Address – The current 16-bit Network address of the Node that was issued by the Coordinator.

Joined- The time and date the sensor first joined the Network

Device Type – Displays whether the device is a Router or an End Device.

Note:Routers are the basis for Mesh Networks and can have other Nodes connect to them, and relay data across the network to the Coordinator/Gateway. Routers are not permitted to Sleep and therefore are best connected to mains. End Devices can only connect to Routers or to the Coordinator. They are Devices that are permitted to sleep and are useful for low power/ battery power applications.

Parent Address – If the specified Node connects to another Node and not directly to the Coordinator, this field displays the 16-bit Network Address of the node that is routing the data. If connected directly to the Coordinator, the default displays FFFE.

Last Sample – Displays the time and Date the last sample was recorded for the specific sensor

Time Between Samples – Displays the time in milliseconds that has elapsed since the last sample was recorded.

Status – Displays whether the device is Active (has sent data within the last 60 seconds) or inactive.

Sensor Readings

The Table in the Right Column displays the current channels readings for the DIO of the Node. Only the Active Channels (ones enabled) and their settings are displayed.
The Table is structured as follows

Field Name - The particular Channel of the Sensor (DIO0-DIO12). Also Corresponds to the particular channel of the Sensor in WotKit that data is being sent to.

Long Name- the Identifier of the channel being used.

Type – The Type of Data being recorded on that Channel. Either Analog or Digital. Channels DIO0-3 can be configured either Analog or Digital All remaining channels are Digital Only.

Value – Digital: HIGH if the Input / or Output Value is High, LOW if reading is Low. Analog: The value of the sensor

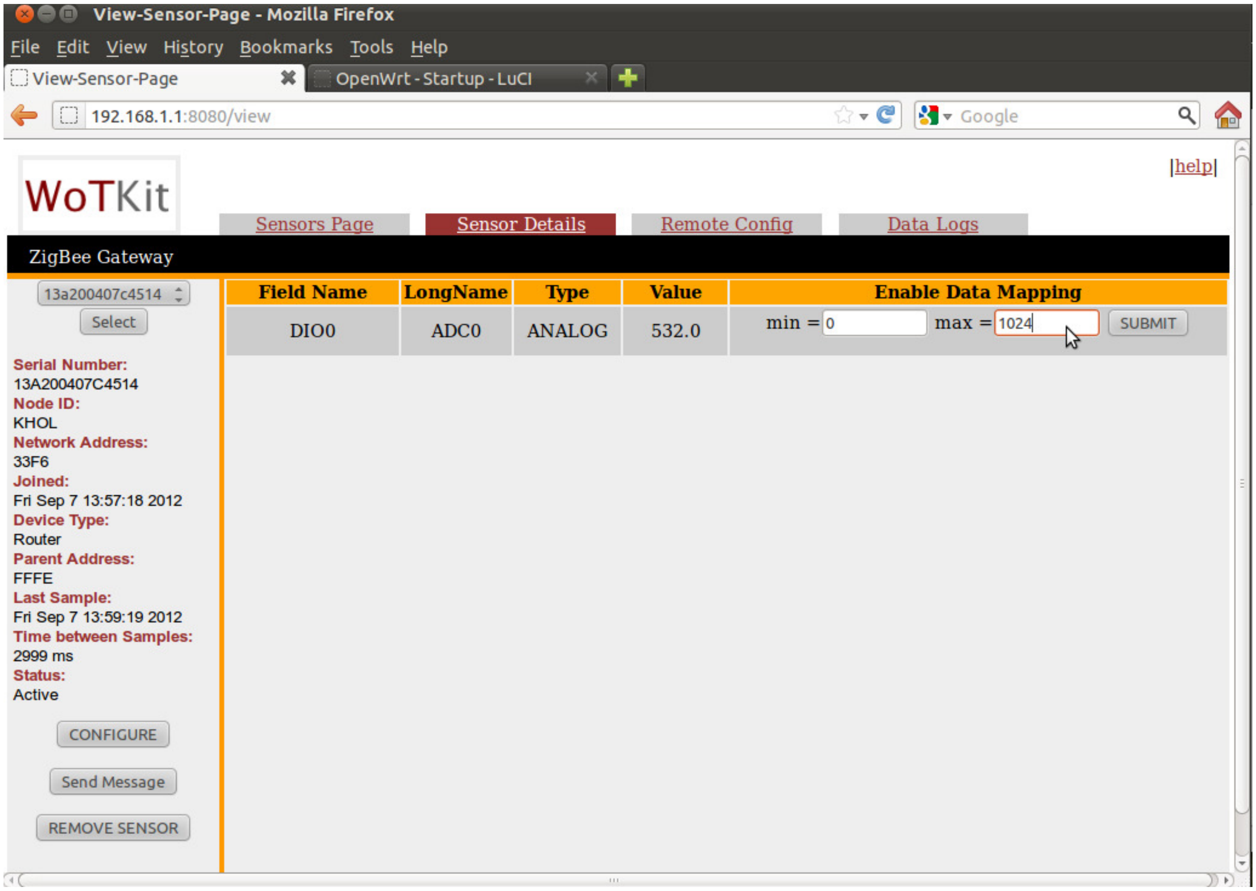
Data Mapping

By default, Value is set to the RAW value of the ADC; however, this can be mapped to something more meaningful/useful by Editing the Data Mapping in the next column. This feature adds the ability to real-world units to readings from a sensor that is attached to a channel of a particular Node.

The user must determine the upper and lower bounds themselves beforehand while taking the following into consideration

- The XBee ADC has is 10-bits. Therefore Raw readings range from 0 to 1024
- The XBee ADC ranges from 0 to 1.2 V max. XBees do not Support input voltage any higher than 1.2V and will result in damage.

- Each increase corresponds to approx 1.20 mV increases.
- The Sensor must operate in the linear region to be re-mapped accurately



Serial Number:
13A200407C4514
Node ID:
KHOL
Network Address:
33F6
Joined:
Fri Sep 7 13:57:18 2012
Device Type:
Router
Parent Address:
FFFE
Last Sample:
Fri Sep 7 13:59:19 2012
Time between Samples:
2999 ms
Status:
Active

Field Name	LongName	Type	Value	Enable Data Mapping
DIO0	ADC0	ANALOG	532.0	min = 0 max = 1024 SUBMIT

Min: Enter in the desired lower boundary of the reading. Negative and Positives are both Valid

Max: Enter in the desired upper boundary of the reading. Negative and Positive Values are valid.

Once Data Mapping is set to a new Value, Each subsequent value will be mapped to the new Value and recorded as the new value.

Some Special Features

Provisions for Smart Sensors

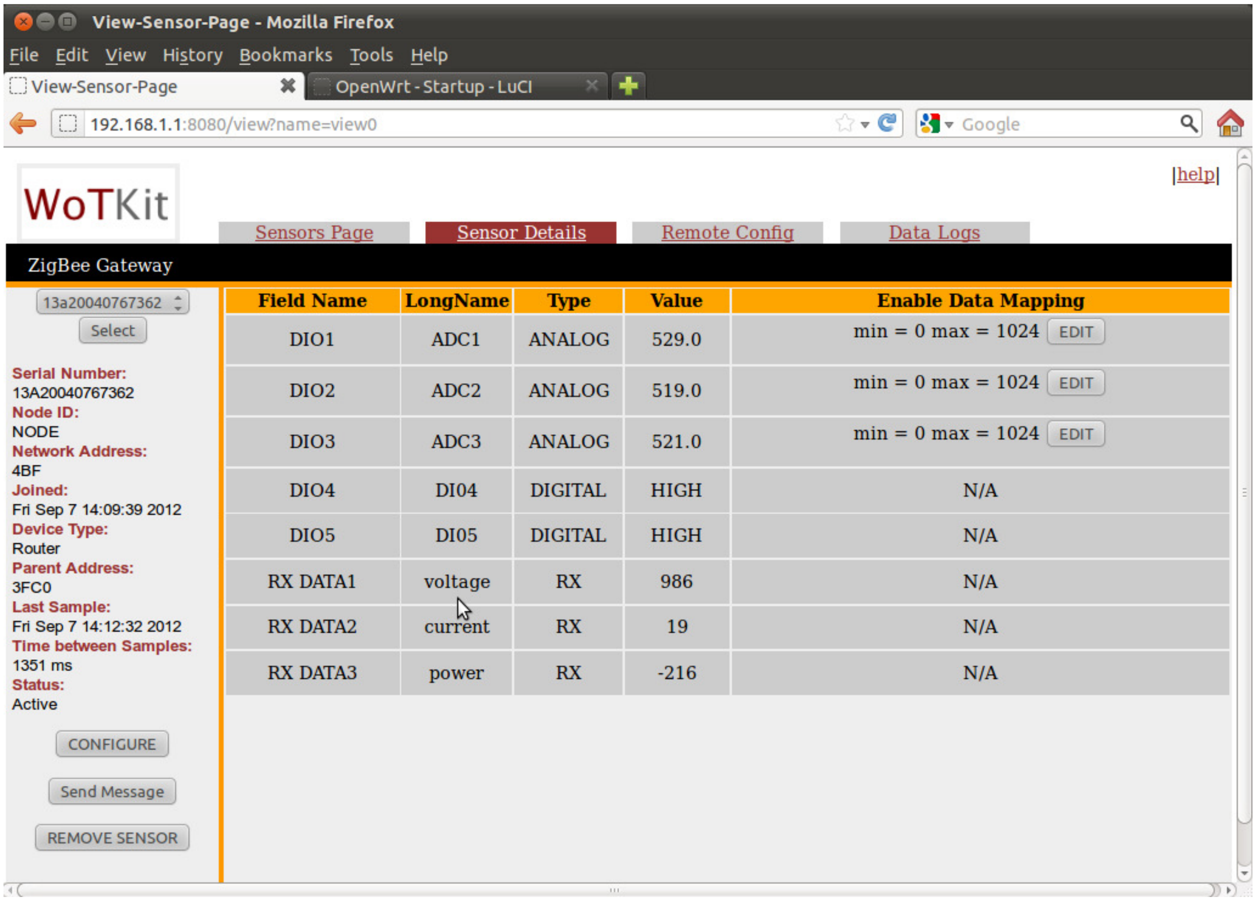
The System supports advanced sensing devices that can send sensor readings via Serial Messages over a ZigBee Network. For example, a “Smart Sensor” could be an Arduino with XBee attached to its UART that is doing some fast sampling and some on-board processing (such as power factor calculation or a

Fast Fourier Transform). The Smart Sensor device can send any type of Data to the gateway via wireless Serial Transmission as long as it follows the specified format.

Readings must contain the Name or ID of the reading followed by the value. The ID and reading must both be terminated by a colon

Multiple readings can be sent in the same packet as long as each reading is separated by colons and as long as the total length is less than 72 bytes (XBee packet length limitation). A sample reading would look like this

- I.e. [Reading 1 ID];[Reading 1 Value];[Reading 2 ID];[Reading 2 Value];



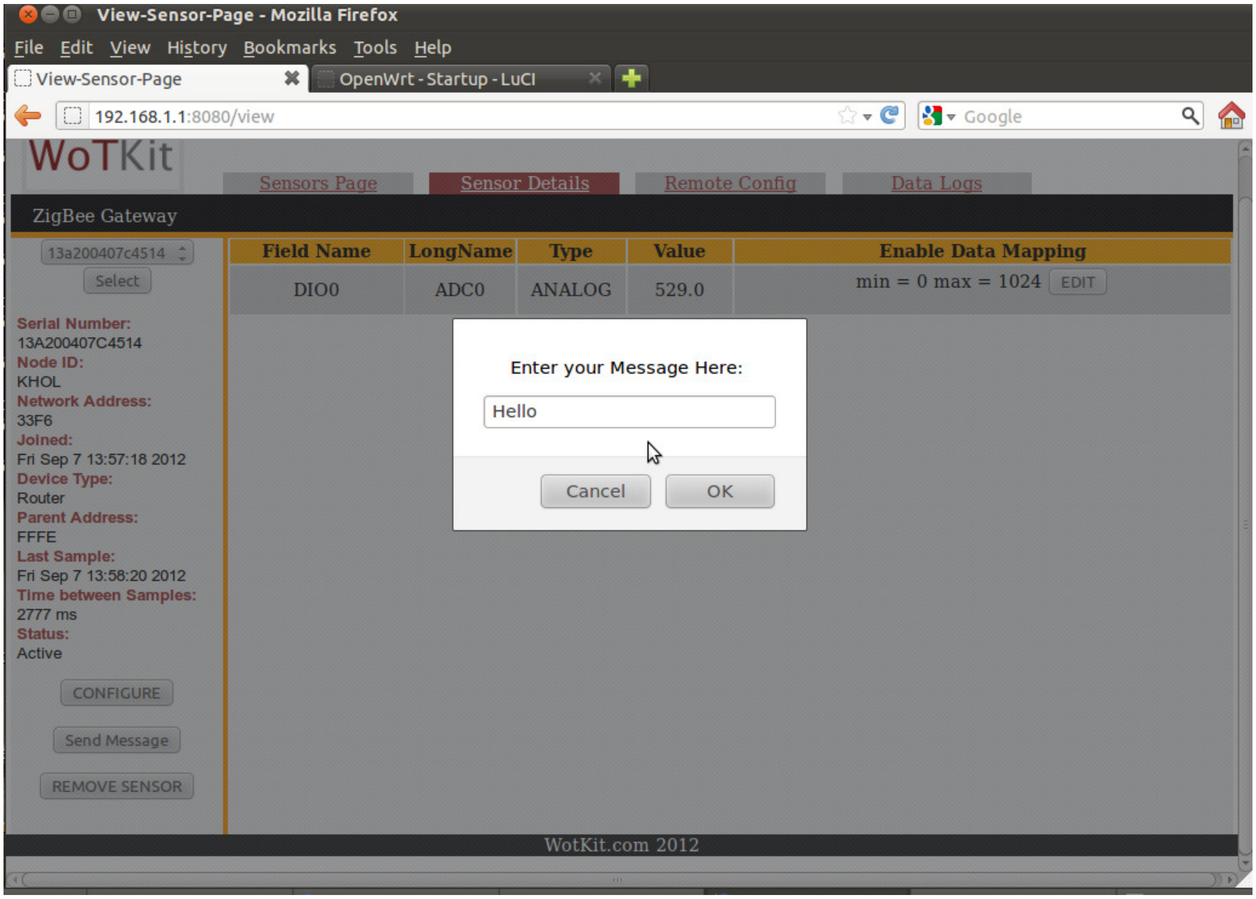
Serial Number:
13A20040767362
Node ID:
NODE
Network Address:
4BF
Joined:
Fri Sep 7 14:09:39 2012
Device Type:
Router
Parent Address:
3FC0
Last Sample:
Fri Sep 7 14:12:32 2012
Time between Samples:
1351 ms
Status:
Active

Field Name	LongName	Type	Value	Enable Data Mapping
DIO1	ADC1	ANALOG	529.0	min = 0 max = 1024 EDIT
DIO2	ADC2	ANALOG	519.0	min = 0 max = 1024 EDIT
DIO3	ADC3	ANALOG	521.0	min = 0 max = 1024 EDIT
DIO4	DI04	DIGITAL	HIGH	N/A
DIO5	DI05	DIGITAL	HIGH	N/A
RX DATA1	voltage	RX	986	N/A
RX DATA2	current	RX	19	N/A
RX DATA3	power	RX	-216	N/A

Send a Message over the network

The user can send a Serial Message to a specified Sensor Node. This can be useful for signaling a “Smart Sensor” to send data or to stop and start taking readings etc.

This could potentially enable higher levels of control of more advanced devices for home automation applications.



Serial Number:
13A200407C4514
Node ID:
KHOL
Network Address:
33F6
Joined:
Fri Sep 7 13:57:18 2012
Device Type:
Router
Parent Address:
FFFE
Last Sample:
Fri Sep 7 13:58:20 2012
Time between Samples:
2777 ms
Status:
Active

Field Name	LongName	Type	Value	Enable Data Mapping
DIO0	ADC0	ANALOG	529.0	min = 0 max = 1024 EDIT

Enter your Message Here:

Hello

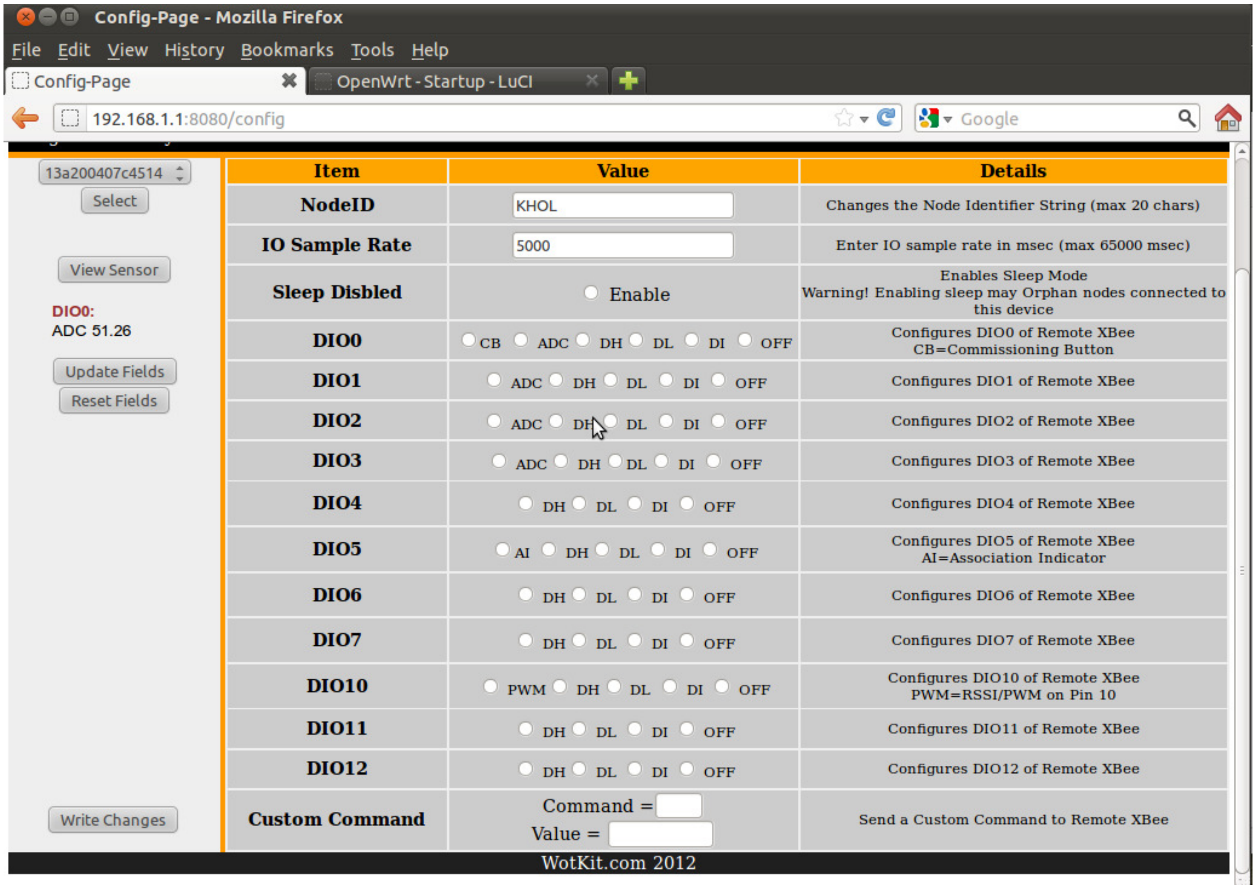
Remove Sensor

Removes the Sensor from the network by Signaling the remote node to Disconnect from the Current PAN.

Configuration Page

This Page allows the user configure any of the networked Nodes from the dropdown list remotely.

The current values of the setting are displayed in the text fields, such as Node ID, Sample Rate. The Current Sensor IO channel settings and readings are displayed in the left column along with their DIO identifier.



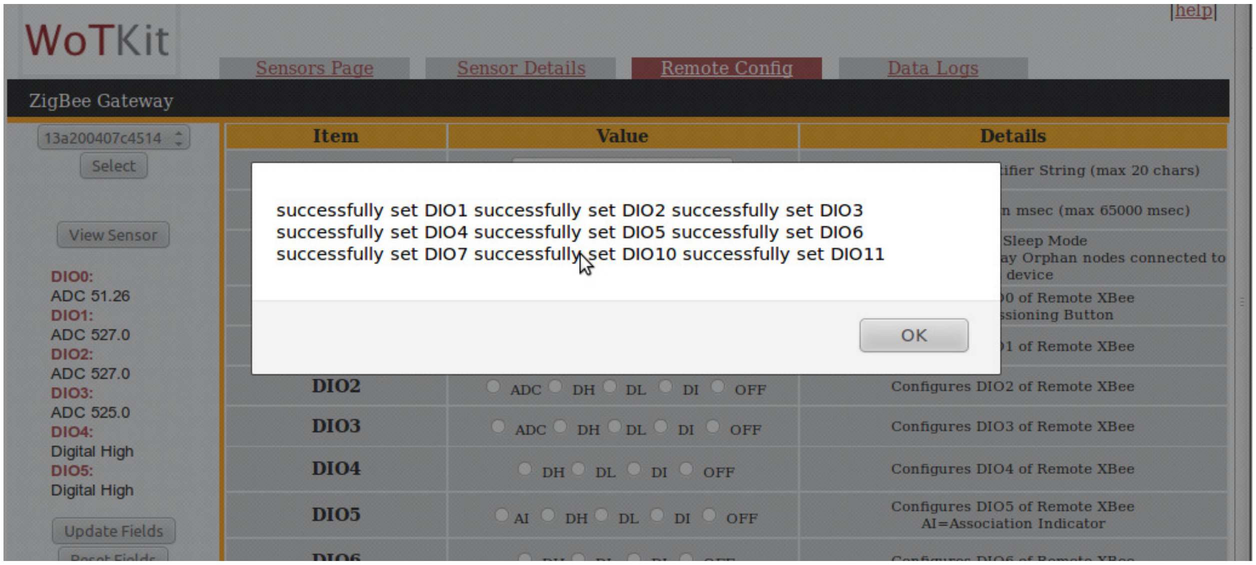
13a200407c4514

DIO0:
ADC 51.26

Item	Value	Details
NodeID	KHOL	Changes the Node Identifier String (max 20 chars)
IO Sample Rate	5000	Enter IO sample rate in msec (max 65000 msec)
Sleep Disbled	○ Enable	Enables Sleep Mode Warning! Enabling sleep may Orphan nodes connected to this device
DIO0	○ CB ○ ADC ○ DH ○ DL ○ DI ○ OFF	Configures DIO0 of Remote XBee CB=Commissioning Button
DIO1	○ ADC ○ DH ○ DL ○ DI ○ OFF	Configures DIO1 of Remote XBee
DIO2	○ ADC ○ DH ○ DL ○ DI ○ OFF	Configures DIO2 of Remote XBee
DIO3	○ ADC ○ DH ○ DL ○ DI ○ OFF	Configures DIO3 of Remote XBee
DIO4	○ DH ○ DL ○ DI ○ OFF	Configures DIO4 of Remote XBee
DIO5	○ AI ○ DH ○ DL ○ DI ○ OFF	Configures DIO5 of Remote XBee AI=Association Indicator
DIO6	○ DH ○ DL ○ DI ○ OFF	Configures DIO6 of Remote XBee
DIO7	○ DH ○ DL ○ DI ○ OFF	Configures DIO7 of Remote XBee
DIO10	○ PWM ○ DH ○ DL ○ DI ○ OFF	Configures DIO10 of Remote XBee PWM=RSSI/PWM on Pin 10
DIO11	○ DH ○ DL ○ DI ○ OFF	Configures DIO11 of Remote XBee
DIO12	○ DH ○ DL ○ DI ○ OFF	Configures DIO12 of Remote XBee
Custom Command	Command = Value =	Send a Custom Command to Remote XBee

WotKit.com 2012

Settings do not change until Update Fields Button is pressed. This allows for multiple fields to be set at once. If the Setting was successfully changed on the remote Node, an acknowledgement is returned and a notification message pops up to inform the user.



successfully set DIO1 successfully set DIO2 successfully set DIO3 successfully set DIO4 successfully set DIO5 successfully set DIO6 successfully set DIO7 successfully set DIO10 successfully set DIO11

DIO0:
ADC 51.26
DIO1:
ADC 527.0
DIO2:
ADC 527.0
DIO3:
ADC 525.0
DIO4:
Digital High
DIO5:
Digital High

Item	Value	Details
		ifier String (max 20 chars)
		n msec (max 65000 msec)
		Sleep Mode ay Orphan nodes connected to device
		0 of Remote XBee ssioning Button
		1 of Remote XBee
DIO2	○ ADC ○ DH ○ DL ○ DI ○ OFF	Configures DIO2 of Remote XBee
DIO3	○ ADC ○ DH ○ DL ○ DI ○ OFF	Configures DIO3 of Remote XBee
DIO4	○ DH ○ DL ○ DI ○ OFF	Configures DIO4 of Remote XBee
DIO5	○ AI ○ DH ○ DL ○ DI ○ OFF	Configures DIO5 of Remote XBee AI=Association Indicator

Brief Explanation of the Fields

More information about the fields can be found on digi.com

Node ID

Allows the user to set / change the Node ID string of the XBee which makes it easier to identify (i.e.: OUTSIDE TEMP SENSOR).The Node ID is limited to 20 chars.

IO Sample Rate

Sets how often the device takes samples of its IO pins in milliseconds. The Gateway forces a minimum sample rate of 500 ms to prevent overloading the system. This could be reduced in future revisions. Maximum Allowable is 65000ms. (Note that if you wish to have faster samples than 500 ms, you can manually set the sample rate using the X-CTU software from digi.com for that particular Node. Use at your own risk)

Sleep Enable/Disable

Enables the Device to become an End Device and Sleep or switch to Router.

- If the Device is acting as a Router Sleep Fields are Hidden until Enable is requested
- If the Device is an End Device, Sleep **Disable** Option appears and the Sleep Fields are shown

Sleep Fields

Displayed only when the Device is an End Device

- **Sleep Rate** – Allows the User to set the amount of time the Device will go to sleep periodically. The units for this setting are tens of milliseconds (I.e. 2000 = 20 seconds). Max Sleep time is 28 seconds.
- **Time Until Sleep** – Sets the amount of time the device will stay awake before going back to sleep for another period in milliseconds. If the user attempts to enter a value less than the sample rate, the value will be set to the current sample rate. This is done to ensure that data is always sent when the device wakes up.

IO Channel Settings

IO channels can be enabled, disabled and configured by selecting the corresponding radio buttons

ADC = Enable ADC on Channel

AI = Option only valid for DIO5 to enable the Association Indicator to generate status LED blink codes on DIO5.

CB= Option only valid for DIO0 to set as Commissioning Button. Commissioning button allows a device to join a network when the button is pushed. See XBee Documentation for more Info.

DH = Sets the Pin High

DL = Sets the Pin Low

DI = Sets Pin as Digital Input

OFF = Disable Use of the Pin, Channel Readings will not be sent for this channel

PWM = Only Valid for DI010. This enables the Node to generate a PWM signal on DIO10 signaling the RSSI of the Radio Signal.

Send Custom Command

Allows the user to send any valid AT command to the Remote Node. This gives an advanced user more control over the Node to set any of the registers. See XBee Documentation on digi.com for more details.

Writing the Settings to Non-Volatile Memory

All changed settings take effect immediately once Update Fields is pressed, however, the settings are not written to non-volatile memory. If the device power cycles, any new settings are lost and the previous settings are loaded.

The on-board Non-volatile memory has a limited number of write cycles, therefore the Gateway enforces the user to specifically notify the Node to commit the changes to Non-Volatile Memory .This is done by pressing the **Write Changes** in the bottom of the left column.

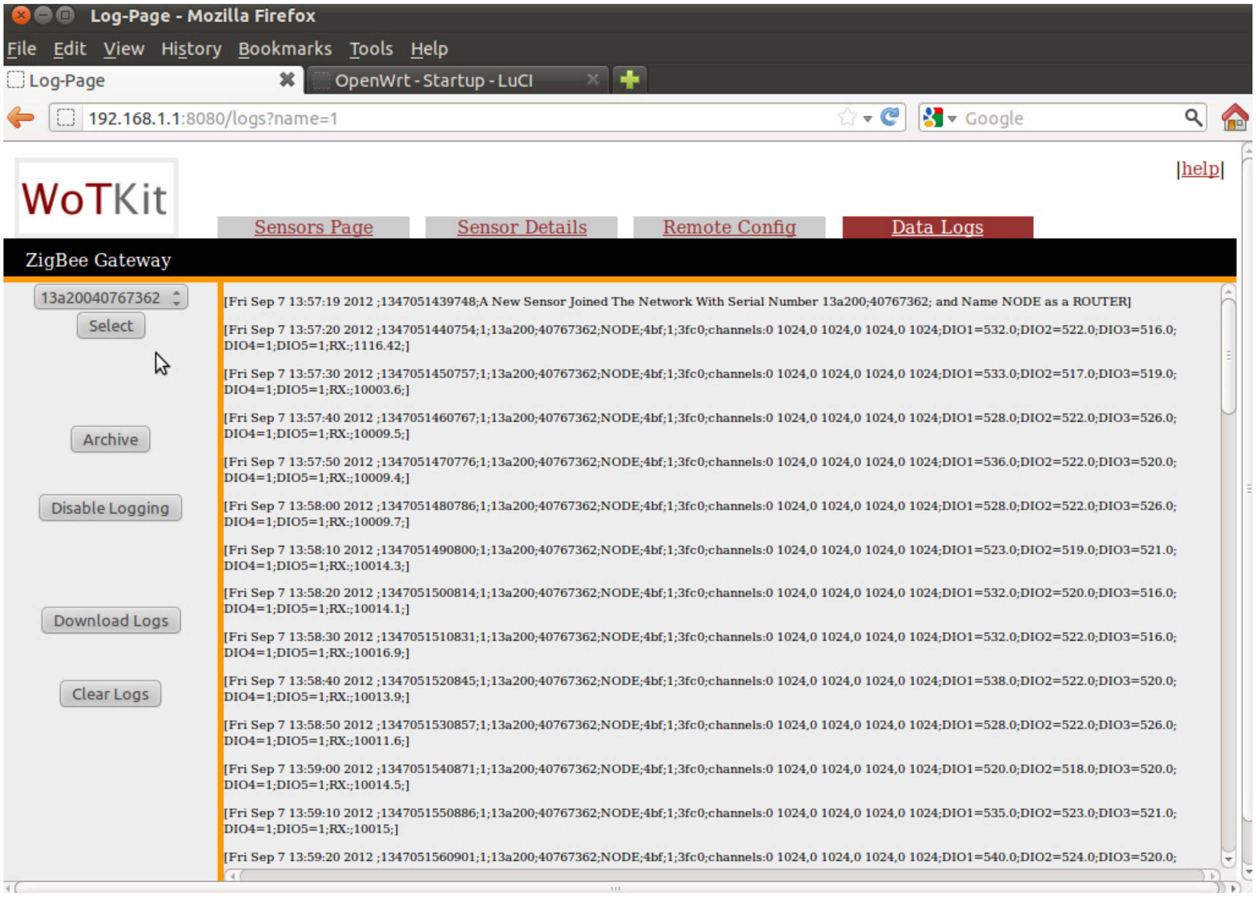
Data Logging

The Gateway simultaneously logs sensor activity on top of pushing data to WotKit. This ensures that valuable sensor data will be backed up. Logs can be disabled from either the Main Sensor Page or the Data Logs page

Viewing Logs

Logs are displayed in a scrolling window which the user can go through to view all activity on the Sensor Network and are each dated and given a timestamp.

The Logs can be filtered by Node from the drop down menu. This is useful if you only want to view the activity of a specific Device.



```
[Fri Sep 7 13:57:19 2012 ;1347051439748;A New Sensor Joined The Network With Serial Number 13a200;40767362; and Name NODE as a ROUTER]

[Fri Sep 7 13:57:20 2012 ;1347051440754;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=532.0;DIO2=522.0;DIO3=516.0;
DIO4=1;DIO5=1;RX:;1116.42;]

[Fri Sep 7 13:57:30 2012 ;1347051450757;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=533.0;DIO2=517.0;DIO3=519.0;
DIO4=1;DIO5=1;RX:;10003.6;]

[Fri Sep 7 13:57:40 2012 ;1347051460767;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=528.0;DIO2=522.0;DIO3=526.0;
DIO4=1;DIO5=1;RX:;10009.5;]

[Fri Sep 7 13:57:50 2012 ;1347051470776;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=536.0;DIO2=522.0;DIO3=520.0;
DIO4=1;DIO5=1;RX:;10009.4;]

[Fri Sep 7 13:58:00 2012 ;1347051480786;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=528.0;DIO2=522.0;DIO3=526.0;
DIO4=1;DIO5=1;RX:;10009.7;]

[Fri Sep 7 13:58:10 2012 ;1347051490800;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=523.0;DIO2=519.0;DIO3=521.0;
DIO4=1;DIO5=1;RX:;10014.3;]

[Fri Sep 7 13:58:20 2012 ;1347051500814;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=532.0;DIO2=520.0;DIO3=516.0;
DIO4=1;DIO5=1;RX:;10014.1;]

[Fri Sep 7 13:58:30 2012 ;1347051510831;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=532.0;DIO2=522.0;DIO3=516.0;
DIO4=1;DIO5=1;RX:;10016.9;]

[Fri Sep 7 13:58:40 2012 ;1347051520845;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=538.0;DIO2=522.0;DIO3=520.0;
DIO4=1;DIO5=1;RX:;10013.9;]

[Fri Sep 7 13:58:50 2012 ;1347051530857;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=528.0;DIO2=522.0;DIO3=526.0;
DIO4=1;DIO5=1;RX:;10011.6;]

[Fri Sep 7 13:59:00 2012 ;1347051540871;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=520.0;DIO2=518.0;DIO3=520.0;
DIO4=1;DIO5=1;RX:;10014.5;]

[Fri Sep 7 13:59:10 2012 ;1347051550886;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=535.0;DIO2=523.0;DIO3=521.0;
DIO4=1;DIO5=1;RX:;10015;]

[Fri Sep 7 13:59:20 2012 ;1347051560901;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=540.0;DIO2=524.0;DIO3=520.0;
```

Archiving Logs

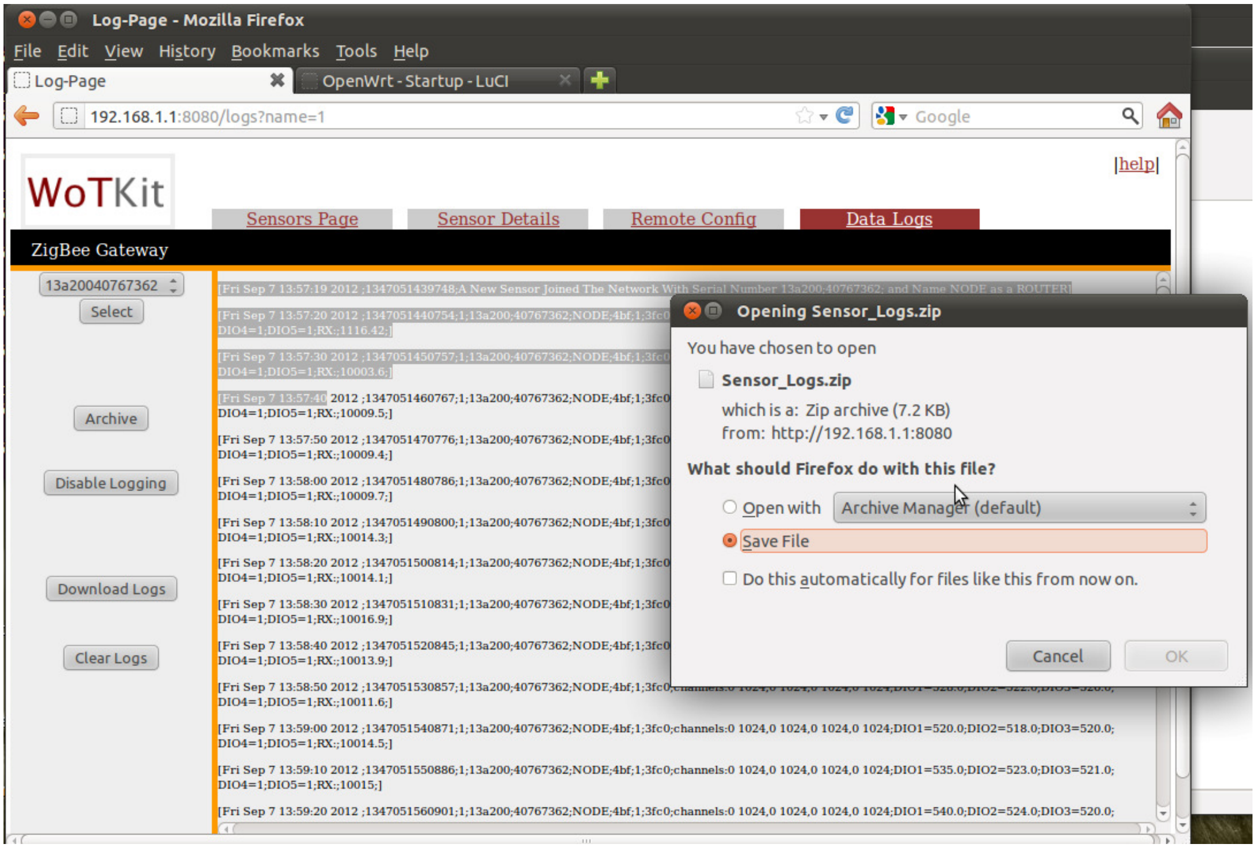
Logs are auto-archived every 24hours but the user can also trigger an archive at any time.

When an archive happens, the current Log Files are appended, dated, saved and compressed into a zip file and stored on the file System. Then the current Log Screen is cleared / refreshed. The file system can store up to 700MB of compressed logs.

Downloading and Retrieving the Logs

The user can download the archived logs at anytime through their browser just by clicking Download Logs. Clicking Download Logs button starts a download of the zipped folder (Sensor_Logs.zip)

The user can then unzip the folder and view all the saved logs as a .txt file.



```
[Fri Sep 7 13:57:19 2012 ;1347051439748;A New Sensor Joined The Network With Serial Number 13a200;40767362; and Name NODE as a ROUTER]

[Fri Sep 7 13:57:20 2012 ;1347051440754;1;13a200;40767362;NODE;4bf;1;3fc0
DIO4=1;DIO5=1;RX:;1116.42;]

[Fri Sep 7 13:57:30 2012 ;1347051450757;1;13a200;40767362;NODE;4bf;1;3fc0
DIO4=1;DIO5=1;RX:;10003.6;]

[Fri Sep 7 13:57:40 2012 ;1347051460767;1;13a200;40767362;NODE;4bf;1;3fc0
DIO4=1;DIO5=1;RX:;10009.5;]

[Fri Sep 7 13:57:50 2012 ;1347051470776;1;13a200;40767362;NODE;4bf;1;3fc0
DIO4=1;DIO5=1;RX:;10009.4;]

[Fri Sep 7 13:58:00 2012 ;1347051480786;1;13a200;40767362;NODE;4bf;1;3fc0
DIO4=1;DIO5=1;RX:;10009.7;]

[Fri Sep 7 13:58:10 2012 ;1347051490800;1;13a200;40767362;NODE;4bf;1;3fc0
DIO4=1;DIO5=1;RX:;10014.3;]

[Fri Sep 7 13:58:20 2012 ;1347051500814;1;13a200;40767362;NODE;4bf;1;3fc0
DIO4=1;DIO5=1;RX:;10014.1;]

[Fri Sep 7 13:58:30 2012 ;1347051510831;1;13a200;40767362;NODE;4bf;1;3fc0
DIO4=1;DIO5=1;RX:;10016.9;]

[Fri Sep 7 13:58:40 2012 ;1347051520845;1;13a200;40767362;NODE;4bf;1;3fc0
DIO4=1;DIO5=1;RX:;10013.9;]

[Fri Sep 7 13:58:50 2012 ;1347051530857;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=520.0;DIO2=522.0;DIO3=520.0;
DIO4=1;DIO5=1;RX:;10011.6;]

[Fri Sep 7 13:59:00 2012 ;1347051540871;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=520.0;DIO2=518.0;DIO3=520.0;
DIO4=1;DIO5=1;RX:;10014.5;]

[Fri Sep 7 13:59:10 2012 ;1347051550886;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=535.0;DIO2=523.0;DIO3=521.0;
DIO4=1;DIO5=1;RX:;10015;]

[Fri Sep 7 13:59:20 2012 ;1347051560901;1;13a200;40767362;NODE;4bf;1;3fc0;channels:0 1024,0 1024,0 1024,0 1024;DIO1=540.0;DIO2=524.0;DIO3=520.0;
```

Opening Sensor_Logs.zip

You have chosen to open

Sensor_Logs.zip

which is a: Zip archive (7.2 KB)
from: http://192.168.1.1:8080

What should Firefox do with this file?

○ Open with Archive Manager (default)

● Save File

□ Do this automatically for files like this from now on.

Cancel OK

Clearing Logs

The Clear Logs button clears all the current logs and deletes any compressed Log Data. A message prompt is triggered to confirm that is the action the user really wants to take in order to prevent accidental file deletion.

-

APPENDIX A: DESCRIPTION OF HOW TO CONFIGURE THE ZIGBEE SENSORS FOR USE WITH MESH NETWORKS AND THE GATEWAY

From DESIGN OF A ZIGBEE SENSOR INTERNET GATEWAY, by Khol Haggerty
Dec 12th 2011. UBC Electrical Engineering

Setting up the Coordinator with X-CTU

1. Select the proper function set. **ZNet 2.5 Coordinator API** should be selected because we will use API structured packets to communicate over network.

2. Since the Coordinator selects the operating channel and PAN. Must set the ID value in X-CTU to a desired PAN. In this case I chose 1234 to be the PAN ID.

3. To ensure new sensor nodes can join the network at any time, ensure the NJ (Node Join Time) parameter is set to 0xFF.

4. Set the destination address. The destination address tells the XBee module which other XBee module to communicate with. Setting to a specific value tells the module to communicate with just that single node, which is useful if you needed only connect only 2 nodes. By setting the destination address to broadcast, we can enable communication to many nodes on the network. Individual node addressing however can still be achieved through the API protocol. set the Broadcast address is set as follows:

   DH (Destination Address High (MSB)) = 0x00000000

   DL (Destination address low (LSB)) = 0x0000FFFF

5. It is often difficult to remember which sensor node is which just by looking at the 64-bit serial number, so it is a good idea to give the XBee modules on your network a unique name. This helps the user identify different sensor nodes when remotely configuring the network. Each XBee node can have a 20 character ASCII string assigned to Node ID (NI). Label the Coordinator COORDINATOR to distinguish it as the Coordinator.

6. Set the API mode. For this network select AP (API mode) = 2. This enables escaped characters. This is necessary for use with the XBee Arduino library and offers the advantage by preventing some specific API identifiers from being confused with API frame data.

Configuration of the Coordinator is complete.

Setting up a Router / End Device with X-CTU

Routers and End Devices are set up very similar. The difference between an End Device and a Router is that an End Devices will have sleep enabled.

1. To begin configuring a Router Device, select ZNet2.5 Router/End Device API function set.

2. The Router should attempt to connect to any available PAN. To do this set the PAN ID to FFFF.

3. Ensure that NJ is set to FF to always allow new nodes to connect to the router. This is important for creating a dynamic sensor network.

4. Set Channel Verification JV to 1 to ensure the Routers always switch to the operating channel of the Coordinator. If this is not set properly, the Router may not broadcast on the right channel.

5. There is no need to configure a destination address because the end destination is the Coordinator. Leave DH and DL as 0.

6. Enable AP (API mode) = 2 so the data packets are sent in the proper format

7. Give node a unique Node Identifier NI.

8. To enable the device to collect analog data samples, enable the ADC pins of the Router. There are 4 channels, corresponding to pins DIO0-DIO3. DIO0 should be reserved for commissioning. The commissioning button is used when adding a new sensor to the network. A push of the button sends out a node identifier packet to acknowledge that the node is present and available to participate in a network. Therefore there will be only 3 useable ADC channels for analog measurements. Each ADC is 10-bits long over a range of 0-1200mV.

9. If you wish to use pins as digital IO instead, DIO1-DI03 can be configured as digital. DIO4, DI010-12 are also allowed to be configured as digital. Other pins are available as well, but are considered reserved for the scope of this gateway.

IO Sampling

10. Next Configure the sampling rate. Sample rate defines how often a sensor sends I/0 samples to the Coordinator. Rates can be as low as 1 ms, however this is too fast for useful operation with a

large network. This would overload Arduino gateway with samples. Most sensor data doesn't need to be updated so frequently. Default sample rate is 5 seconds, and is assigned automatically by the gateway at initialization to keep sampling rate in a safe range. The user can further increase sampling rate through the remote configuration page on the gateway, up to a minimum of 1 second and maximum of 65 seconds.

Sleep Modes

11. The only sleep modes that are allowed in my configuration design are 0 – OFF and 5- Cyclic Sleep. The other modes are not applicable for XBees acting as standalone devices. If sleep mode is set to 0, the device acts as a Router and therefore can allow other devices to connect to it. If sleep mode is 5, the device is an End Device and cannot allow other devices to connect to it. If at any time the sleep mode is changed, the device type is also changed. The device will always dissociate from the network first, before re-associating as a different device type. Switching from a Router to an End Device may cause other child devices to become orphaned.

12. ST – Time before sleep. This defines how long the device remains awake until the next sleep cycle. Set this to a value greater than the sample rate so that the module can send a sample when awake. Note values are entered in hexadecimal.

13. SP – Sleep Period. Defines the period of time the device powers down for on each sleep cycle. The minimum acceptable sleep time for my configuration is 1 second. The maximum sleep time is 28 seconds. Note values are entered in hexadecimal.

14. SN – Number of Sleep cycles. This should be set to 1 except in the special case of a low-power node. Total sleep duration is calculated as the value of SP times the value of SN.

15. SO – Sleep Options. Define the options for sleep. Set this to 0 except only in the case of a Low Power node.

Low Power Node

Configuring a Low Power node is similar to configuring a Router/ End Device, with some differences to the sleep mode configuration and additional configuration of power modes. Low power nodes cannot be configured remotely via the gateway. They are special, slave devices can only send sample data. Any further configuration must be done through X-CTU.

1. Set the Sleep Mode SM to 5 for cyclic sleep.

2. Set ST to greater than sample rate. For this device only IO sample rate is strictly 200 milliseconds and time until sleep 250 milliseconds. The device is powered on just long enough to send a quick sample then return to sleep. The less time spent transmitting the more energy will be conserved.

3. The Sleep period is defined by both SP and SN parameters. Sleep period will be SP times SN. For this battery powered sensor, I have chosen to select the sleep period to be 30 mins.

4. I set the sleep period SP to 15 seconds because a quarter of a minute is a nice value to use for calculating the next total sleep period.

5. Set SN to 120 achieve 30 mins.

6. Set Sleep Options SO to 0x04. This option enables the device to sleep for SN*SP.

7. Configure the power of the transmitting radio on the device such that it conserves energy by setting Power Mode PM to 0. Default configuration has boost mode enabled, which outputs 2 dBm signal power. We will want to disable boost mode to reduce output signal to 1 dBm. Broadcast range and sensitivity of the device will reduced however.

8. We will also want to adjust the power Level. Set Power Level PL to 1 to decrease the power level to -3dBm. This halves the power.

9. When an XBee powers down to sleep will consume less than 1uA. This allows the device to be powered by a battery source such as a watch battery.

10. Pin 13 of the low power node will be asserted when the device is awake, and will be pulled down when asleep. This allows control over sensor devices that are attached to the Xbee. This can be made so that the sensors power down as well when the device sleeps. This was the case for my low power example.

With this set-up the network is capable of dynamically configuring itself. When the Coordinator is on-line and chooses a channel, it can allow any device on the PAN to connect to it at any time. The Routers and End Devices are configured to Connect to any available PAN, and therefore they will automatically try to join the Coordinator. Once joined, the Routers will be assigned a 16-Bit network address. This 16-Bit address is assigned to the Router/ End Device once it joins the network. It is unique to each node in the network, however it can change if the device status changes.

1. If the device cannot connect to its parent (in the case where maybe the parent has been removed). If leaves the network and finds a new parent and rejoins. It will be assigned a new address.

2. If the device switches from being a Router to an End Device. When this happens, the device leaves the network, then rejoins its parent, however the new address will be the parents address plus the child number it is of the parent. For example if the parents address is 74 F0, and the device was now the second child of the device, then the new 16 bit address of that device would be 74 F2.

This is the basis for creating a self-healing network