

Zigbee, domotic and open source !



Alexis Lothoré - Meetup 02/18

What will we see ?

- Zigbee : what is it ?
- Zigbee fundamentals
- Our case : the « bulb issue »
- Implementing our very own gateway : steps and feedback
- Next steps and improvements

- **Zigbee : what is it ?**
- Zigbee fundamentals
- Our case : the « bulb issue »
- Implementing our very own gateway : steps and feedback
- Next steps and improvements

Zigbee : what is it ?

- Radio protocol for IoT
- Goals : simplicity, low cost and low energy, scalable
- Specified by Zigbee Alliance
- EU : 868Mhz/2,4Ghz



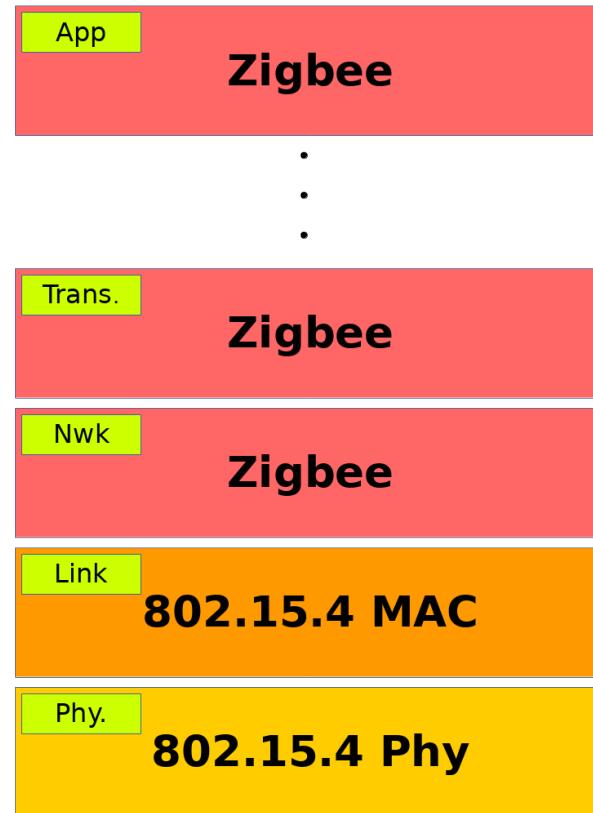
Zigbee : what is it ?



<http://www.zigbee.org/zigbeealliance/our-members/>

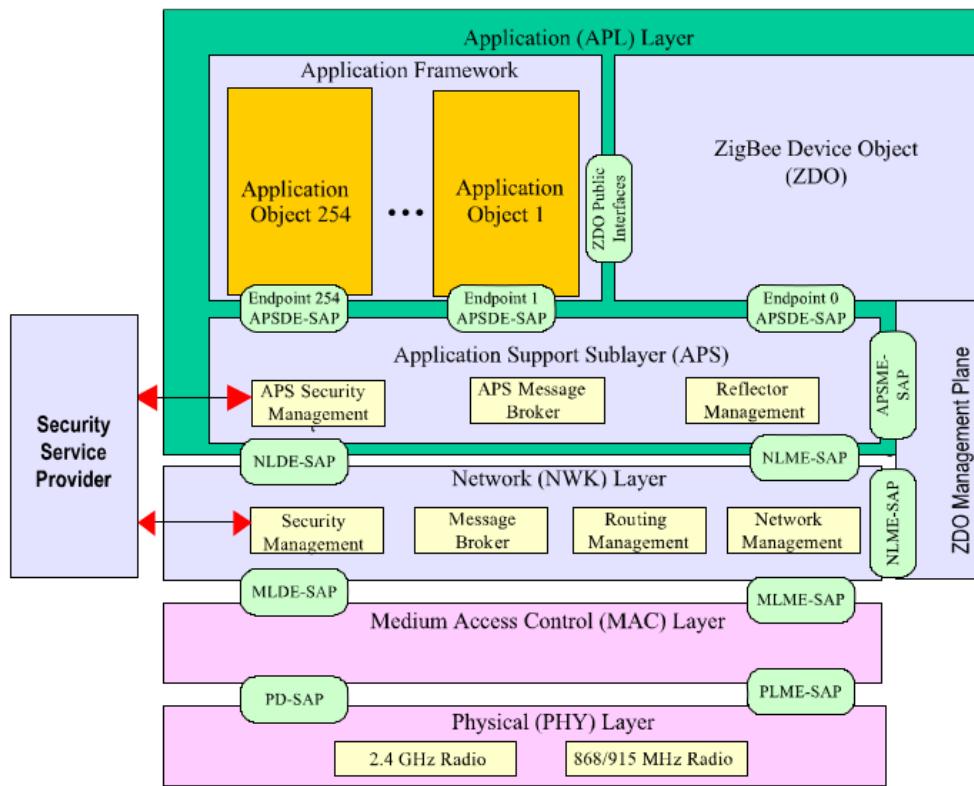
- Zigbee : what is it ?
- **Zigbee fundamentals**
- Our case : the « bulb issue »
- Implementing our very own gateway : steps and feedback
- Next steps and improvements

Zigbee fundamentals



=> Zigbee protocol takes multiple OSI layers in charge

Zigbee fundamentals

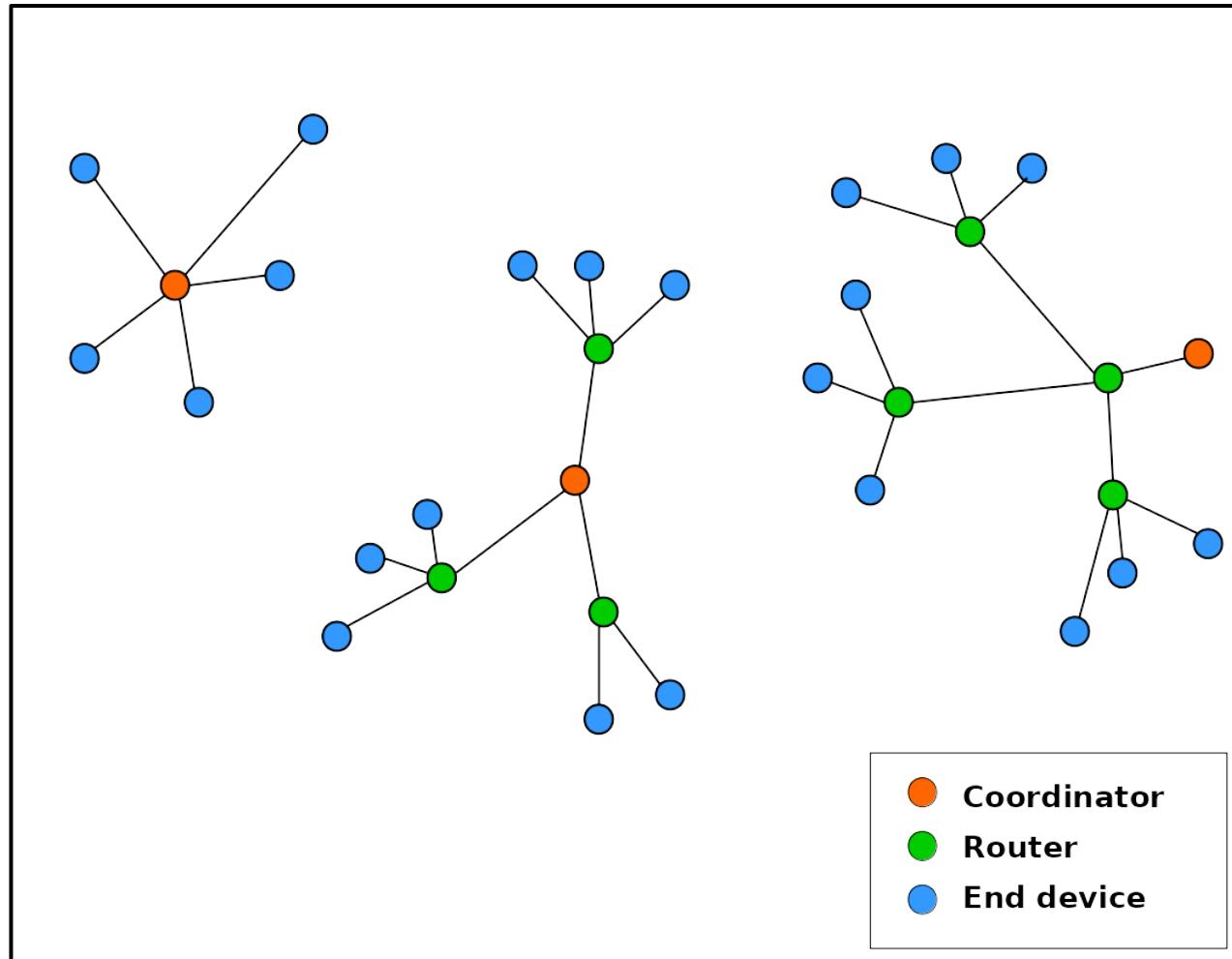


Zigbee specification

- NWK : network layer
- APS : Application Support Sublayer
- ZDO : Zigbee Device Object
- AF : Application framework

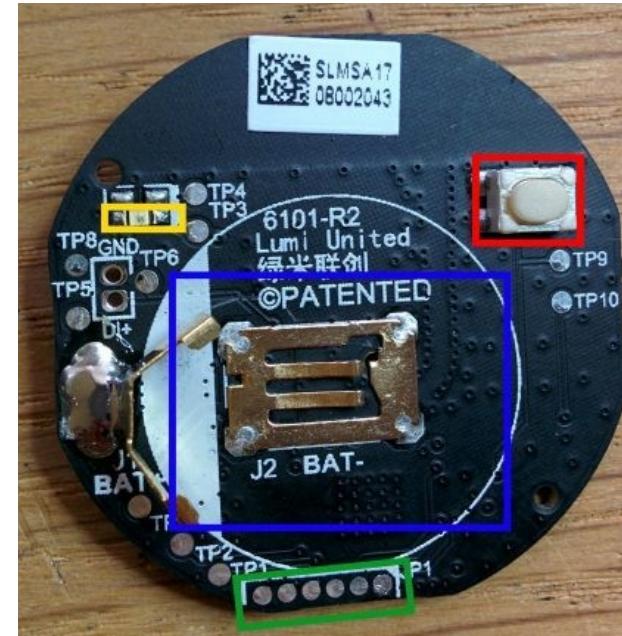
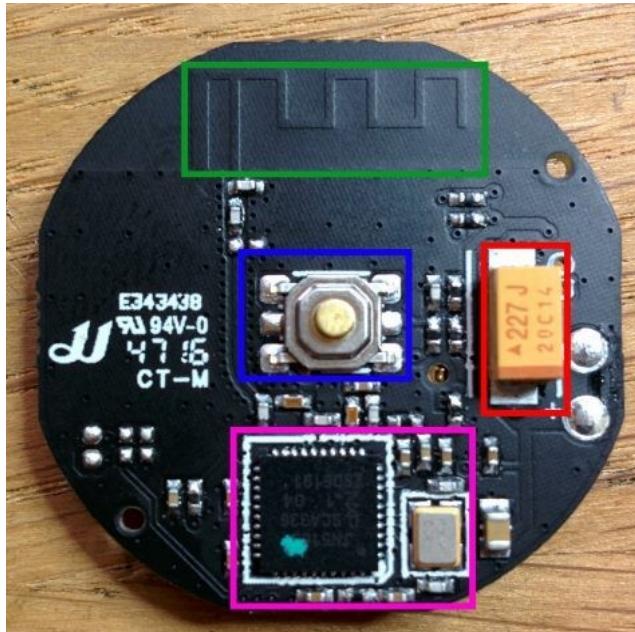
Zigbee fundamentals

- Different network topologies



Zigbee fundamentals

- Low power nodes



<https://faire-ca-soi-meme.fr/domotique/2017/04/15/test-xiaomi-mijia-6-in-1-smart-home/>

Zigbee fundamentals

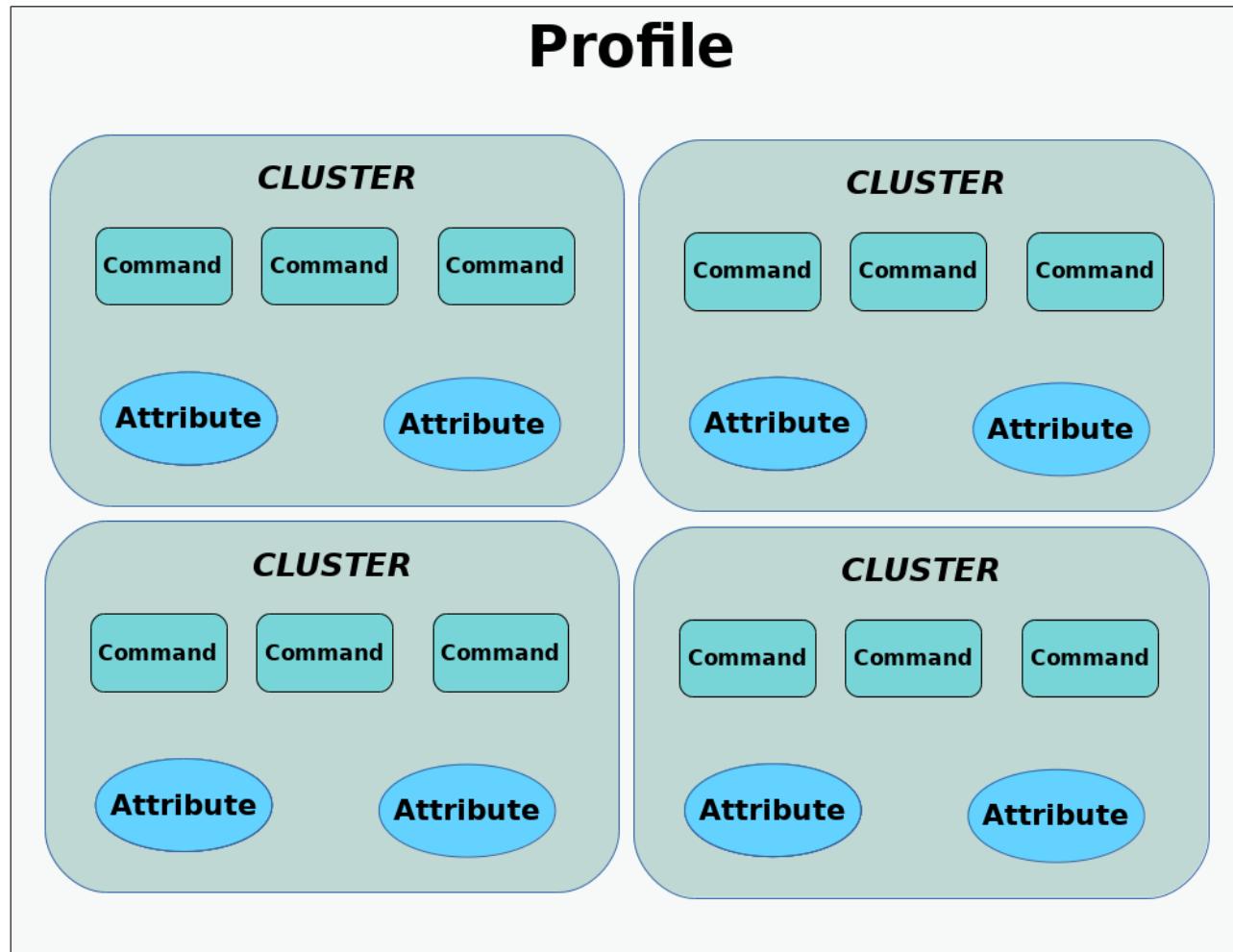
- Routing features
 - Coordinator (and possibly routers) have/has a routing table, and a route discovery table
 - Route requests/replies between nodes to establish and maintain routes : unicast, multicast, many-to-one
 - Routing based on path cost algorithm, with cost variable and estimation set by each manufacturer.
 - Can be driven from higher layers of protocol

Zigbee fundamentals

- Security
 - Based on 128 bits symmetric keys
 - Application security (link key) and/or network security (network key)
 - Key retrievement ?
 - Key transport
 - Key establishment
 - Pre-installation
 - Presence of a « Trust Center »

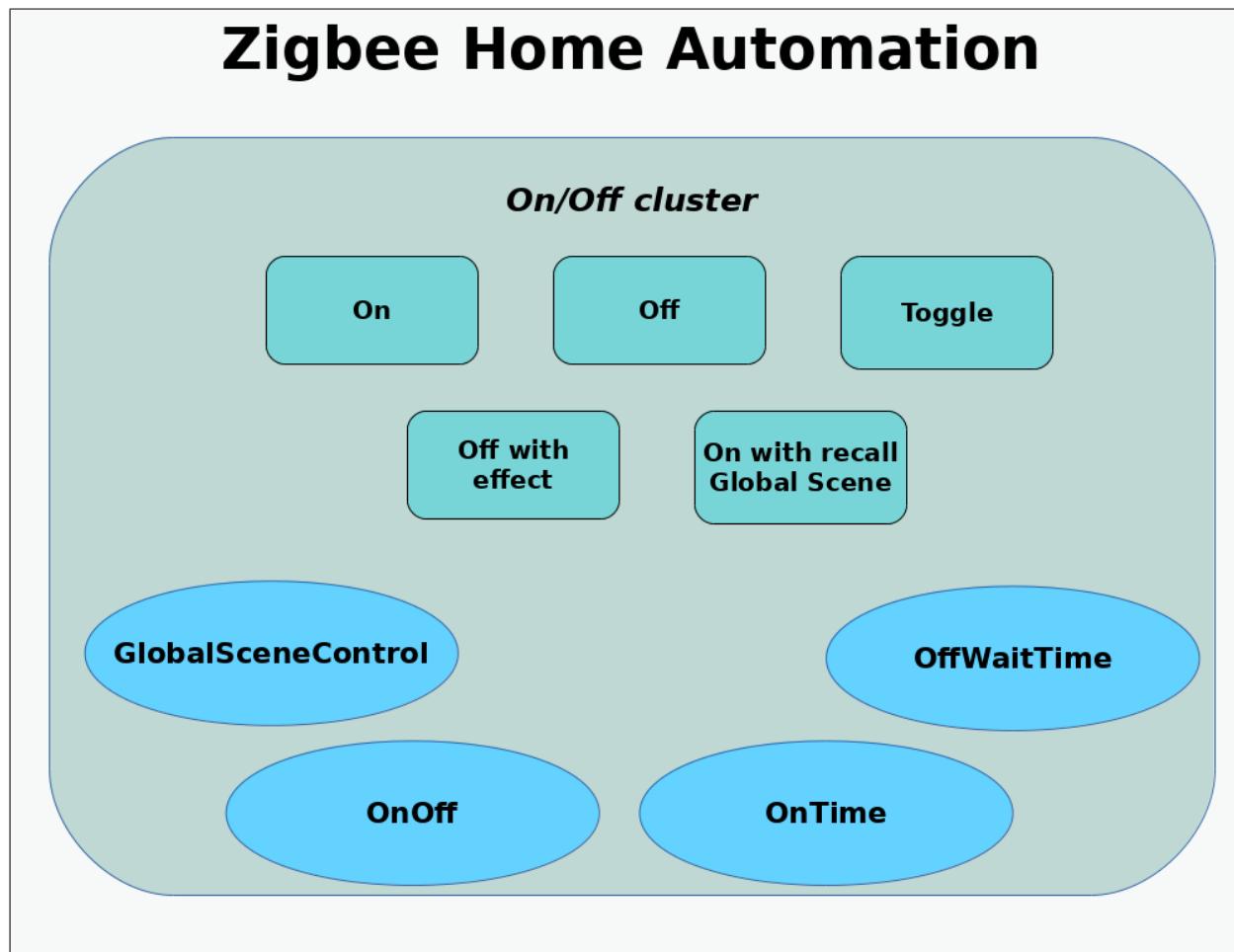
Zigbee fundamentals

- Application framework



Zigbee fundamentals

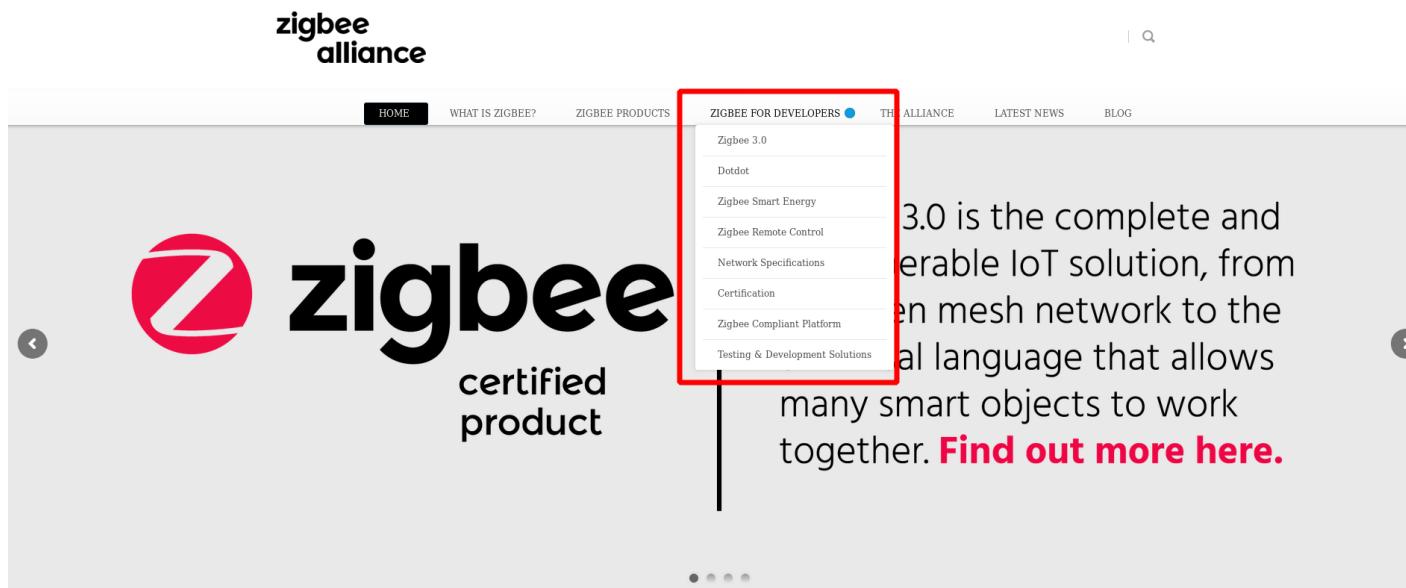
- Application framework



Sources

To Go Further :

- Zigbee specification
- Zigbee Cluster Library Specification
- Zigbee Home Automation Public Application profile



www.zigbee.org

- Zigbee : what is it ?
- Zigbee fundamentals
- Our case : the « bulb issue »
- Implementing our very own gateway : steps and feedback
- Next steps and improvements

Our case : the « bulb issue »

Goals :

- add the bulb to my domotic environment
- If successful, add different devices to it
- With a (very) low cost solution, and embedded
- Software part must be open source
- Increase skills : Zigbee learning

*« So you have got a Hue lamp ?
Great, now go buy a bridge and
download the app ! »*



Nope !

Our case : the « bulb issue »

Existing solutions

zigbee-shepherd

kappaIO

cc-znp

ZiGate



Does not totally fit my needs

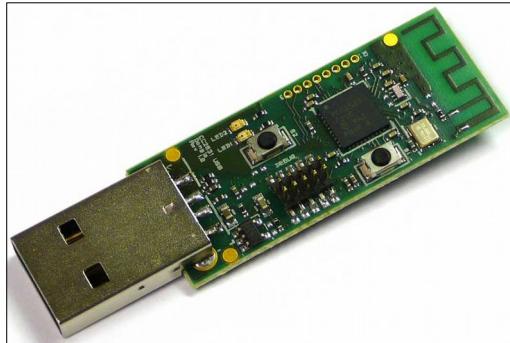
- Zigbee : what is it ?
- Zigbee fundamentals
- Our case : the « bulb issue »
- **Implementing our very own gateway : steps and feedback**
- Next steps and improvements

Implementing our very own gateway

First step : select the hardware for

- Our gateway
- The dev tools (e.g. : sniffer)

The contenders :



www.ti.com

CC253X

VS

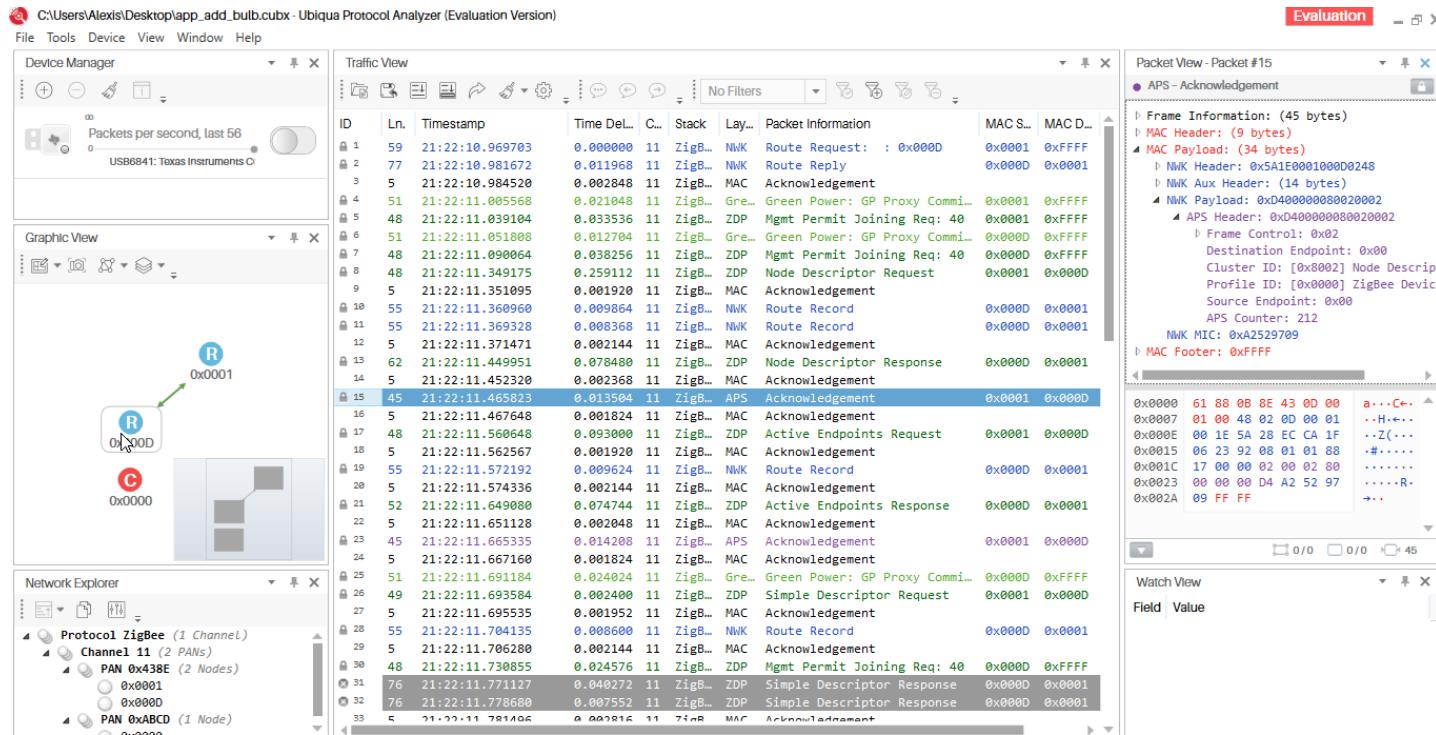


www.nxp.com

JN516x

Implementing our very own gateway

Second step : select the software for sniffing/debugging
The ultimate IoT (Zigbee) radio sniffer : Ubiqua



Implementing our very own gateway

Third step : mix it !



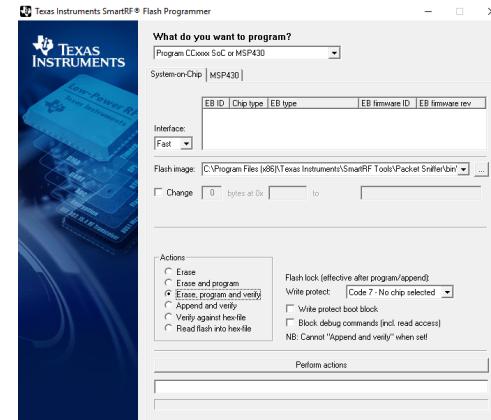
<http://www.ti.com/tool/Z-STACK>

sniffer_fw_cc2430.hex	12/06/2014 14:39	Fichier HEX	10 Ko
sniffer_fw_cc2530.hex	19/06/2014 10:07	Fichier HEX	9 Ko
sniffer_fw_cc2531.hex	19/06/2014 10:07	Fichier HEX	23 Ko
sniffer_fw_cc2533.hex	19/06/2014 10:07	Fichier HEX	8 Ko
sniffer_fw_cc2540.hex	19/06/2014 10:07	Fichier HEX	28 Ko
sniffer_fw_cc2540_uart.hex	12/06/2014 14:39	Fichier HEX	24 Ko
sniffer_fw_cc2540_usb.hex	19/06/2014 10:07	Fichier HEX	47 Ko
sniffer_fw_cc2544.hex	19/06/2014 10:07	Fichier HEX	26 Ko
sniffer_fw_ccx10_usart0_alt1.hex	12/06/2014 14:39	Fichier HEX	7 Ko
sniffer_fw_ccx10_usart1_alt2.hex	12/06/2014 14:39	Fichier HEX	7 Ko
sniffer_fw_ccx11.hex	12/06/2014 14:39	Fichier HEX	21 Ko

C:\|Program Files(x86)|Texas Instruments|SmartRF Tools|\
Packet Sniffer\bin\general\firmware



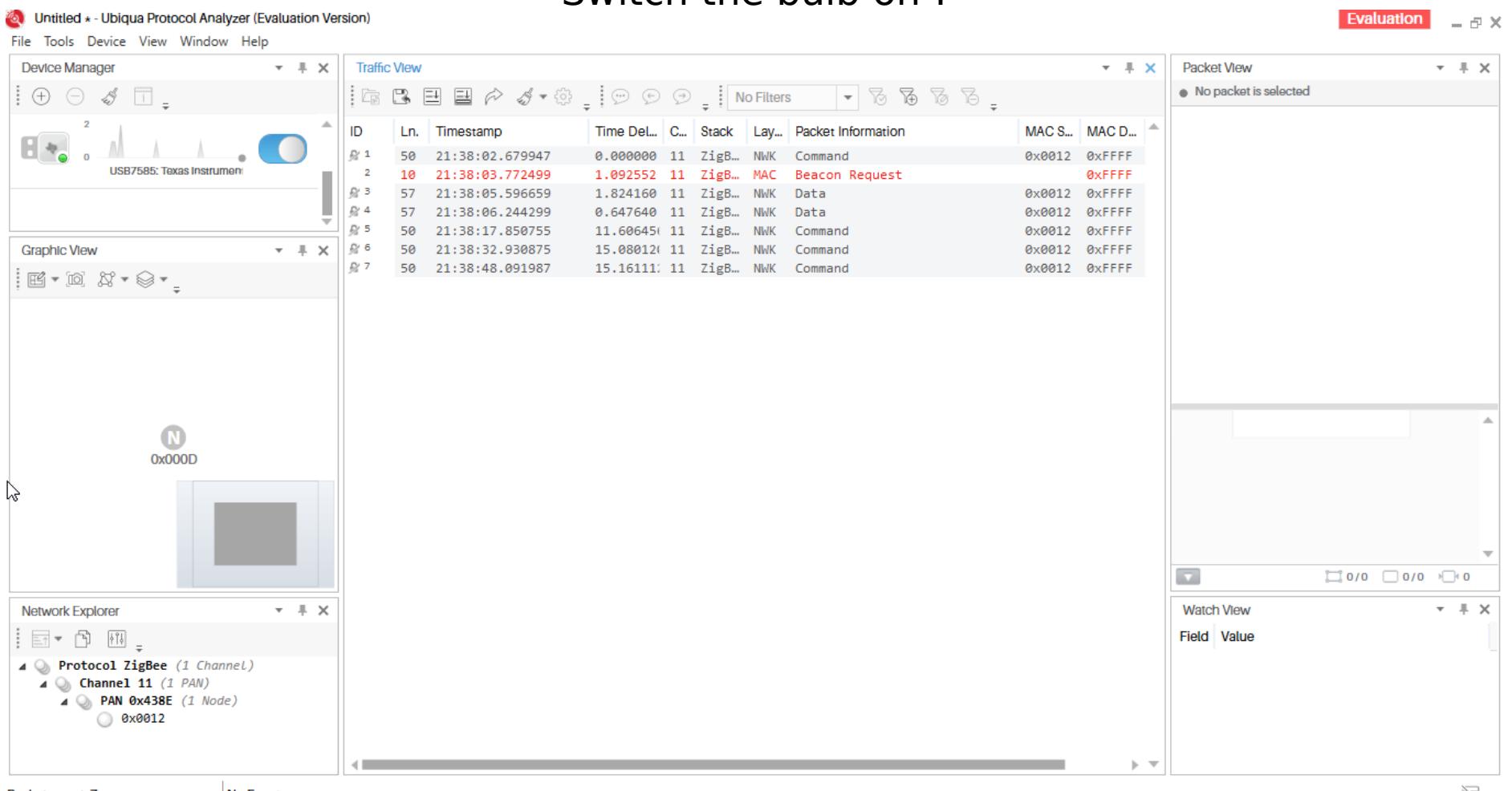
<http://www.ti.com/tool/CC-DEBUGGER>



C:\|Program Files(x86)|Texas Instruments|SmartRF Tools|\
Flash Programmer\bin

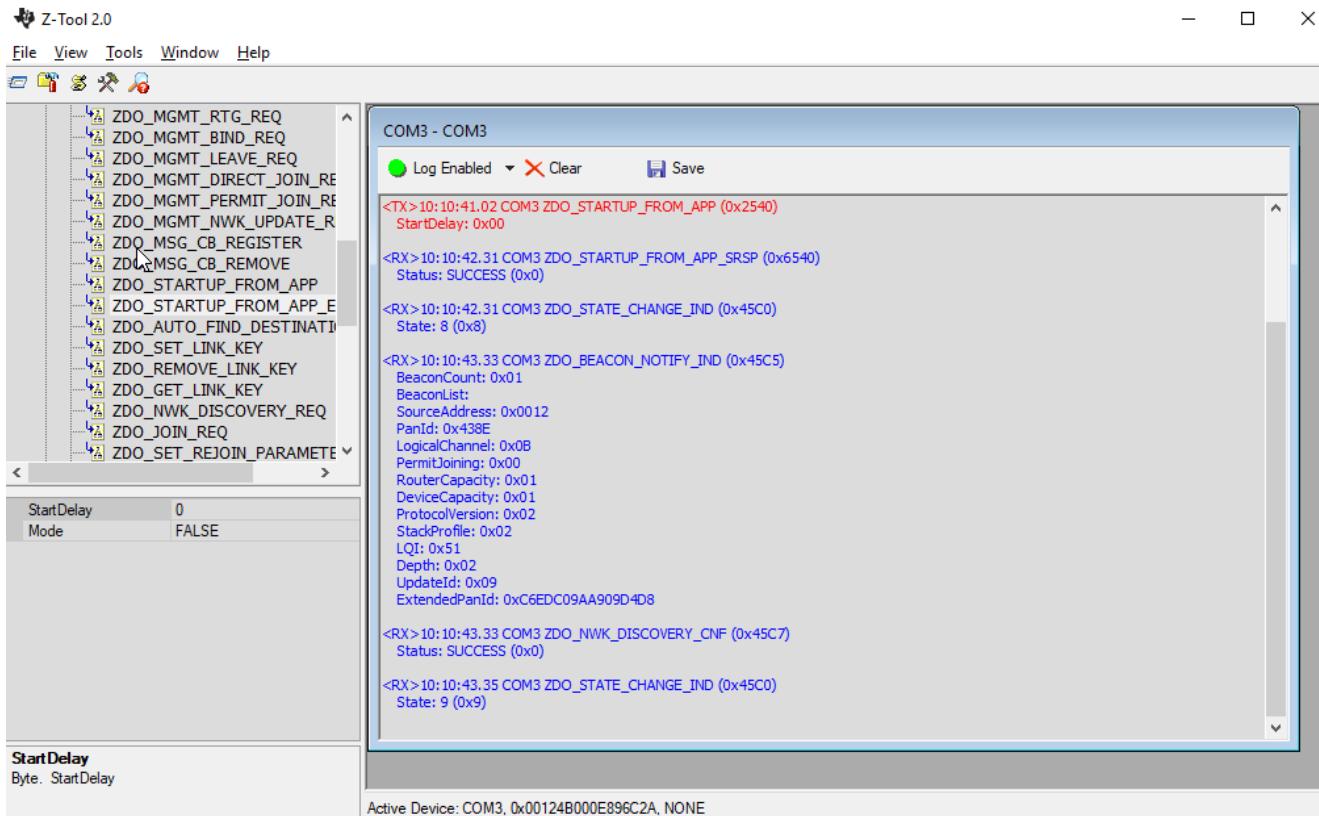
Implementing our very own gateway

Switch the bulb on !



Implementing our very own gateway

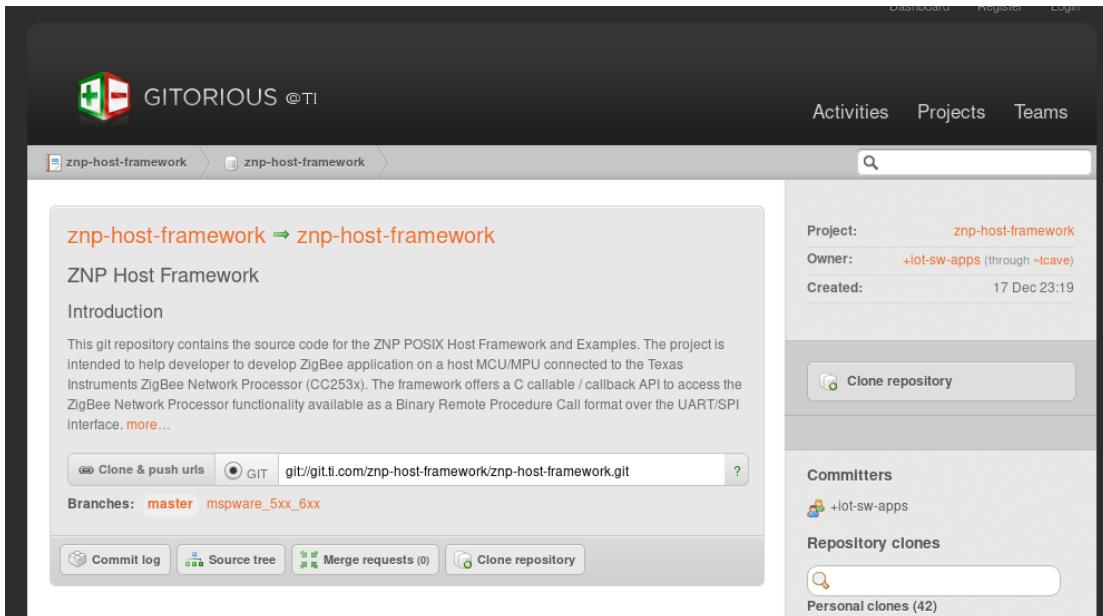
First interactions with bulb : Z-Tool and ZNP firmware



Firmware : C:\Texas Instrument\Z-Stack 3.0.1\Projects\HomeAutomation\GenericApp\CC2531

Implementing our very own gateway

Going further : ZNP host in C



<https://git.ti.com/znp-host-framework/znp-host-framework>

- RPC (remote procedure call) processing : physical medium
- MT (Monitor and Test) processing : commands
 - MT_SYS
 - MT_ZDO
 - MT_AF
 - MT_UTILS
 - ...
- Firmwares
- Example binaries

Implementing our very own gateway

ZNP Recipe



<http://www.ti.com/tool/Z-STACK>

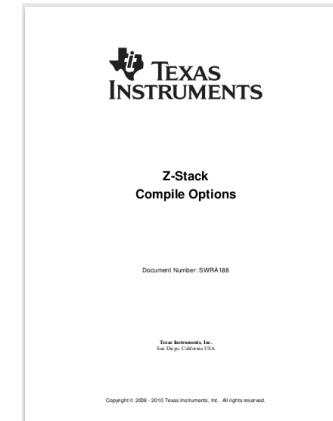


<http://www.ti.com/tool/CC-DEBUGGER>



GenericApp,
Home automation project

<https://www.iar.com/iar-embedded-workbench/#!/?device=CC2531F128&architecture=8051>



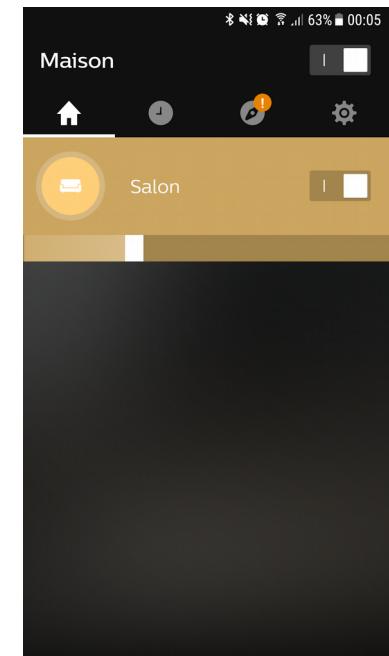
http://e2e.ti.com/cfs-file/_key/communityserver-discussions-components-files/158/4606.z_2d00_stack-compile-options.pdf

Implementing our very own gateway

Hue bridge « reverse engineering »



Hue bridge



Android App

**Goal : intercept traffic between Hue bridge and lamp,
during a « touchlink » procedure**

Implementing our very own gateway

Bulb/bridge first capture while pairing

Traffic View

ID	Ln.	Timestamp	Time Del...	C...	Stack	Lay...	Packet Information	MAC S...	MAC D...
1	47	21:19:19.721024	0.000000	11	ZigB...	NWK	Command	0x0001	0xFFFF
2	50	21:19:24.568320	4.847296	11	ZigB...	NWK	Command	0x1B23	0xFFFF
3	51	21:19:25.928600	1.360280	11	ZigB...	NWK	Command	0x0001	0xFFFF
4	35	21:19:27.030160	1.101560	11	ZigB...	ZCL	Commissioning: Scan Request	00:17...	0xFFFF
5	107	21:19:27.076760	0.046600	11	ZigB...	MAC	MAC	88:01...	00:17
6	5	21:19:27.079416	0.002656	11	ZigB...	MAC	Acknowledgement		
7	35	21:19:27.284440	0.205024	11	ZigB...	ZCL	Commissioning: Scan Request	00:17...	0xFFFF
8	35	21:19:27.536151	0.251712	11	ZigB...	ZCL	Commissioning: Scan Request	00:17...	0xFFFF
9	35	21:19:27.789455	0.253304	11	ZigB...	ZCL	Commissioning: Scan Request	00:17...	0xFFFF
10	35	21:19:28.041248	0.251792	11	ZigB...	ZCL	Commissioning: Scan Request	00:17...	0xFFFF
11	32	21:19:29.123895	1.082648	11	ZigB...	NWK	Link Status	0x0000	0xFFFF
12	41	21:19:32.104952	2.981056	11	ZigB...	ZCL	Commissioning: Identify Req...	00:17...	00:17
13	82	21:19:34.232431	2.127480	11	ZigB...	ZCL	Commissioning: Network Join...	00:17...	00:17
14	5	21:19:34.235440	0.003008	11	ZigB...	MAC	Acknowledgement		
15	40	21:19:34.248640	0.013200	11	ZigB...	ZCL	Commissioning: Network Join...	00:17...	00:17
16	5	21:19:34.250304	0.001664	11	ZigB...	MAC	Acknowledgement		
17	57	21:19:34.383312	0.133008	11	ZigB...	NWK	Data	0x000D	0xFFFF
18	57	21:19:34.396992	0.013680	11	ZigB...	NWK	Data	0x0001	0xFFFF
19	50	21:19:34.823456	0.426464	11	ZigB...	NWK	Command	0x0001	0xFFFF
20	57	21:19:35.031992	0.208536	11	ZigB...	NWK	Data	0x000D	0xFFFF
21	50	21:19:36.266536	1.234544	11	ZigB...	NWK	Data	0x0001	0x000
22	5	21:19:36.268520	0.001984	11	ZigB...	MAC	Acknowledgement		
23	57	21:19:36.277463	0.008944	11	ZigB...	NWK	Data	0x000D	0x000
24	5	21:19:36.279671	0.002208	11	ZigB...	MAC	Acknowledgement		
25	45	21:19:36.293191	0.013520	11	ZigB...	NWK	Data	0x0001	0x000
26	5	21:19:36.295023	0.001832	11	ZigB...	MAC	Acknowledgement		
27	64	21:19:36.334136	0.039112	11	ZigB...	NWK	Data	0x0001	0x000
28	5	21:19:36.336576	0.002440	11	ZigB...	MAC	Acknowledgement		
29	50	21:19:36.345608	0.009032	11	ZigB...	NWK	Data	0x000D	0x000

Packet View - Packet #4

● ZCL – Commissioning: Scan Request

- ▷ Frame Information: (35 bytes)
- ▷ MAC Header: (17 bytes)
- ▲ MAC Payload: (16 bytes)
 - ▷ NWK Header: 0x000B
 - ▲ NWK Payload: (14 bytes)
 - ▲ APS Header: 0xC05E10000B
 - ▷ Frame Control: 0x0B
 - Cluster ID: [0x1000] Commissioning
 - Profile ID: [0xC05E] ZigBee Light Link
 - ▲ APS Payload: (9 bytes)
 - ▲ ZCL Header: 0x000011
 - ▷ Frame Control: 0x11
 - Transaction Sequence Number: 0
 - Command ID: [0x00] Scan Request
 - ▲ ZCL Payload: 0x12053F75B5B2
 - Inter-PAN Transaction ID: 0x3F75B5B2
 - ▷ ZigBee Information: 0x05
 - ▷ Touchlink Information: 0x12
- ▷ MAC Footer: 0xFFFF

```

0x0000 01 C8 D9 FF FF FF FF 8E 43 .....C
0x0009 23 92 08 01 01 88 17 00 0B #.....
0x0012 00 0B 00 10 5E C0 11 00 00 ....^....
0x001B B2 B5 75 3F 05 12 FF FF ..u?.....

```

0/0 0/0 35

Implementing our very own gateway

Going further in Zigbee Light Link Standard

Cluster Id	Cluster Name
0x0000	Basic
0x0003	Identify
0x0004	Groups
0x0005	Scenes
0x0006	On/Off
0x0008	Level Control
0x0300	Color Control
0x1000	Commissioning

ZLL Clusters

Implementing our very own gateway

Going further in ZLL Commissioning Cluster

	Command identifier field value	Description	Mandatory/optional	Reference
Touchlink	0x00	Scan request	Mandatory	7.1.2.2.1
	0x02	Device information request	Mandatory	7.1.2.2.2
	0x06	Identify request	Mandatory	7.1.2.2.3
	0x07	Reset to factory new request	Mandatory	7.1.2.2.4
	0x10	Network start request	Mandatory	7.1.2.2.5
	0x12	Network join router request	Mandatory	7.1.2.2.6
	0x14	Network join end device request	Mandatory	7.1.2.2.7
	0x16	Network update request	Mandatory	7.1.2.2.8
	All other values in the range 0x00 – 0x3f	Reserved	-	-
Utility	0x41	Get group identifiers request	Mandatory ⁶	7.1.2.2.9
	0x42	Get endpoint list request	Mandatory ⁶	7.1.2.2.10
	All other values in the range 0x40 – 0xff	Reserved	-	-

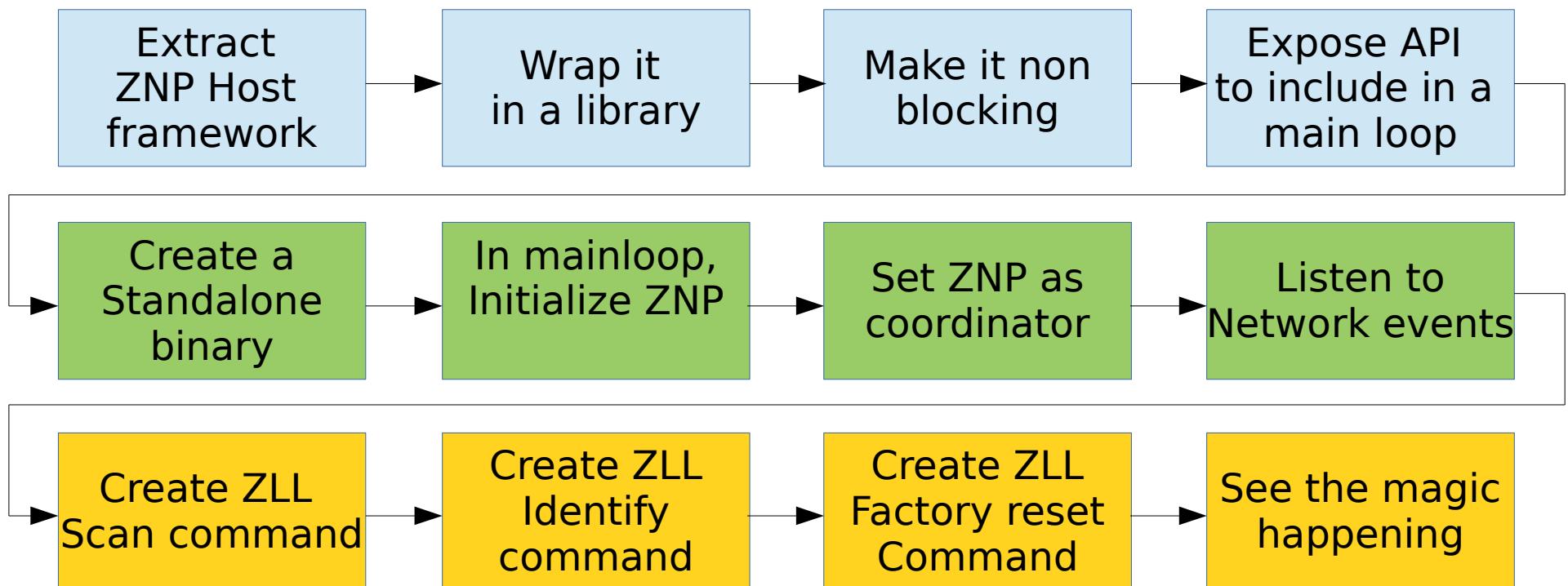
ZLL Commissioning cluster client commands

	Command identifier field value	Description	Mandatory/optional	Reference
Touchlink	0x01	Scan response	Mandatory	7.1.2.3.1
	0x03	Device information response	Mandatory	7.1.2.3.2
	0x11	Network start response	Mandatory	7.1.2.3.3
	0x13	Network join router response	Mandatory	7.1.2.3.4
	0x15	Network join end device response	Mandatory	7.1.2.3.5
Utility	All other values in the range 0x00 – 0x3f	Reserved	-	-
	0x40	Endpoint information	Mandatory ⁷	7.1.2.3.6
	0x41	Get group identifiers response	Mandatory ⁷	7.1.2.3.7
	0x42	Get endpoint list response	Mandatory ⁷	7.1.2.3.8
	All other values in the range 0x40 – 0xff	Reserved	-	-

ZLL Commissioning cluster server commands

Implementing our very own gateway

Let's start building serious things !



Implementing our very own gateway

The lamp has joined the network !

The screenshot shows two main windows from the Wireshark application:

- Traffic View:** A list of captured network frames. Frame 19 is selected, showing it as an "APS Transport Key" frame.
- Packet View - Packet #19:** A detailed view of the selected frame. The frame is identified as a "Beacon Request" (Frame ID 10). It includes fields for MAC Header, MAC Payload (containing NWK Header, NWK Payload, APS Header, APS Aux Header, APS Payload, and APS MIC), and MAC Footer.
- Hex View:** A hex dump of the selected frame's bytes. The bytes are shown in pairs, with some values highlighted in red or blue.
- Watch View:** A table for monitoring variables, currently showing "Field" and "Value" columns.

Implementing our very own gateway

Let's play with ou newly installed device

Identifier	Name
0x00	Move to hue
0x01	Move hue
0x02	Step hue
0x03	Move to saturation
0x04	Move saturation
0x05	Step saturation
0x06	Move to hue and saturation
0x07	Move to color
0x08	Move color
0x09	Step color
0x0a	Move to color temperature

Color control cluster commands

Command Identifier Field Value	Description	M/O
0x00	Off	M
0x01	On	M
0x02	Toggle	M
0x40	Off with effect	O
0x41	On with recall global scene	O
0x42	On with timed off	O

On/Off cluster commands

Octets	1	1
Data Type	uint8	uint8
Field Name	Effect identifier	Effect variant

« Off with effects » command parameters

Implementing our very own gateway

Playing with a new device : Xiaomi smart button

Some main differences :

- Sleeping node : only up when button state changes
- Hardcoded endpoints (new profile : Zigbee Home Automation)
- Difficult to read devices properties

Procedure :

- Hard reset button
- Wait for it to join network
- Push button : report button state visible
- On/Off cluster again !



Xiaomi Smart Button

Implementing our very own gateway

Playing with a new device : Xiaomi room sensor

Procedure :

- Hard reset button
- Wait for it to join network
- Changes on temperature : new temperature report
- New cluster : temperature measurement
- New command : report attribute



Xiaomi room sensor

- Zigbee : what is it ?
- Zigbee fundamentals
- Our case : the « bulb issue »
- Implementing our very own gateway : steps and feedback
- **Next steps and improvements**

Next steps and improvements

Current status

The screenshot shows the GitHub repository page for `Tropicao/zll-gateway`. The repository is described as a "Zigbee gateway implementation". It has 148 commits, 5 branches, 2 releases, and 1 contributor. The license is GPL-3.0. The last commit was made a day ago. The repository has 2 stars and 0 forks.

File	Description	Time Ago
<code>config</code>	Rename sample configuration file to allow syntaxic coloration	17 days ago
<code>doc</code>	Add device discovery state machine doc	5 days ago
<code>scripts</code>	Add utils script	23 days ago
<code>src</code>	Add device id saving/restore	3 days ago
<code>.gitignore</code>	Add basic device persistent registration	15 days ago
<code>LICENSE</code>	Initial commit	2 months ago
<code>README.md</code>	Merge branch 'master' of <code>github.com:Tropicao/zll-gateway</code>	a day ago
<code>README_demo.md</code>	Update demo README	4 days ago
<code>TODO.md</code>	Update TODO	18 days ago

`git@github.com:Tropicao/zll-gateway.git`

Next steps and improvements

Some major features are still missing...

- Multi-channel management
- Groups management
- ZLL : proper device install (i.e. no « raw » reset for install)
- Generic device management
- Remote devices configuration (e.g. : reporting frequency)

Next steps and improvements

... and a lot of products are to be supported !



*osram-lamps.fr
domo-attitude.fr
high5gadgets.ie*



alexis.lothore@gmail.com



<https://github.com/Tropicao>