



life.augmented

Welcome to **STM32 Zigbee workshop**

Workshop team





Introduction

- The purpose of this hands-on session is to demonstrate how to easily start the evaluation and development of Zigbee application on STM32WBx devices and ecosystem
- We will use two ST evaluation kits during this workshop:
 - NUCLEO-WB55RG
 - NUCLEO-WBA55CG



Agenda (9:00 – 12:00)

1

30 min

“Hello World” application – On/Off
Code and architecture walkthrough

5

30 min

Cluster management
Change On/Off to Heater/Thermostat

2

30 min

Extended On/Off example
Light bulb / Light Switch

6

15 min

Large Zigbee network

3

15 min

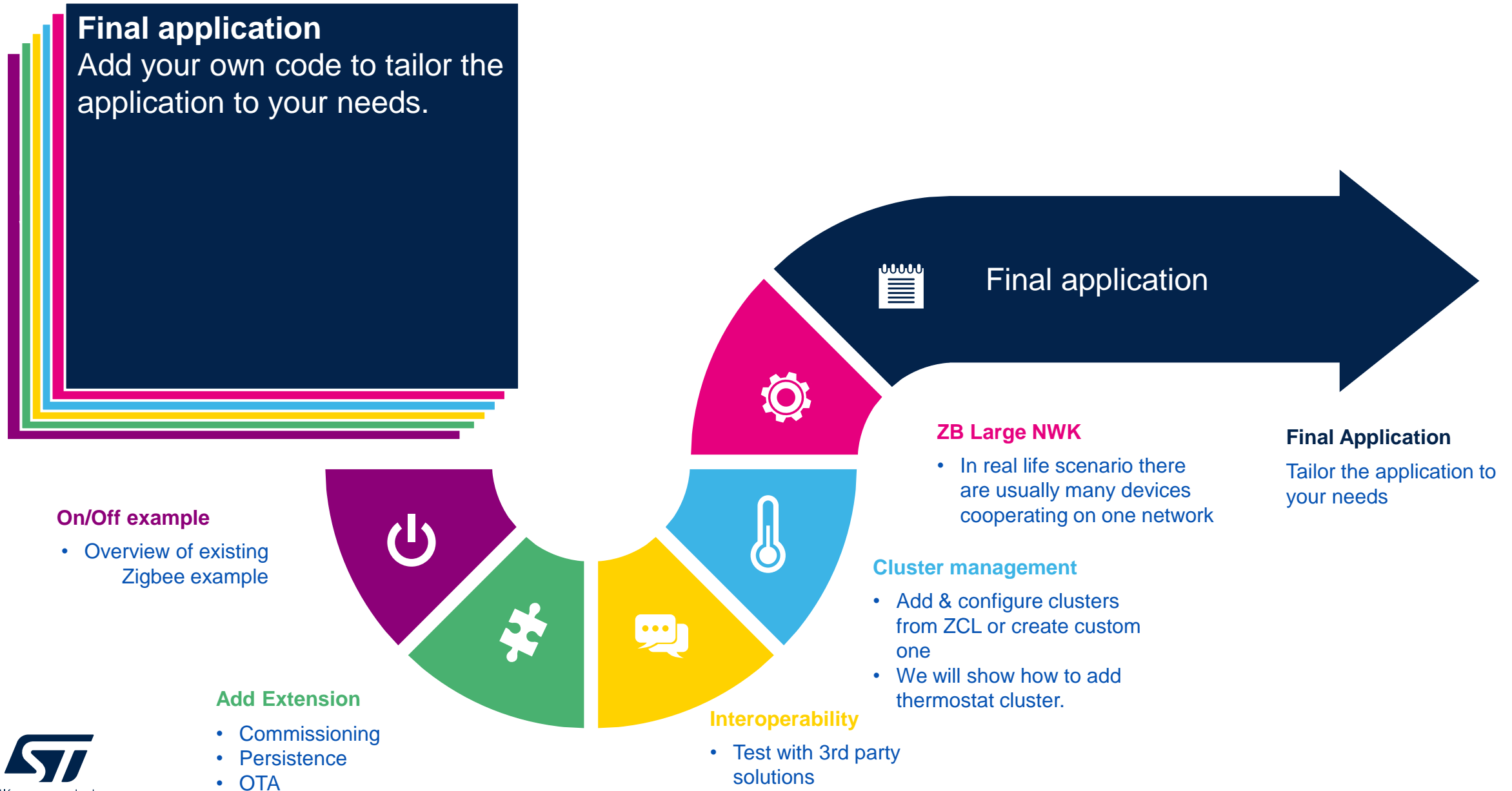
Test inter-operability
With 3rd party coordinators

4

15 min

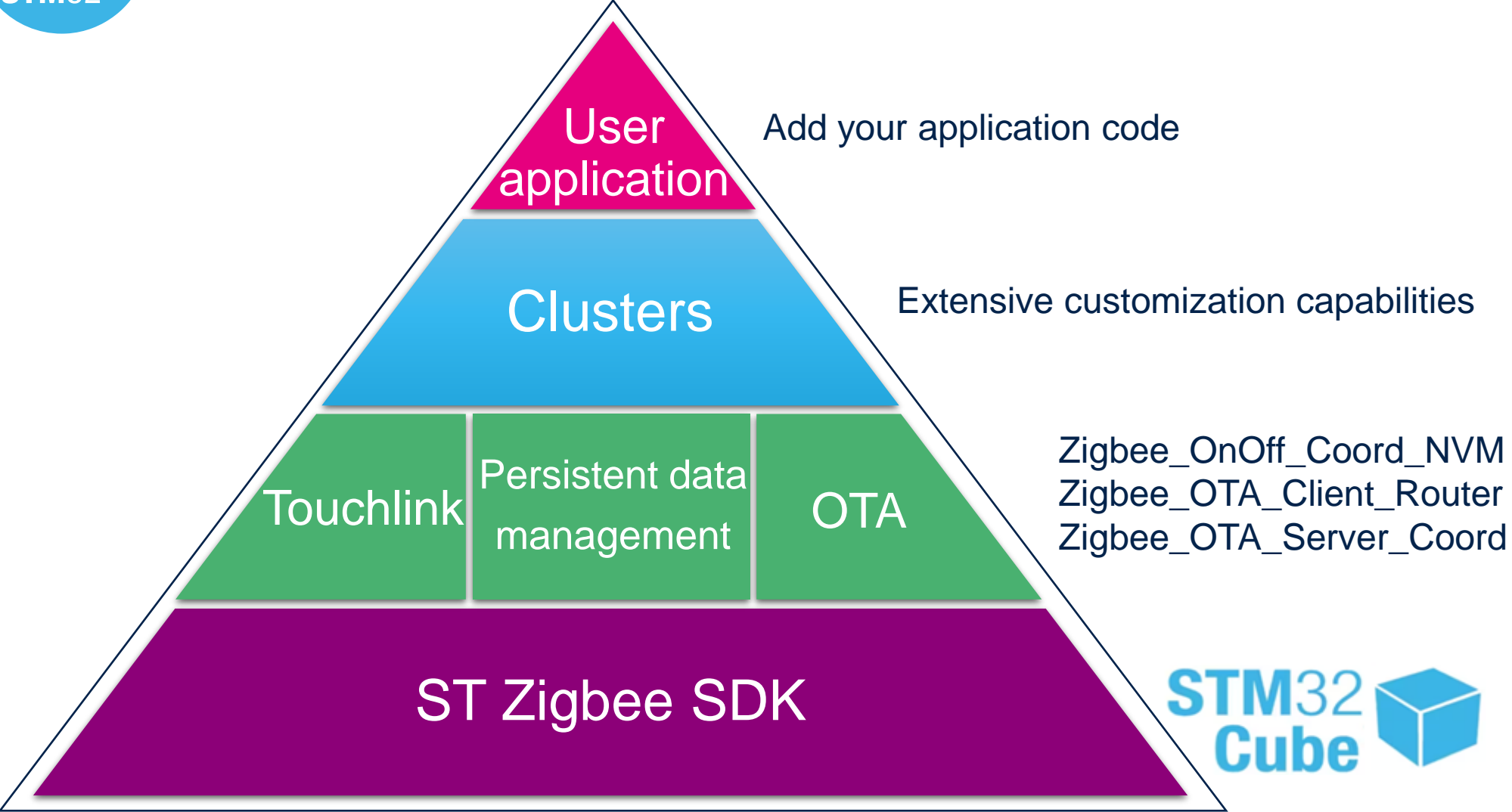
Break

Let's start your Zigbee journey with STM32WBx today!





Zigbee project pyramid



“Hello World” application – On/Off

Code and architecture walkthrough

Let's start your Zigbee journey with STM32WBx today!

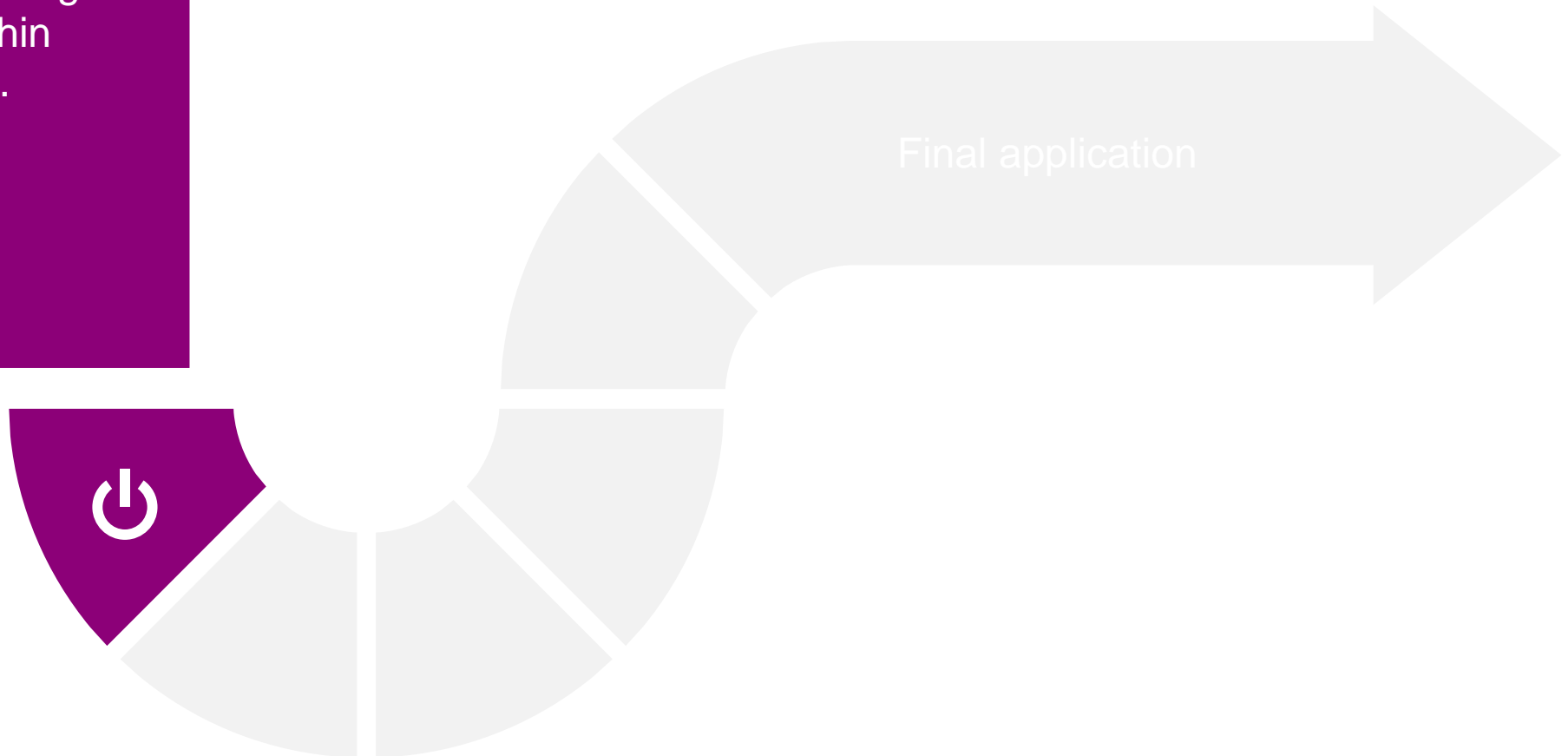
Part 1: "Hello World" application – On/Off

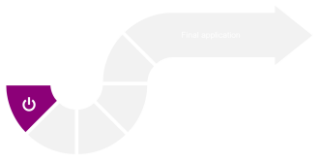
On/Off example

Start with ready to use Zigbee example delivered within STM32Cube package.

On/Off example

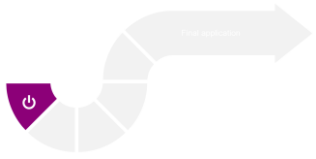
- Overview of existing Zigbee example





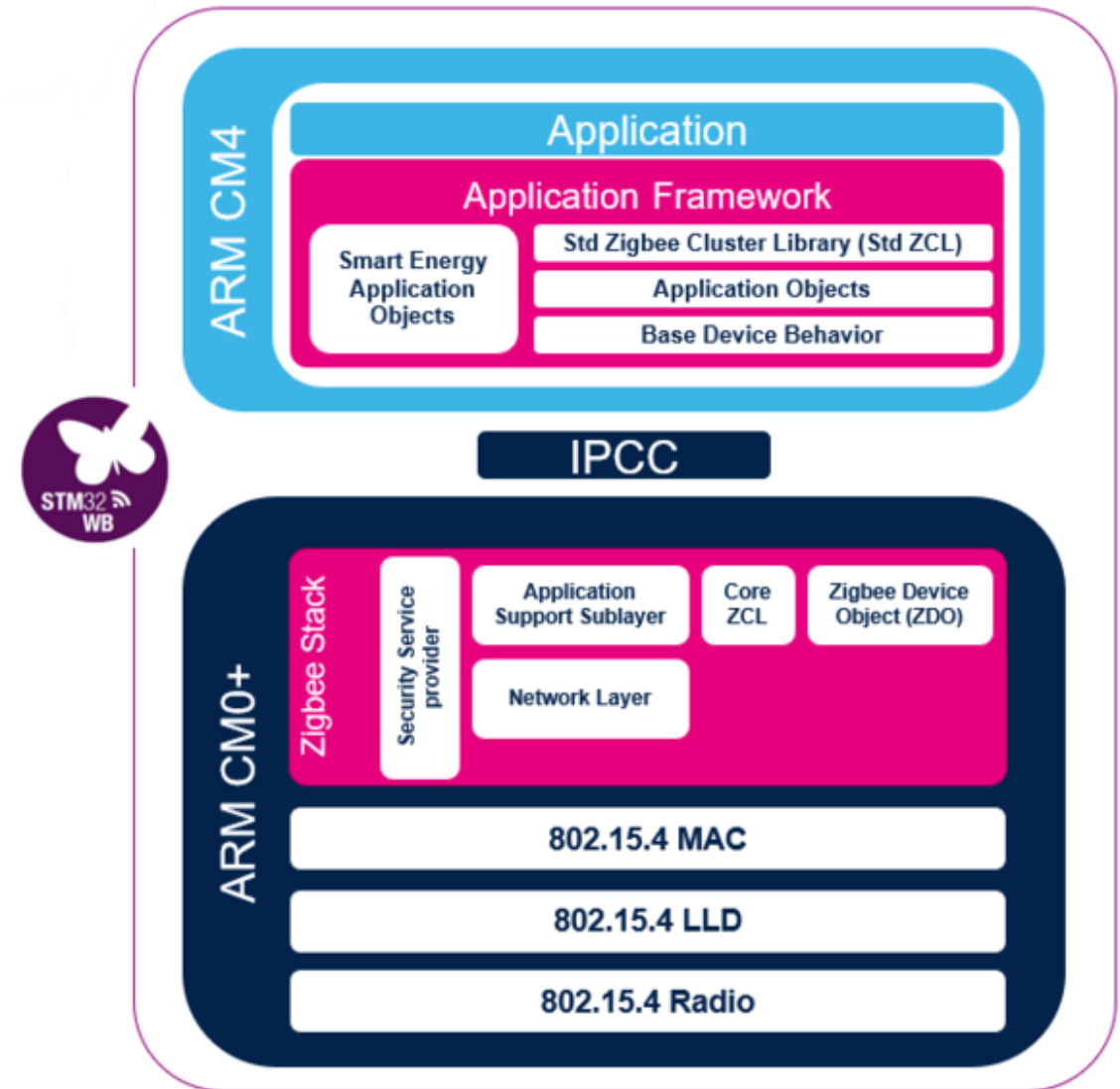
Purpose

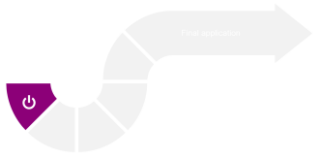
- In this part we will show following:
 - Explanation on architecture of zigbee solution on STM32 MCUs.
 - Step by step how to build and use basic example with On/Off cluster
 - How zigbee coordinator forms network and how other devices join it
 - Zigbee application callbacks
 - Debugging options and configuration of the application



Zigbee architecture overview on STM32WB

- User application runs on M4
- To run Zigbee application on STM32WB it is necessary to load stack using FUS
- Zigbee stack is then running on M0

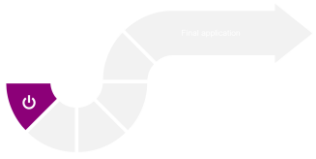




Available zigbee stack

- Two types of the stack supported
- Both stacks are Zigbee PRO 2017 (revision 22) Compliant Platform certified
- Zigbee FFD (Full feature device)
 - can accept any role in the network. It can be a router, a coordinator or an end device.
- Zigbee RFD (Reduced feature device)
 - An RFD can support only router or end device role.
 - Smaller footprint compared to an FFD.

Stacks supported	Firmware associated
Zigbee FFD (Full feature device)	stm32wb5x_ZigBee_FFD_Full_fw.bin
Zigbee RFD (Reduced feature device)	stm32wb5x_ZigBee_RFD_fw.bin



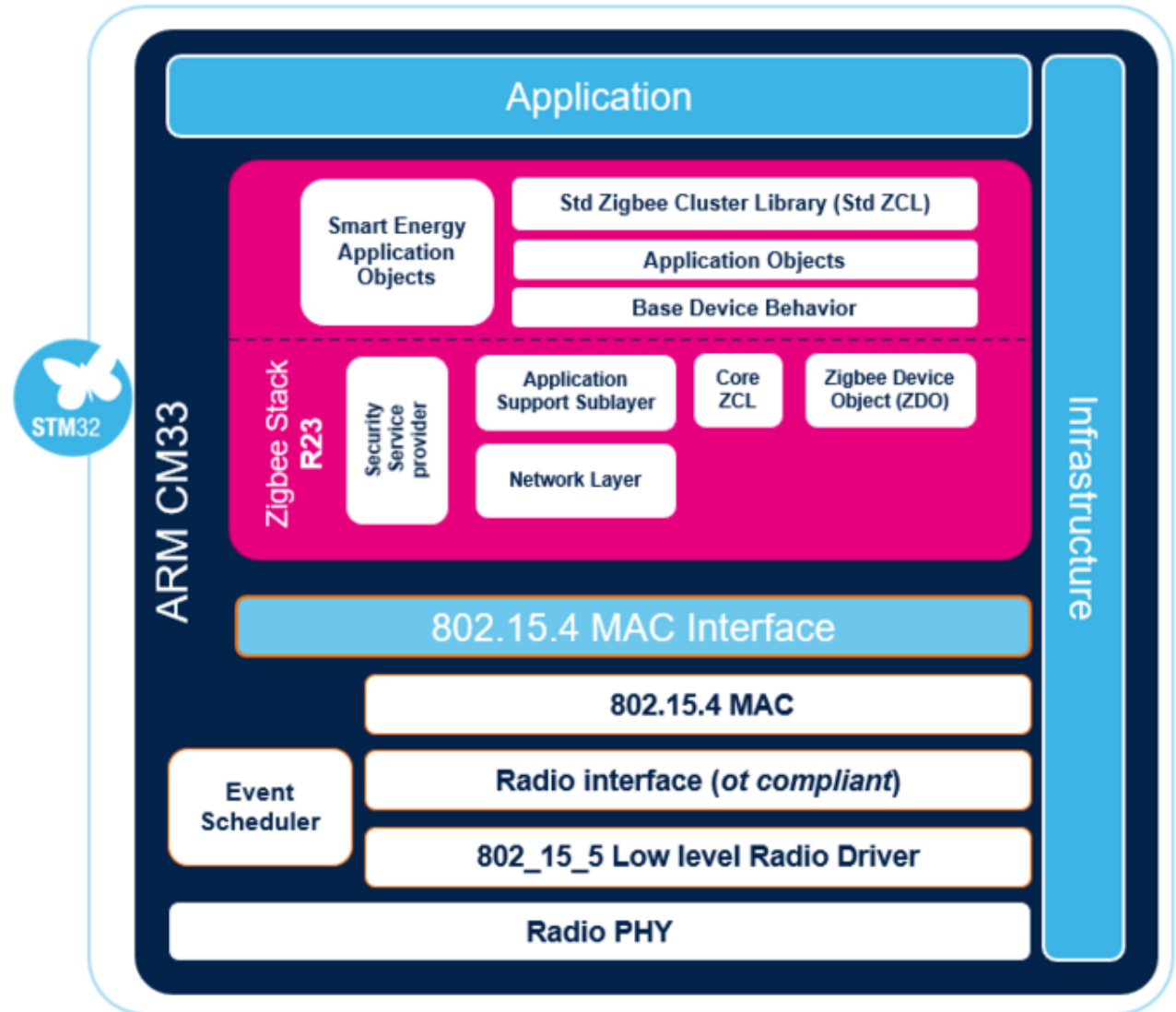
FUS – Firmware Upgrade Service

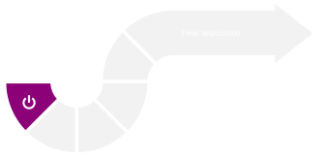
- A piece of FW placed in secured part of memory.
- Responsible for updating either:
 1. FUS itself
 2. A wireless firmware stack (ZigBee, BLE, Thread ...)
- Authenticates, decrypts and installs the downloaded image
- Dedicated Wiki page [here](#)

The screenshot shows the STM32WB Firmware Upgrade Service Wiki page. The sidebar on the left contains a search bar and a list of navigation links under the 'STM32 MCU' header. The main content area has a top navigation bar with 'Welcome', 'Microcontroller', 'Solutions', and 'Software development kit'. Below this is the title 'STM32WB Firmware Upgrade Service'. The main content area displays a 'Contents' section with a list of links for each chapter, including 'What is firmware upgrade services (FUS)?', 'How FUS works', 'FUS versioning and identification', 'FUS activation and upgrade', and 'FAQ'.

Zigbee architecture overview on STM32WBA

- Single core solution
- Supports the Zigbee stack R23, which is also available in two types FFD and RFD.
- Provided as secure library

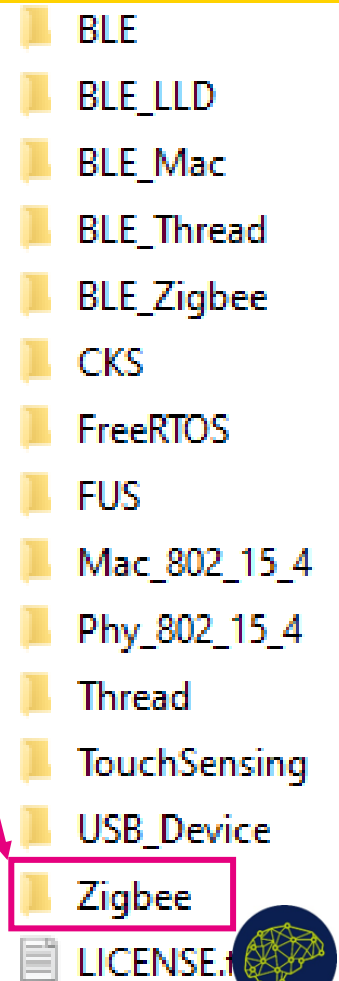
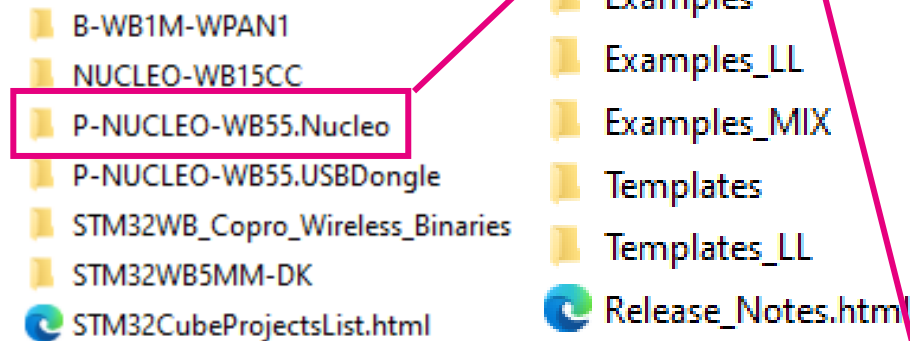


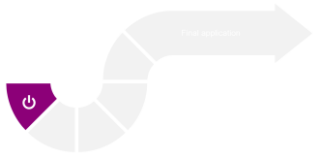


Application example

ST Zigbee project example directory

- Zigbee applications are delivered as source codes for different HW platforms.
- The purpose of these applications is mainly to provide simple examples that highlight the use of specific clusters.
- We will be showing
 - Zigbee_OnOff_Server_Coord_FreeRTOS
 - Zigbee_OnOff_Client_Router_FreeRTOS
- Located in
 - C:\Users\username\STM32Cube\Repository\STM32Cube_FW_WB_V1.19.1\Projects\P-NUCLEO-WB55.Nucleo\Applications\Zigbee folder

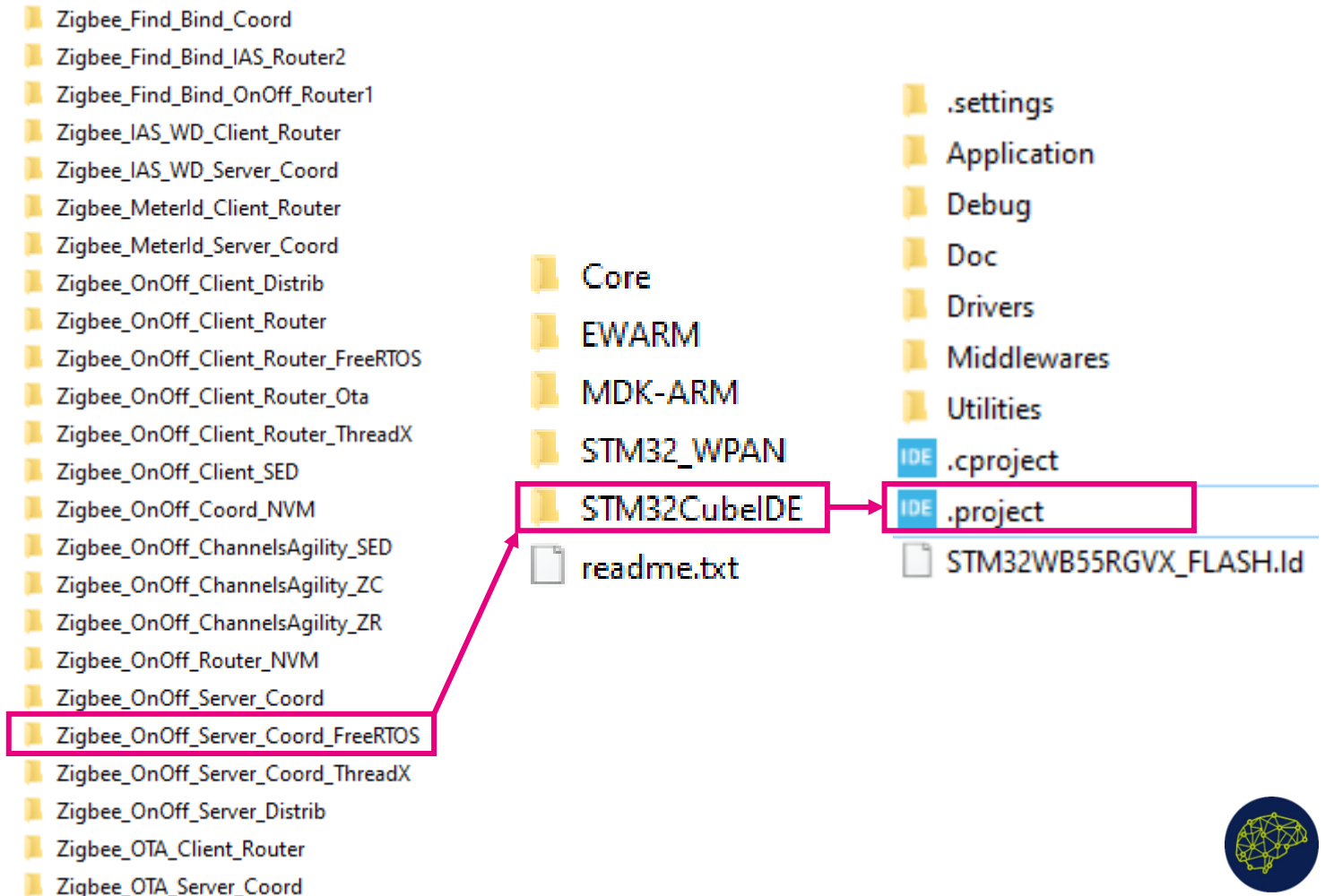


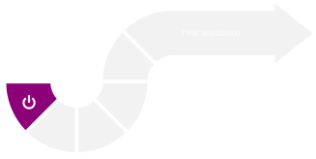


Application example

STM32WB Build Zigbee project

- Open the STM32CubeIDE dedicated directory.
- select the .project of the demonstration
- Launch the STM32CubeIDE

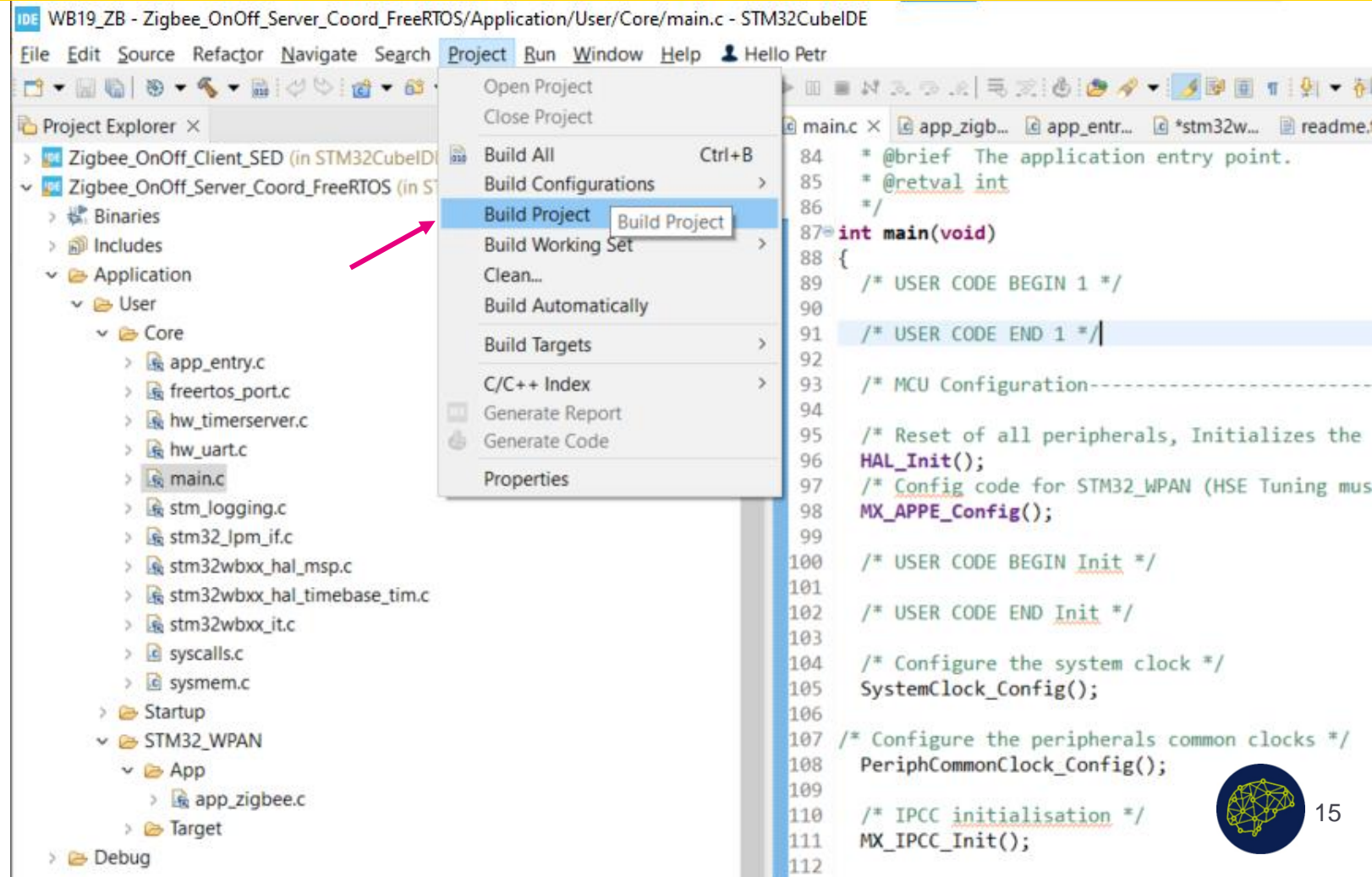


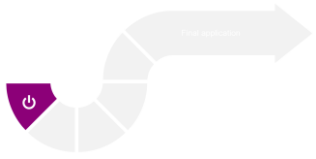


Application example

STM32WB Build Zigbee project

- Ensure that your project is correctly visible into the project explorer view.
- Build your project and run it





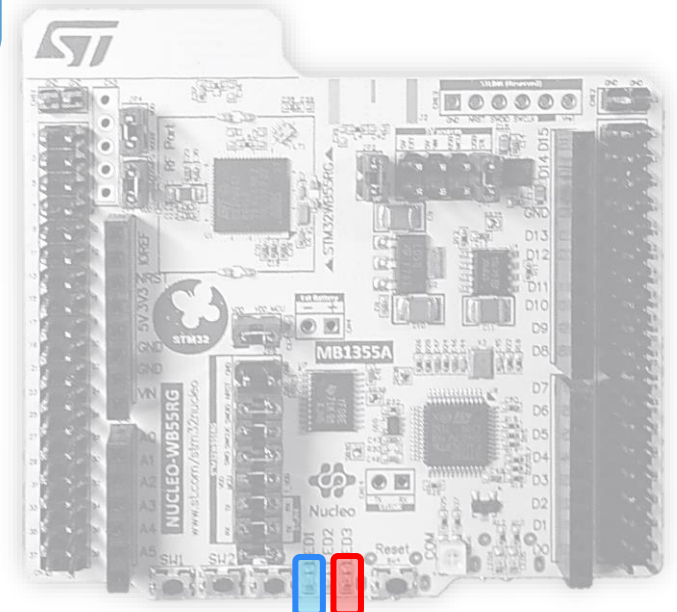
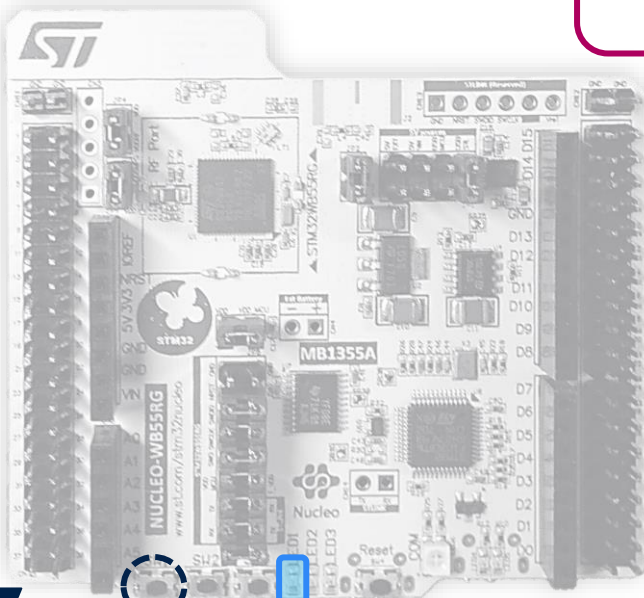
Example explanation

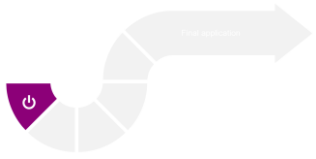
Sending Toggle commands using OnOff Cluster

- Once the Zigbee mesh network is created (**LED** is on), the user can send requests from the client to the server through the push button -> Toggle the **LED**

On/Off
Client Router
(Joining node)

On/Off
Server Coordinator
(Trust Center)





Network joining

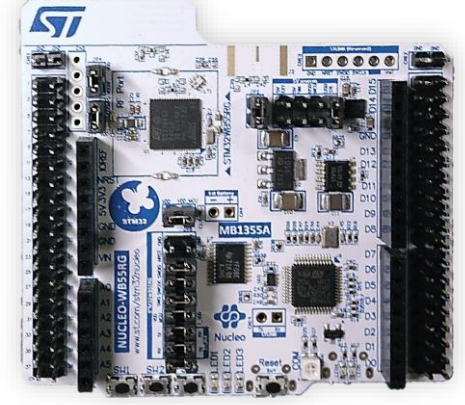
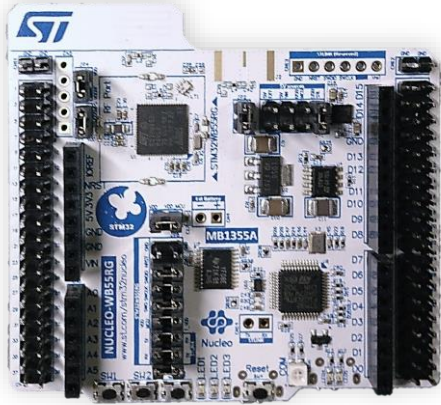
Example explanation

Nucleo-WB55RG

On/Off
Client Router
(Joining node)

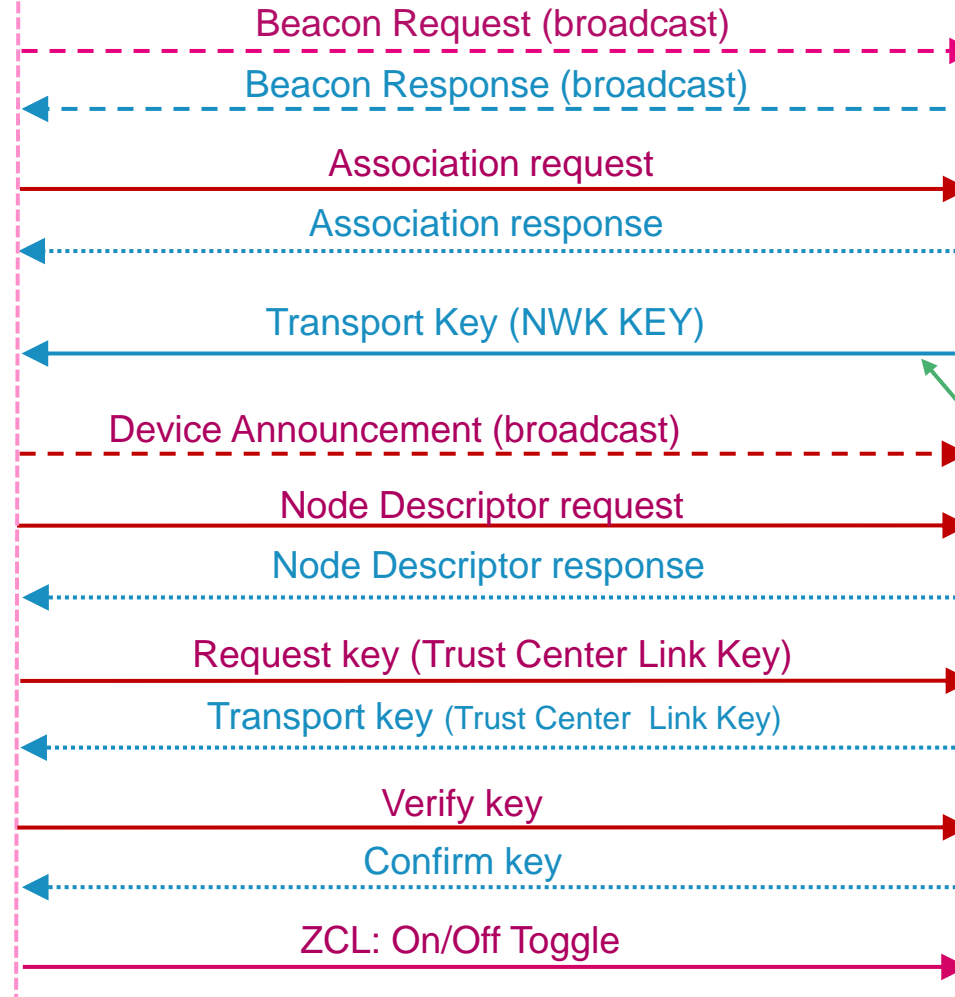
On/Off
Server Coordinator
(Trust Center)

Nucleo-WB55RG



broadcast
direct addressing
response

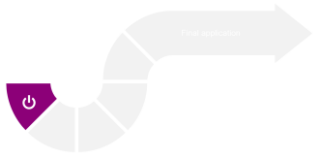
Traffic encrypted with NWK key



Encrypted with well known
TCLK alias *ZigBeeAlliance09*

TCLK update is mandatory
as per R21+ spec





Network joining

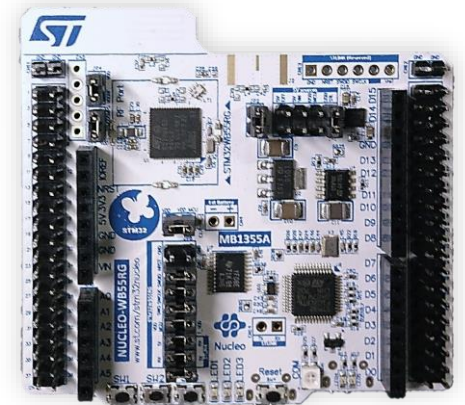
Example explanation

Nucleo-WBA55CG

On/Off
Client Router
(Joining node)

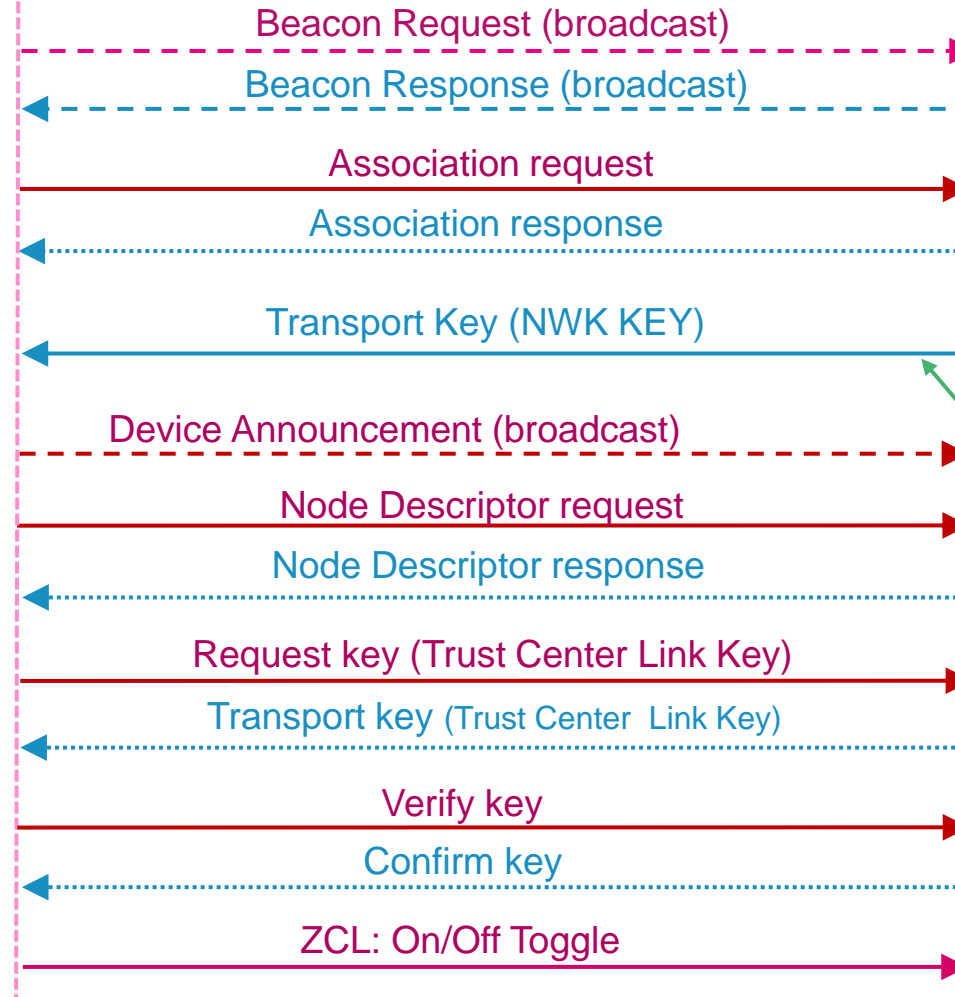
On/Off
Server Coordinator
(Trust Center)

Nucleo-WB55RG



broadcast
direct addressing
response

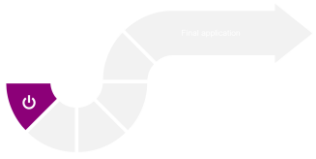
Traffic encrypted with NWK key



Encrypted with well known
TCLK alias *ZigBeeAlliance09*

TCLK update is mandatory
as per R21+ spec





Hands-on demo On/Off

Use serial terminal to see application's reports from both boards

Coordinator

Router

Attempting network discovery.

Associating (src = 0x0080e125011c916d)

Adding network key, sequence number = 0
Sending Device_Annce for 0x6cb3

NLME-NETWORK-FORMATION.request successful

nwk_handle_assoc : Association accepted,

115200 bps
8 data bits
1 stop bit

115200 bps
8 data bits
1 stop bit

Extended On/Off example

Light bulb / Light Switch

Let's start your Zigbee journey with STM32WBx today!

Part 2: Extended On/Off example

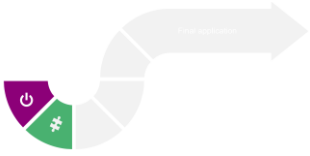
Add Extension

- Commissioning
- Persistence data management
- OTA



Add Extension

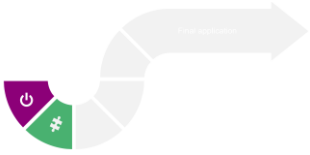
- Commissioning
- Persistence
- OTA



Purpose

- This multi-steps hands-on will improve On/Off example with the features which are needed for real Zigbee product.
- Features:
 - ZLL commissioning (Touchlink)
 - Persistence data management
 - OTA functionality





Key features

- **Commissioning**

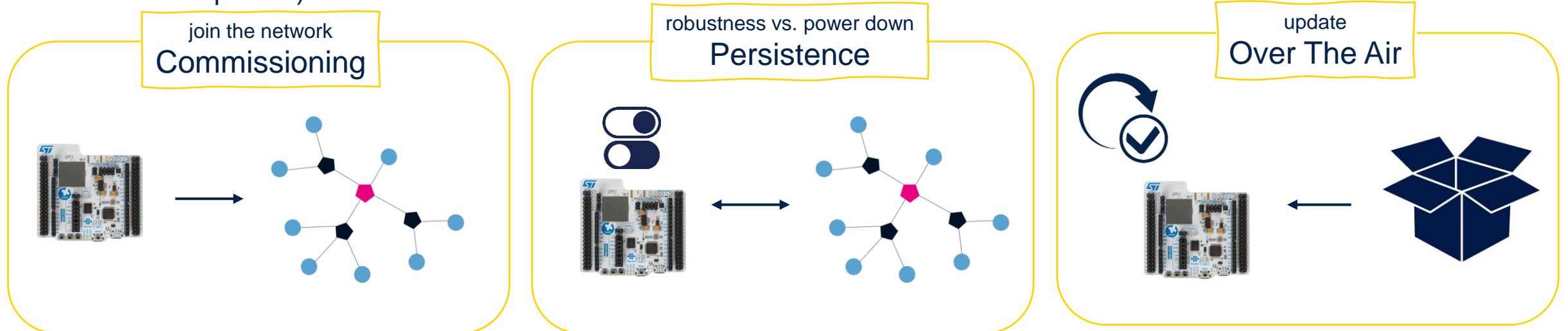
- For every Zigbee end-device, it usually requested to be able join Zigbee networks created and managed by devices produced by 3rd parties

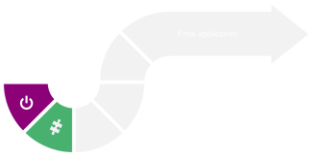
- **Persistence data management**

- When the device joins the network, it must retain the network data (robustness vs. power down), so there is no need to perform the commissioning again after reset

- **OTA**

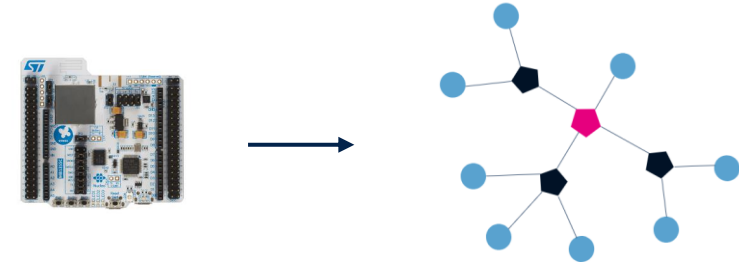
- Usually, it is important to provide the possibility to be able to deploy FW update over the time (bug fix, vulnerability fix, features updates)



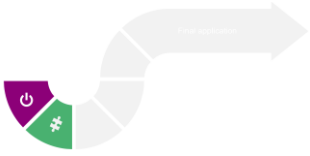


Commissioning

- Commissioning is a process that allows a new Zigbee device to join a Zigbee network.
- Network commissioning covers the following areas:
 - Creating a network
 - Allowing devices to join a network
 - Joining a network
 - Binding a local endpoint to an endpoint on a remote node
 - Adding a remote node to a group

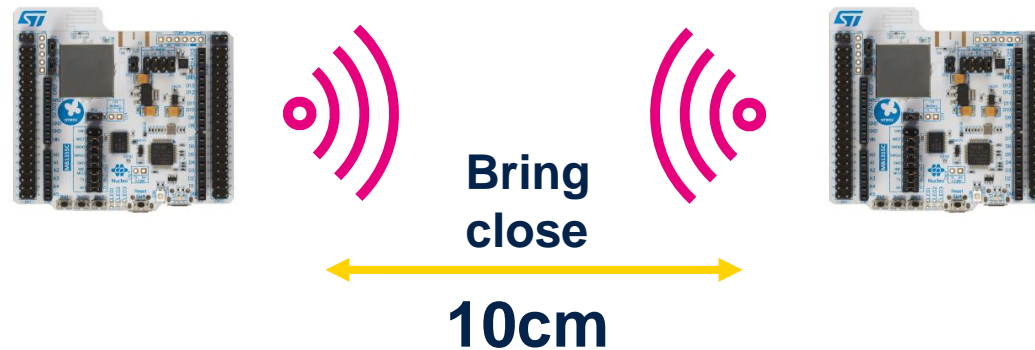


Commissioning mode	Functionality
Touchlink	Creating a new network Allowing other devices to join an existing network Joining local device to an existing network
Network Steering	Allowing other devices to join an existing network Joining local device to an existing network
Network formation	Creating a new network
Finding and binding	Adding a remote node to a group

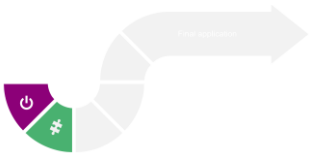


Touchlink commissioning

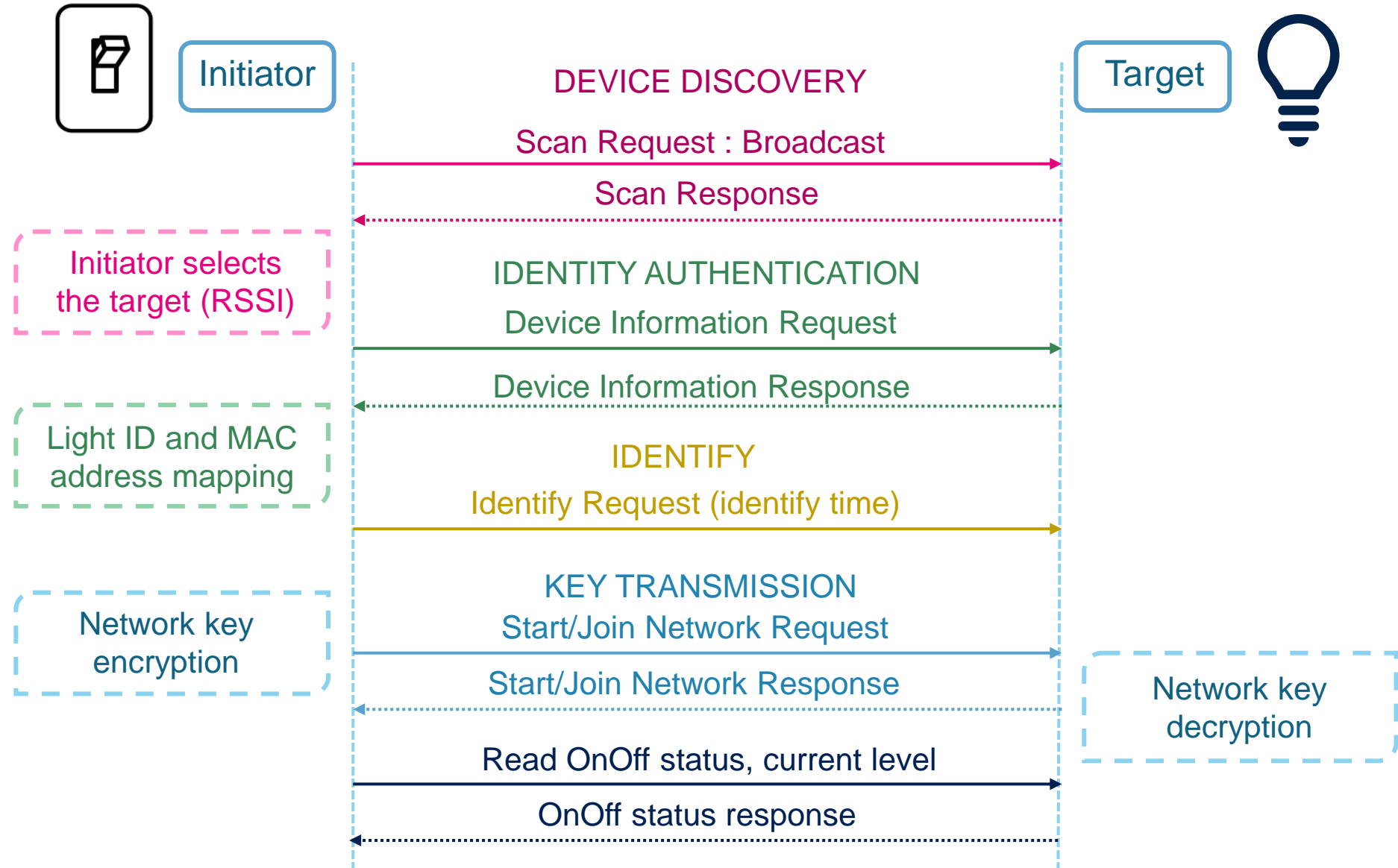
- Touchlink is a form of commissioning mechanism where nodes only join when devices are within close proximity
- The commissioning messages are initially sent through unencrypted InterPAN

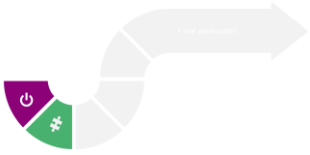


- [Application note AN5227](#)



Touchlink commissioning protocol





Hands-on demo Touchlink

Use serial terminal to see application's reports from both boards

Initiator

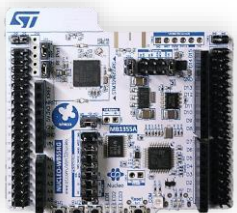
Target

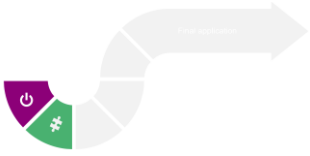
touchlink_start : Setting macPanId = 0x0001, macSh

touchlink_target_step22 : Touchlink Target - Step 22
ZbZdoDeviceAnnce : Sending Device_Annce for 0x0002
touchlink_done : Touchlink completed in 4 seconds

115200 bps
8 data bits
1 stop bit

115200 bps
8 data bits
1 stop bit

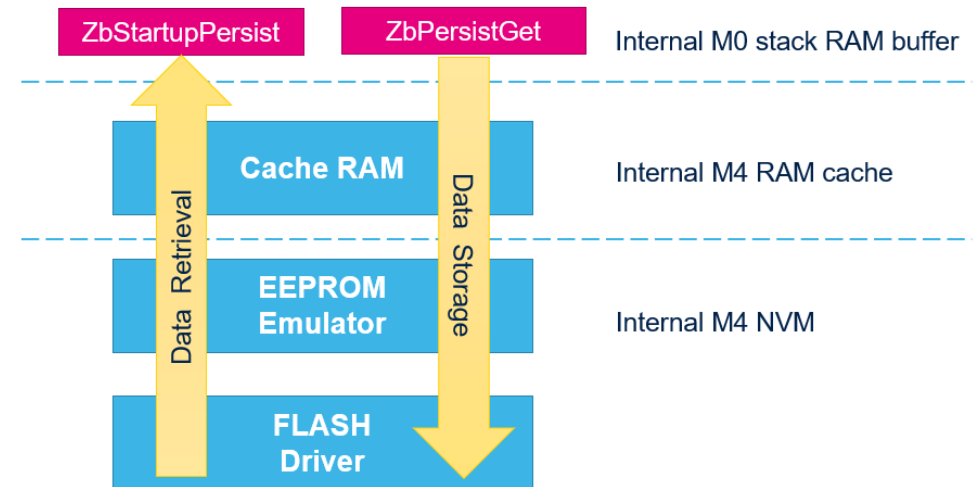


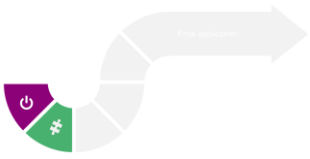


Zigbee Persistence data management

- Zigbee network has set of important information affecting the functioning & operating of the device in the network
- To keep the device operating even after reset (power shutdown, system reset...), it is necessary to move the data from RAM to FLASH (NVM) memory
- This is called **Persistent data management**
 1. Stack notifies the application each time the data change
 2. Application store the data to FLASH using EEPROM EMULATION

This implementation ensure that the device is still up to date and can recover from unexpected reset.



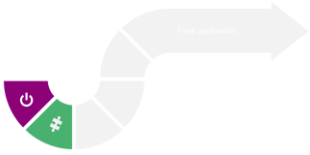


Persistence data – content

- The persistence data which are stored to NVM memory
 - Device behavior information (device role, ...)
 - Network information (channel, PANID, ...)
 - Neighbour and Routing tables
 - Keys (network key table, link key, Touchlink key)
 - Address mapping
 - Biding table
 - Zigbee cluster library

More details are available in [Application note AN5492](#)

Name	Definition	Name	Definition
Persistence table version	Persistence table version	APS channel mask	Application support sublayer channel mask
bdb_ib	Base device behavior information base	APS pre-conf link key	Application support sublayer pre-confirmation link key
aps_ib	Application support Sublayer Information Base	APS binding table	Application support sublayer binding table
nwk_ib	Network information base	APS group table	Application support sublayer group table
bdb_tl_key	Touchlink key	APS link key table	Application support sublayer link key table
address_map	Address map	MAC channels	Media access control channels
NNT	Network neighbour table	MAC power Table	Media access control power table
NRT	Network routing table	EUI address	Extended unique identifier address
NWK RREC	Table of outstanding route requests	ZCL persist server	Zigbee cluster library persist server
NWK key table	Network key table		



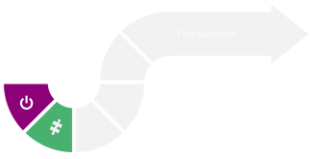
Zigbee Persistence data management

- **Stack notification** – The Zigbee stack notifies the application each time the persistent data change

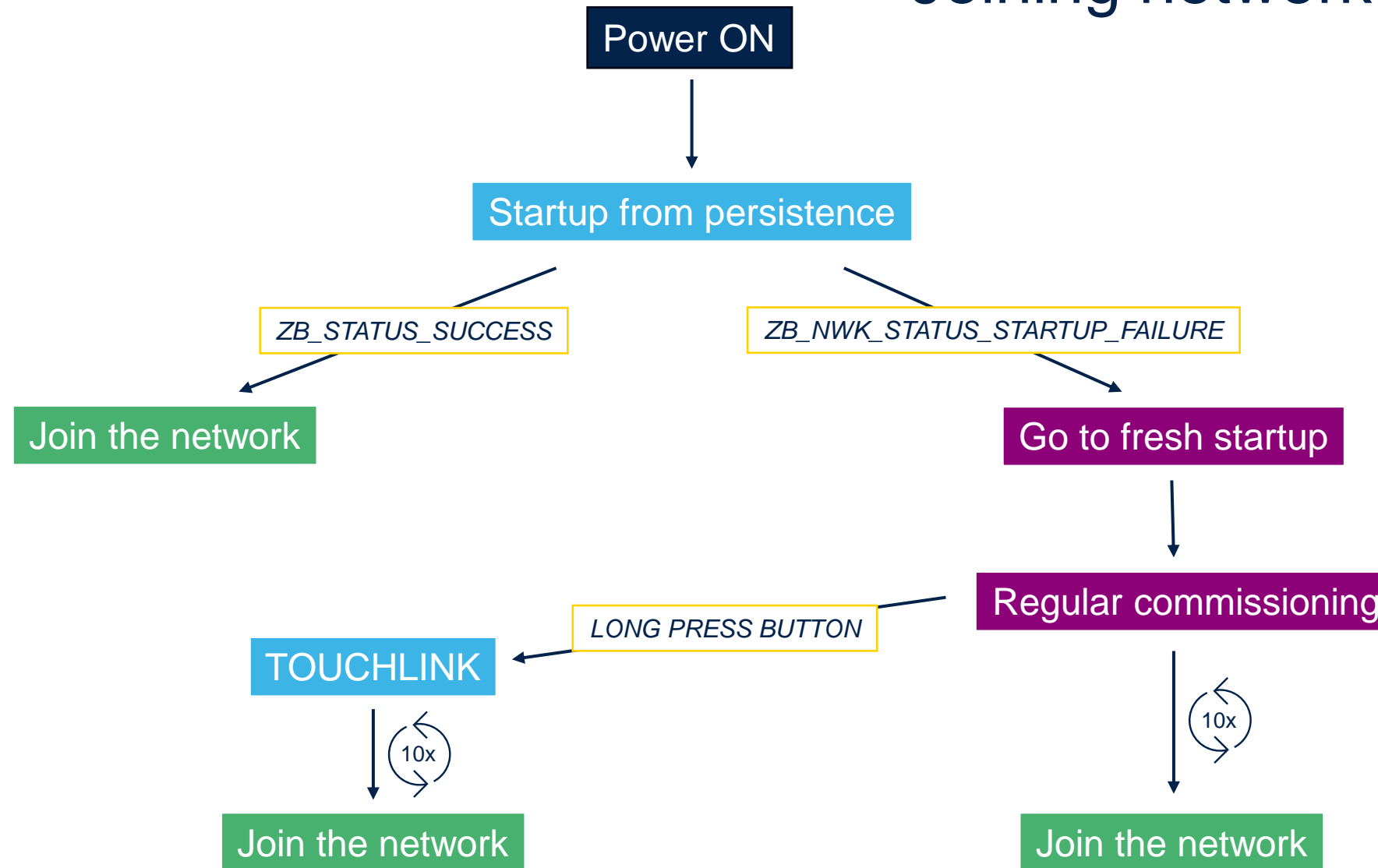
```
/* Register persistent data change notification */  
ZbPersistNotifyRegister(zigbee_app_info.zb, APP_ZIGBEE_persist_notify_cb, );
```

```
static void APP_ZIGBEE_persist_notify_cb(struct ZigbeeT * zb, void *cbarg)  
{  
    APP_DBG("Notification to save persistent data requested from stack");  
    /* Save the persistent data */  
    APP_ZIGBEE_persist_save();  
}
```

- The **ZbPersistGet API** allows the application to get data and its length from an internal stack buffer in the RAM. They are then copied into a RAM cache and written in the flash memory through the EEPROM emulator.
- The **ZbStartupPersist API** allows the Zigbee stack to initialize/start, using persistent data from the buffer (previously copied from the flash memory through the EEPROM emulator).

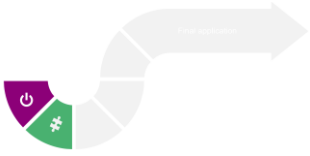


Joining network - flowchart



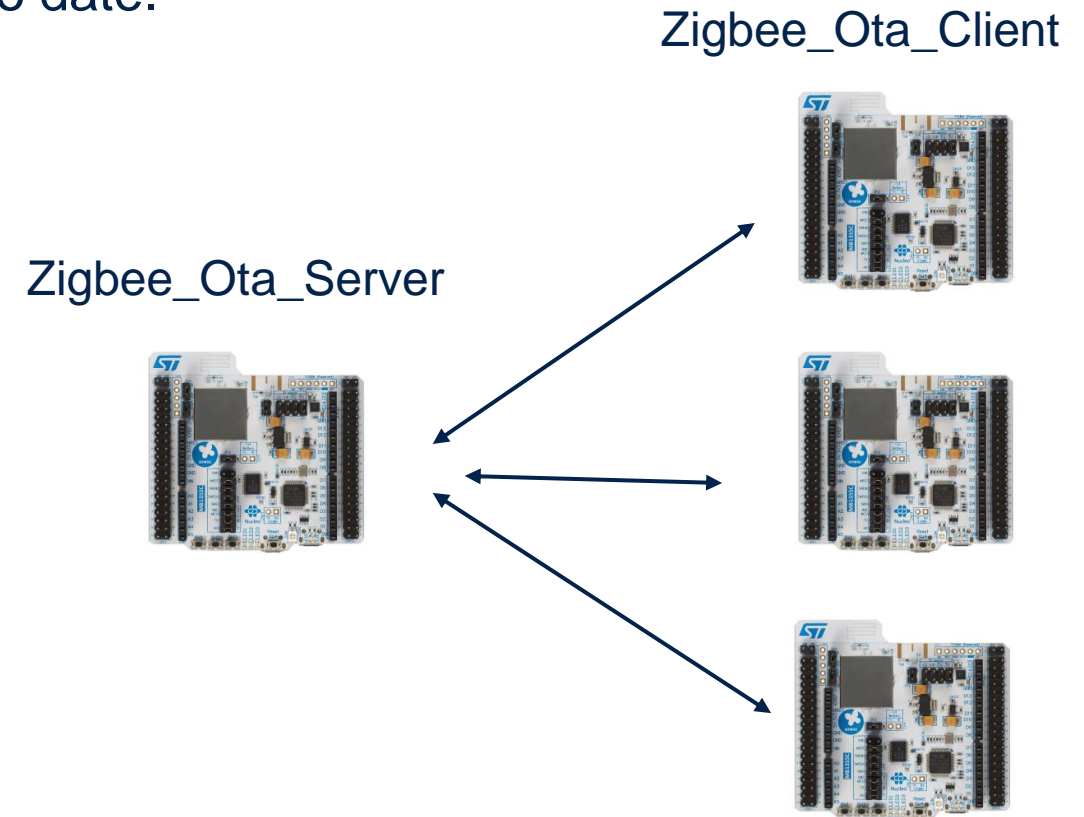
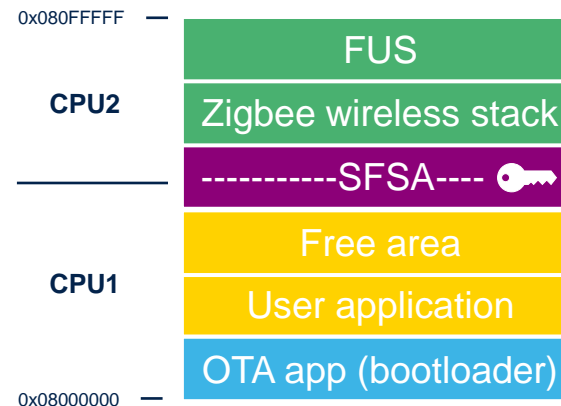


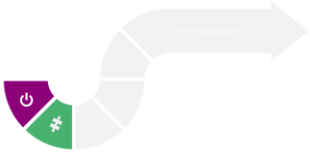
32



STM32WB Zigbee OTA

- The OTA is a feature that allows the update of Zigbee devices Over-The-Air
- It is a key feature to keep your devices, in the field, up to date.
- OTA Server/Client model
 - Server is sending new binary to the clients
- The application should integrate
 1. Flash storage – part of memory where new bin is stored
 2. Bootloader – it can install the binaries and boot from new FW
 3. Zigbee OTA protocol





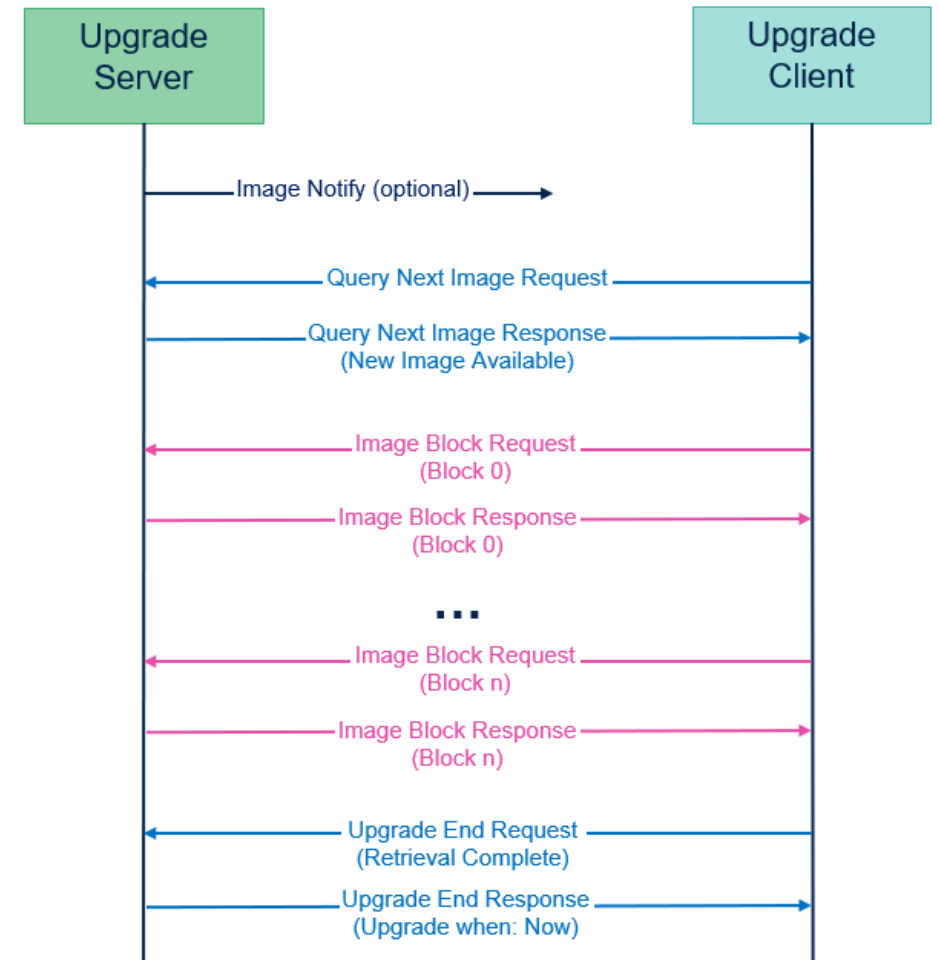
STM32WB Zigbee OTA

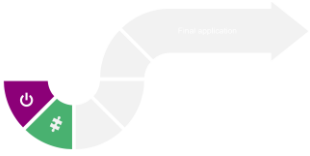
- Zigbee OTA protocol - Upgrade Diagram

1. Server notifies the client about new FW (optional)
2. Client ask Server about availability of new FW
3. If available, Client request Image Block
4. Once finished, client end up the process with Upgrade End Request

- List of APIs

ZbZclOtaClientAlloc	Create new OTA client cluster
ZbZclOtaServerAlloc	Create new OTA server cluster
ZbZclOtaClientDiscover	Discover OTA server
ZbZclOtaClientDiscoverForced	Set the OTA server directly
ZbZclOtaClientImageTransferResume	Resume an OTA upgrade transfer
ZbZclOtaClientImageTransferStart	Initiate an OTA transfer





Hands-on demo OTA

Use serial terminal to see application's reports from both boards

OTA Server with Short Address 0x0000.
OTA Server init done!

Query Next Image request received.
Sender device manufacturer code: 0x1041.
Sender device current file version: 0x00000001.
Sender device current hardware version: 0x0001.
Sender device is requesting an upgrade for 0x0002 image.
A such image is available.

[OTA] Upgrade End request received.
UpgradeEnd status SUCCESS, responding with:
Upgrade time: 12
- Average throughput = 5.76 kbit/s.

Server

Client

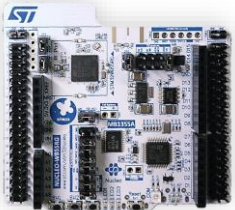
OTA Client with Short Address 0x809E.
OTA Client init done!

Sending Query Next Image request.
Client Query Next Image request response
A such image is available.

Starting download.

FUOTA Transfer (current_offset = 0x0429)
FUOTA Transfer (current_offset = 0x082E)

FUOTA_CLIENT : END OF TRANSFER COMPLETED
--> Request to reboot on FW Application



115200 bps
8 data bits
1 stop bit



115200 bps
8 data bits
1 stop bit

Inter-operability tests

With 3rd party coordinators

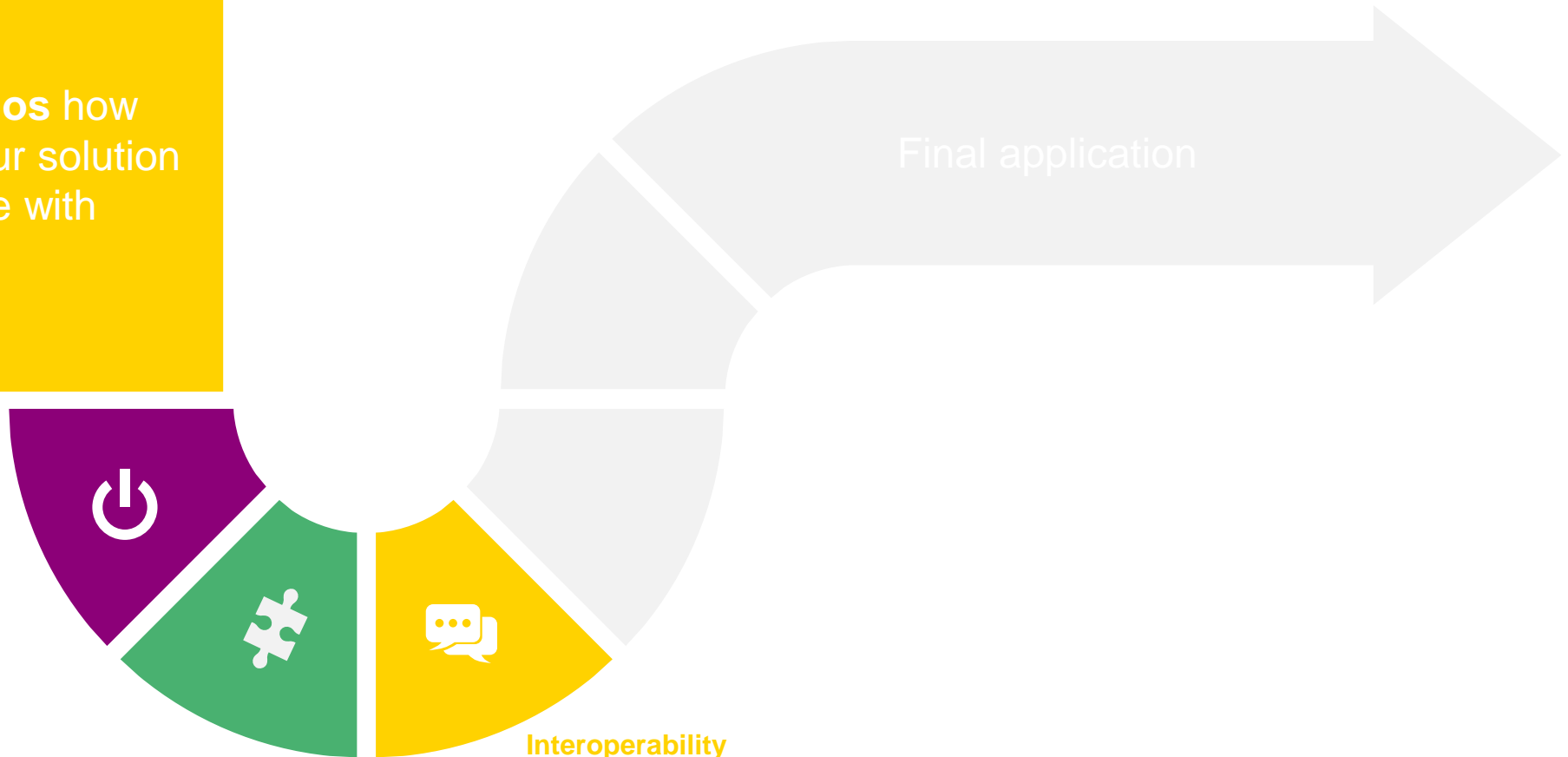
Let's start your Zigbee journey with STM32WBx today!

Part 3: Inter-operability tests

Interoperability

Test with on-the-shelf Zigbee products & 3rd party ecosystem.

We will show **3 demos** how devices based on our solution can easily cooperate with known solutions





Direct control of commercial zigbee devices supporting Touchlink

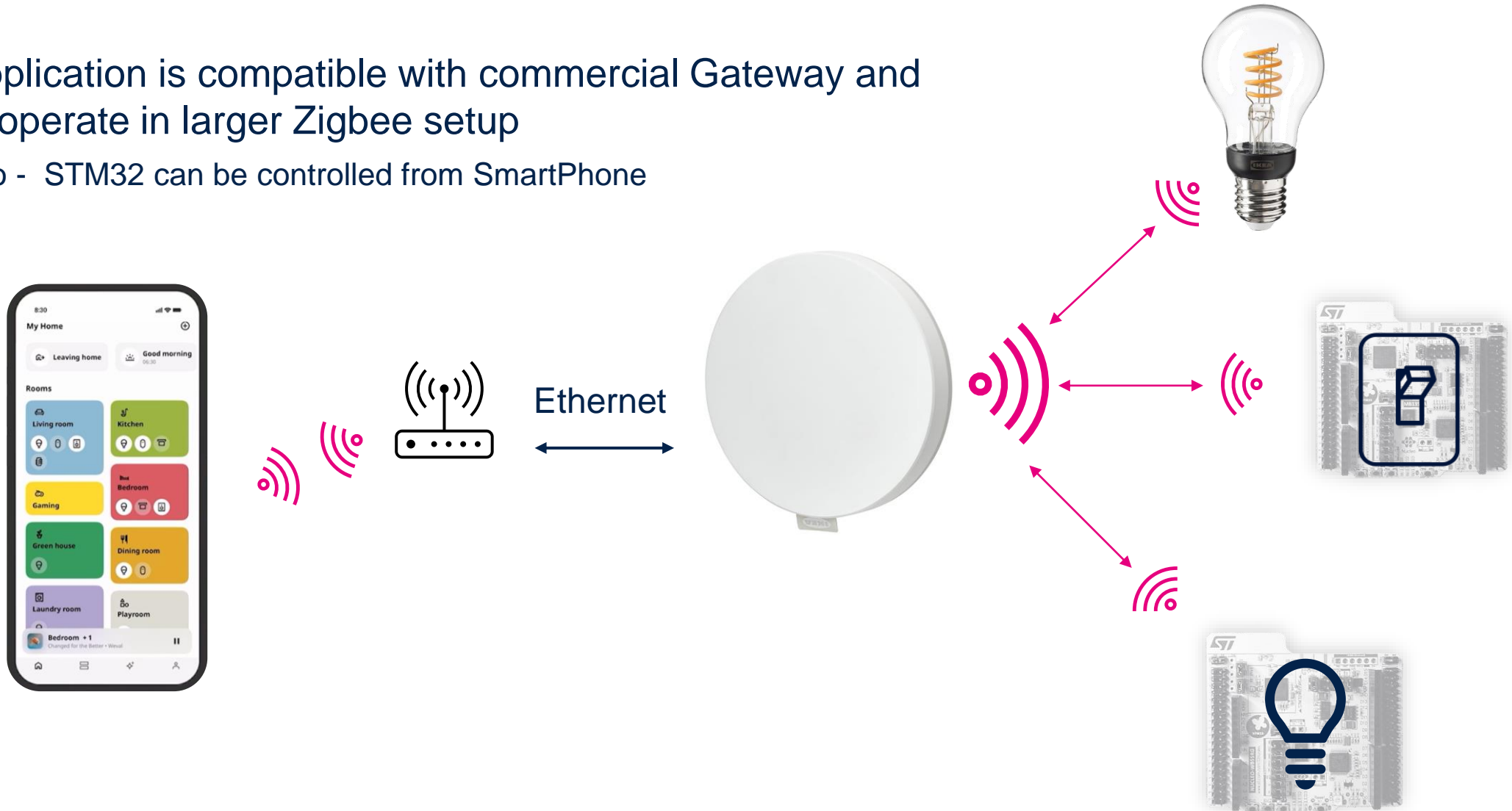
- The application is compatible with 3rd party products available on the market
 - Connect STM32 to the product with Touchlink
 - Control 3rd party Lightbulb product with STM32WB Nucleo
 - Control STM32WB Nucleo LEDs with 3rd party Lightswitch





Larger setup with commercial coordinator

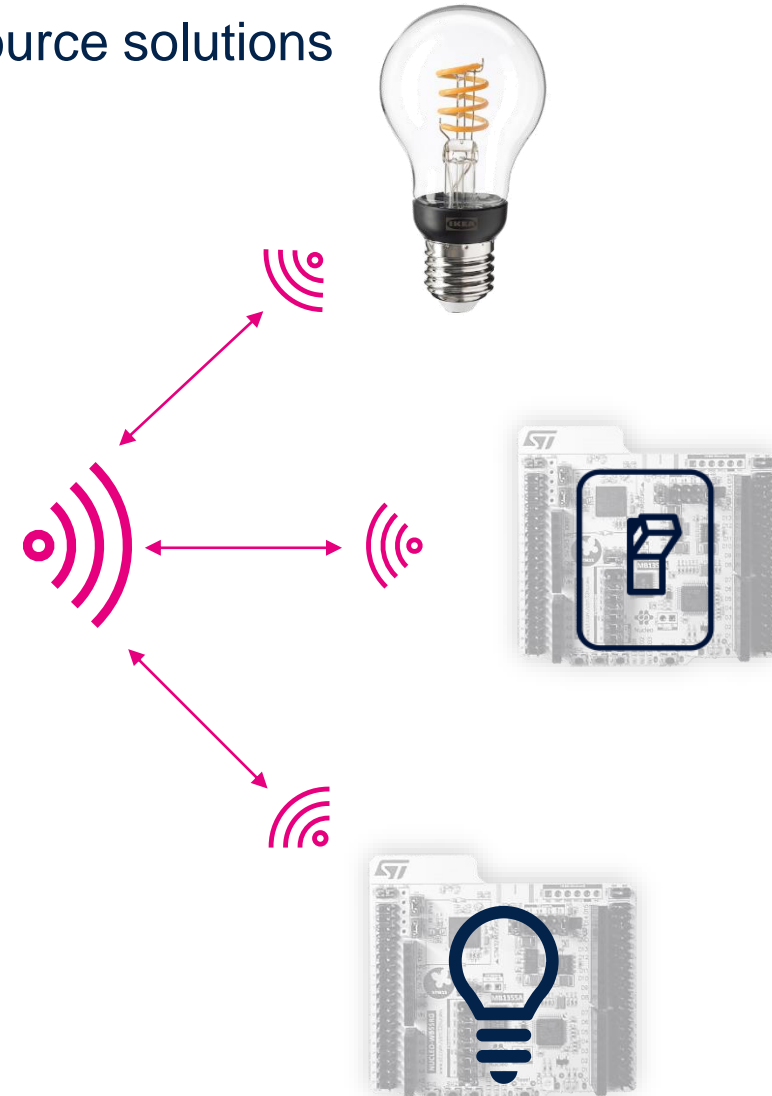
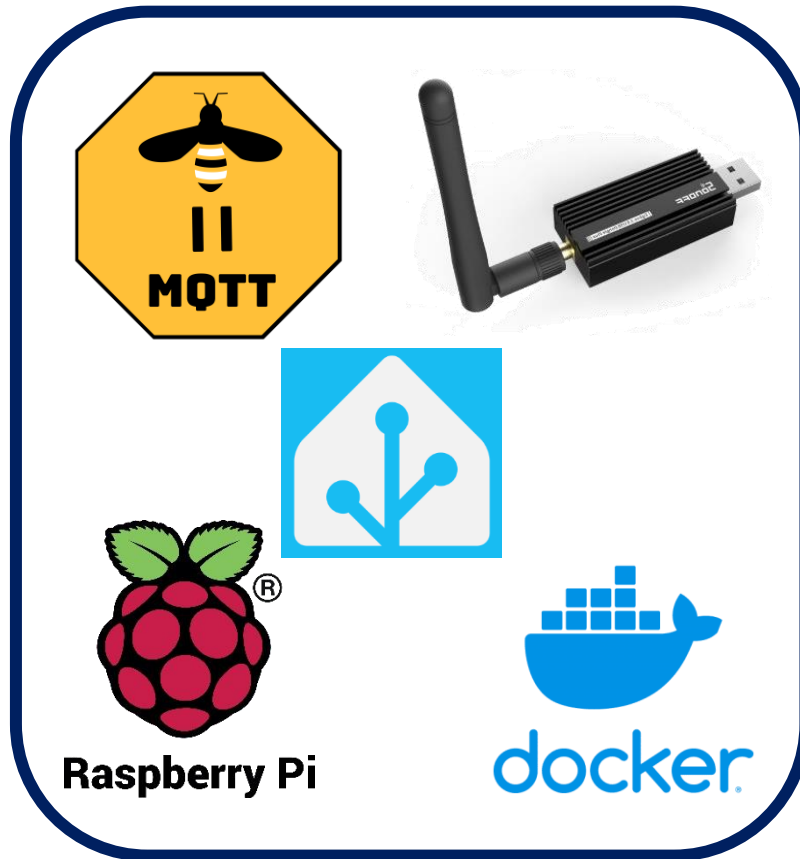
- The application is compatible with commercial Gateway and can cooperate in larger Zigbee setup
 - Demo - STM32 can be controlled from SmartPhone





Open source solution

- STM32 Zigbee solution is compatible with OpenSource solutions like Home Assistant and others...



Cluster management

Change On/Off to Heater/Thermostat



Let's start your Zigbee journey with STM32WBx today!

Part 4: Switching On/Off to Heater/Thermostat

Cluster management

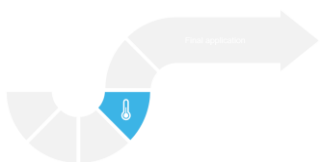
Create advanced application scenarios by customizing clusters.

We will add thermostat cluster and show how callbacks could be implemented.



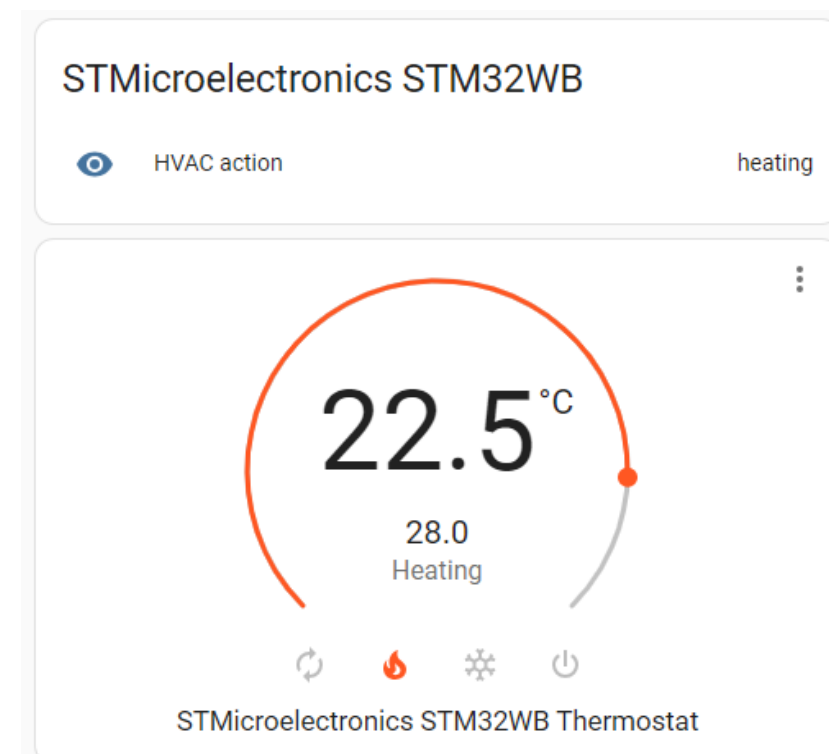
Cluster management

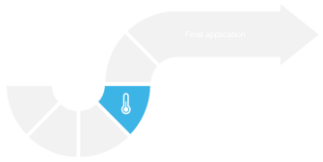
- Add & configure clusters from ZCL or create custom one
- We will show how to add thermostat cluster.



Purpose

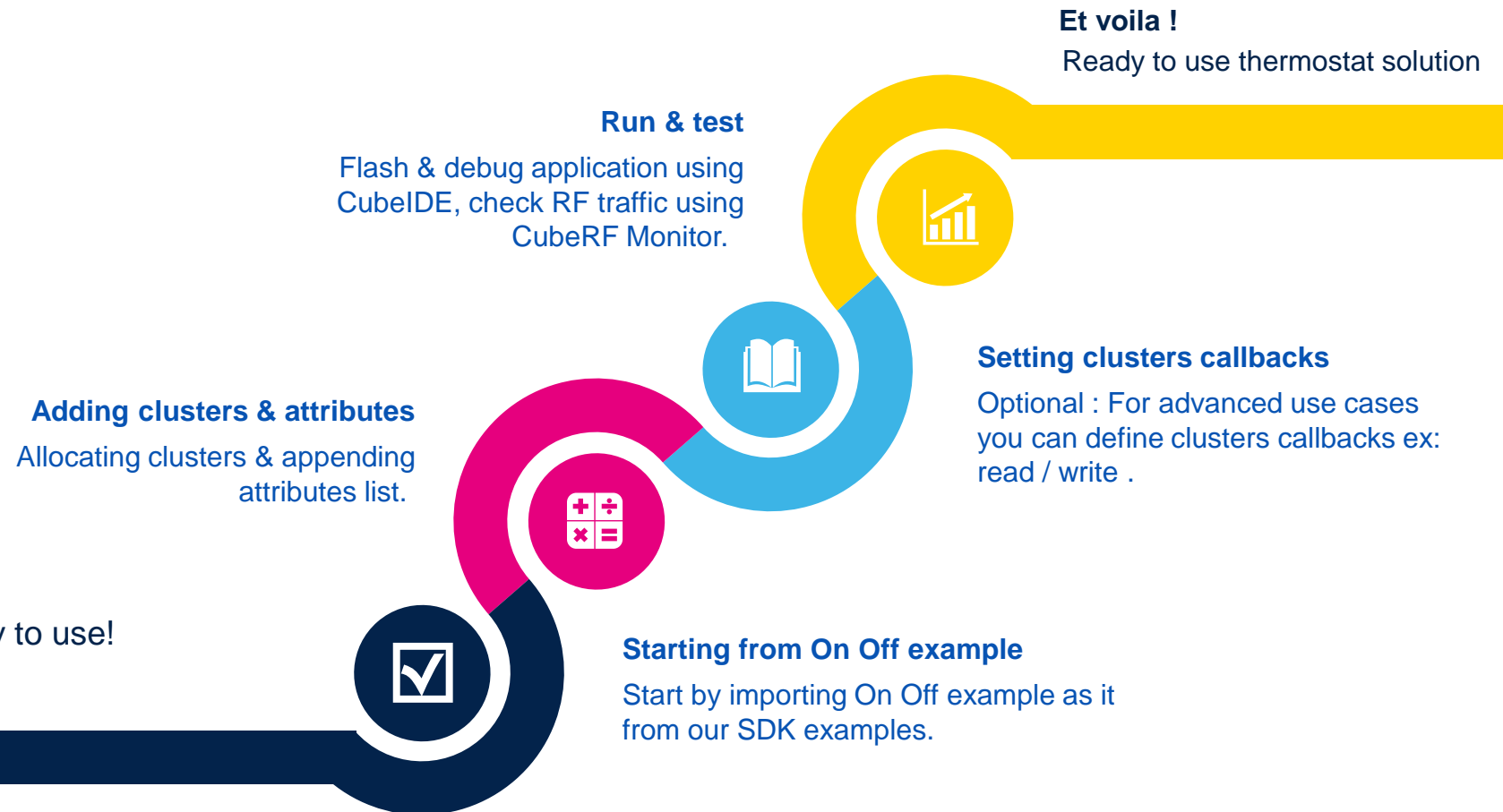
- In this hands-on we will demonstrate how to add & configure ZCL thermostat cluster.
- Application Features:
 - ZCL thermostat cluster with standard & custom attributes
 - Integration with Home Assistant
 - Managing cluster read / write callbacks
 - OTA Client support





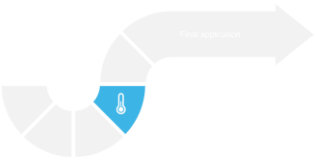
Thermostat application

Setting up thermostat application in 4 steps



Visit STM32Developer Zone

It all starts here, hundreds of demos ready to use!



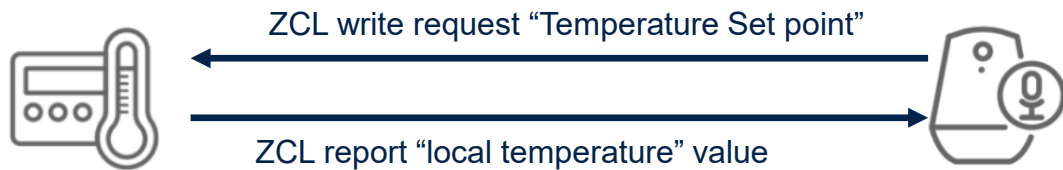
Thermostat Application : ZCL introduction

Client / Server clusters

A cluster defines a set of attributes, commands & responses. A cluster can be implemented in two ways:

- As server
- As client

Application can configure **Binding / reporting** between different clusters/endpoints to automate data transfer across network nodes.

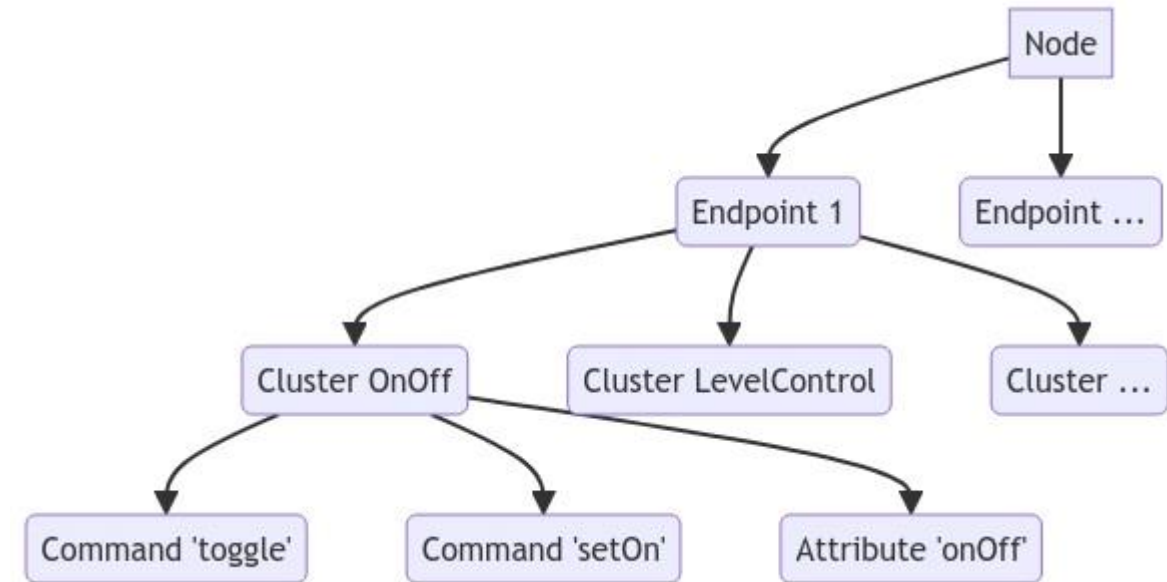


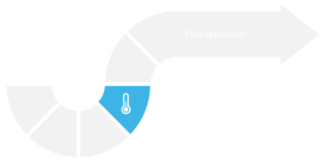
Thermostat
Thermostat Server

Gateway
Thermostat Client

Zigbee Cluster Library

Defined in Zigbee Document 075123





Thermostat Application : ZCL binding concept

ZigBee binding

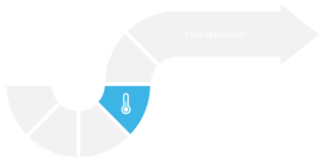
Binding in ZigBee allows an endpoint/cluster identifier pair on one node to be connected to one or more endpoints on other nodes

How to configure binding

- Local configuration using **ZbApsmeBindReq()**
- Remote configuration using **ZbZdoBindReq()**.
- During joining using Find and Bind mechanism
By setting bdbCommissioningMode to
BDB_COMMISSIONING_MODE_FIND_BIND

Binding Table Format

Source			Destination	
IEEEAddr	EP	ClusterID	IEEEAddr	EP
Addr_0	EP_0	CI_0	Addr_00	EP_00
...
Addr_i	EP_i	CI_i	Addr_i0	EP_i0



Thermostat Application : ZCL reporting concept

ZigBee reporting

Attributes reporting allow application to **automatically** / **periodically** report's attribute value to cluster clients.

On regular time interval

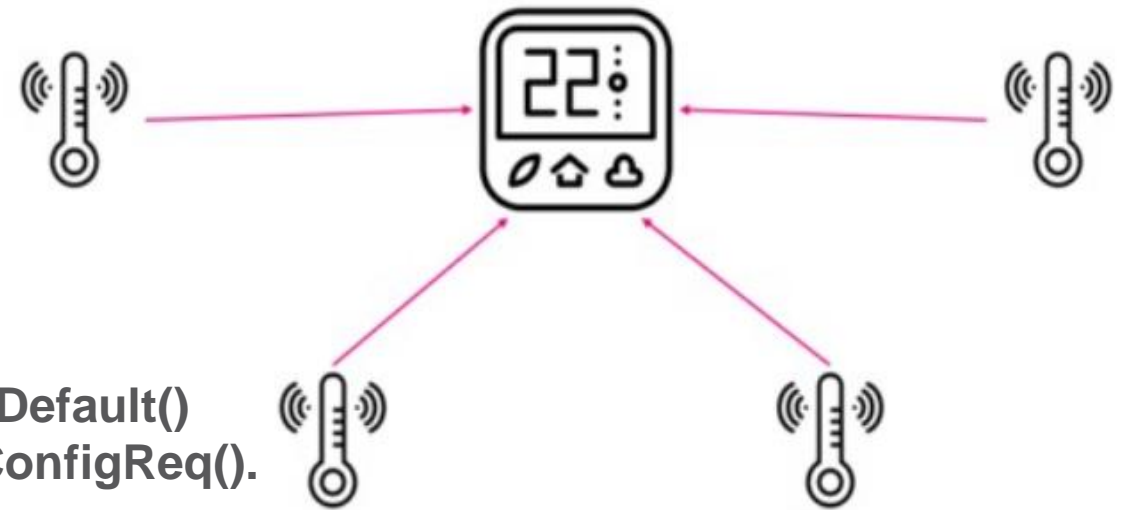
On value change

How to configure reporting

- Local configuration on server via **ZbZclAttrReportConfigDefault()**
- Remote configuration from client using **ZbZclAttrReportConfigReq()**.

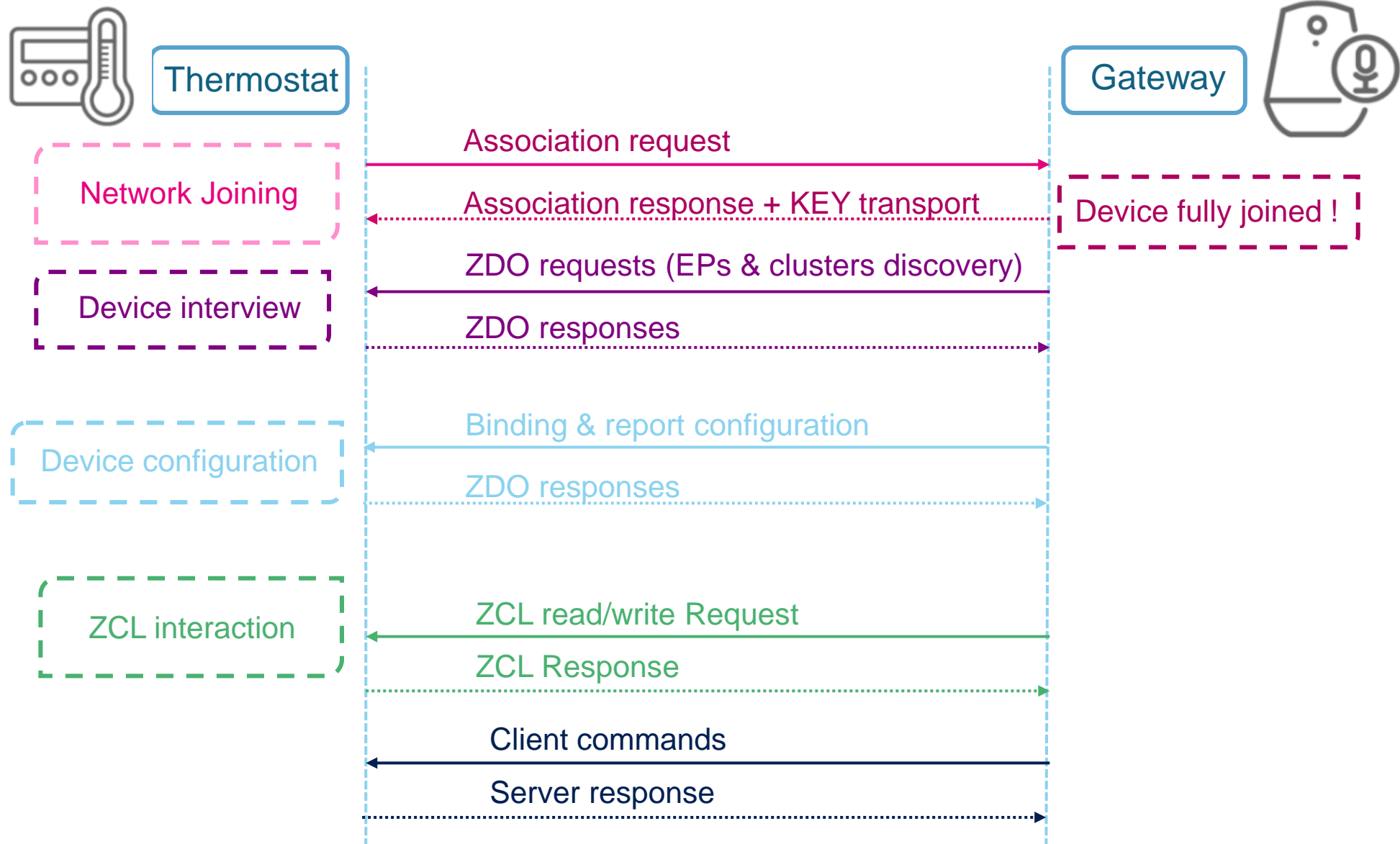
Note that not all attributes in a cluster are reportable. The ZCL specification defines whether reporting is mandatory, optional, or not supported for an attribute.

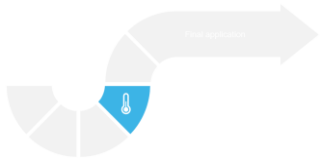
Thermostat use case





Thermostat : Device onboarding

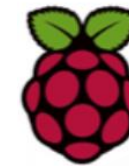




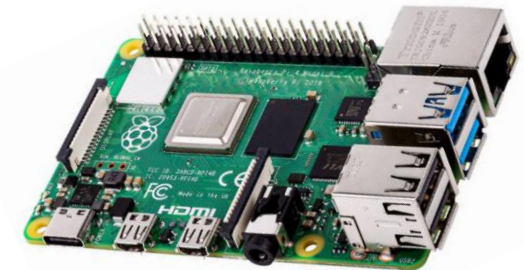
Thermostat application : Demo set up



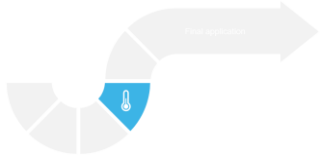
STM32WB55 running
Thermostat demo application



STM32WB USB dongle
802.15.4 sniffer



Raspberry 4B
Home Assistant + Zigbee2MQTT

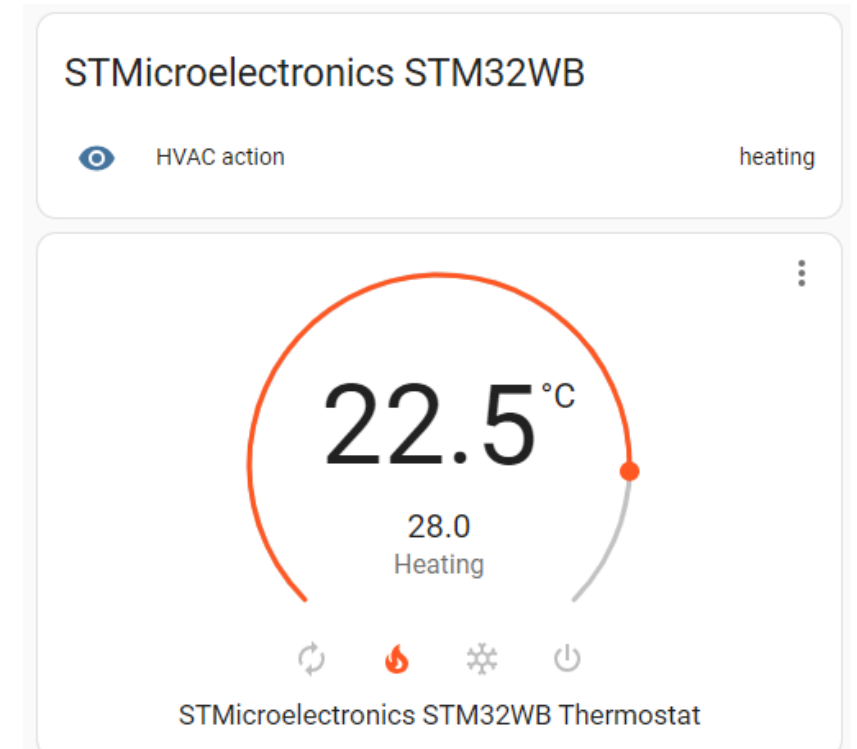


Thermostat application : Live demo



Get callback for each new request.

Print new temperature on serial monitor.



Change set point / thermostat mode from HA GUI

Large Zigbee network

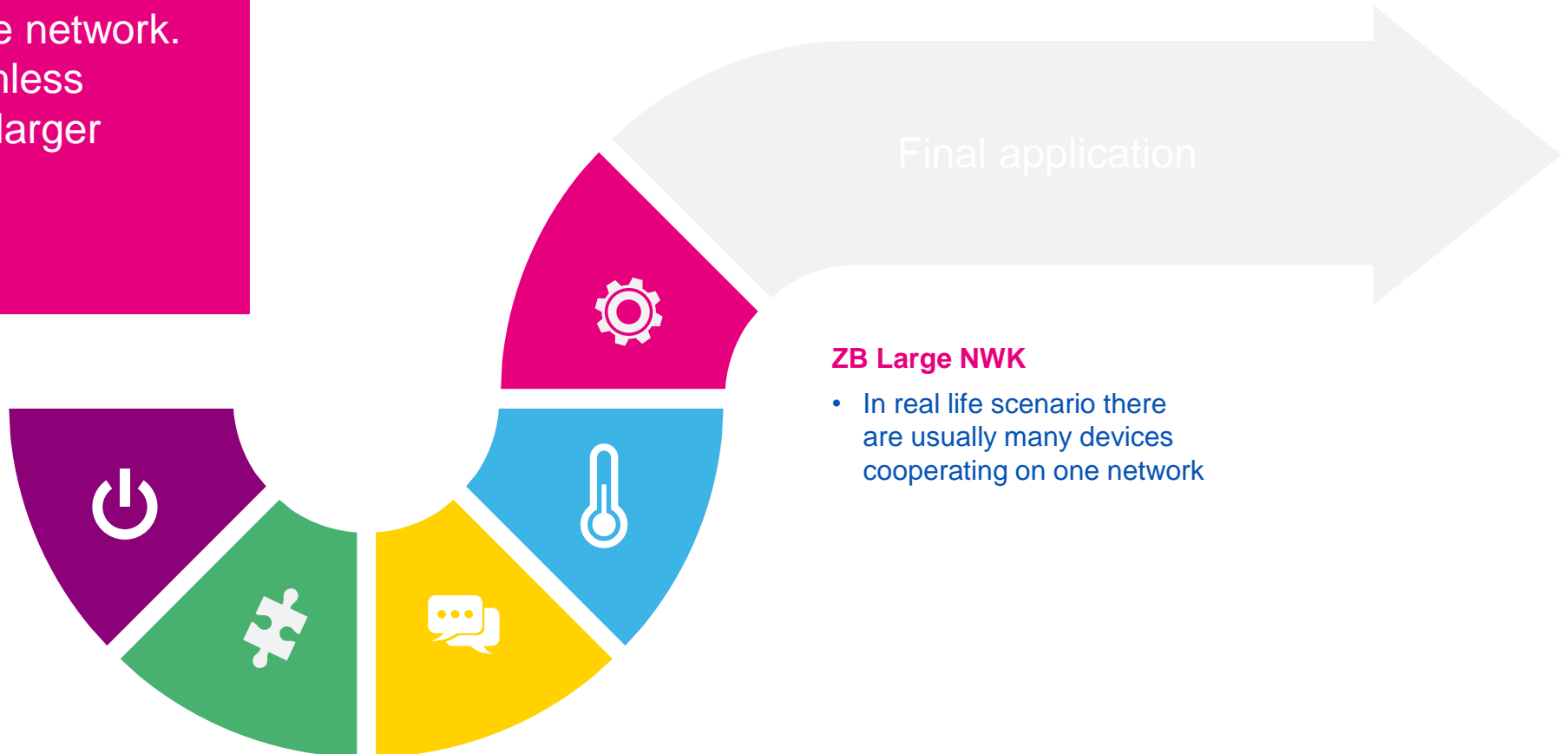
Let's start your Zigbee journey with STM32WBx today!

Part 5: Large Zigbee Network

ZB Large network

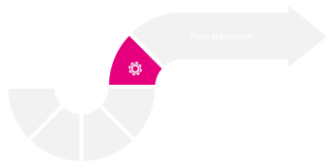
Test in real-life scenarios.

Usually many devices are cooperating on one network. We will show seamless integration within larger network

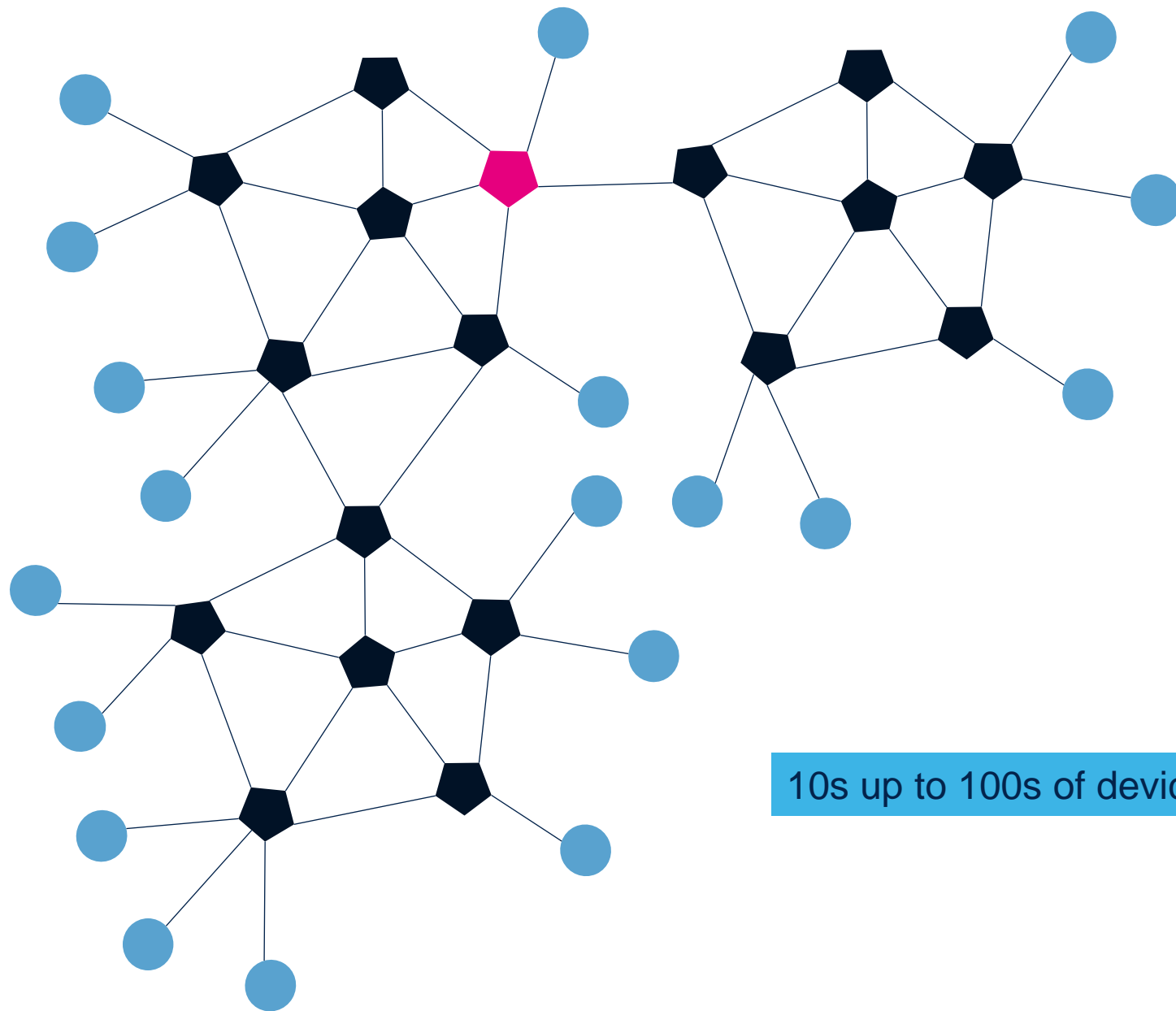


ZB Large NWK

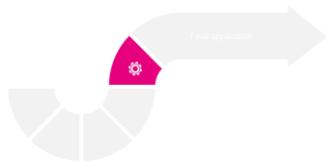
- In real life scenario there are usually many devices cooperating on one network



Large network

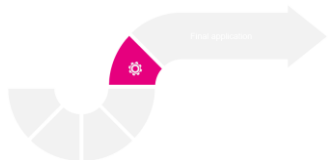


10s up to 100s of devices in the same network typically



Purpose

- In this part we will demonstrate how to use our examples to setup and test a larger Zigbee network and a practical example of what else to consider during development and deployment to the field
- Application Features:
 - Large network setup
 - Coordinator configuration for large network support
 - Noisy environment
 - Zigbee Channels Agility



Larger network test setup

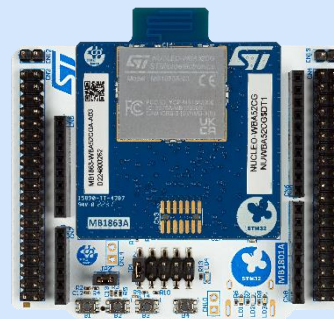
A

5x NUCLEO-WB55RG

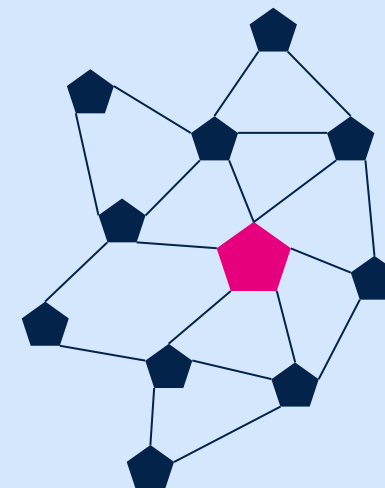


Zigbee Coordinator / Router

5x NUCLEO-WBA55CG



Zigbee Router



B

1x NUCLEO-WB55RG

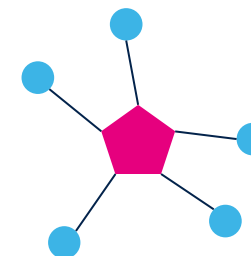


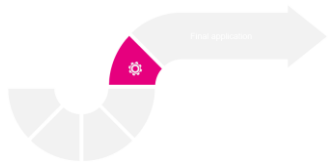
Zigbee Coordinator

60x MB1293 (P-NUCLEO-WB55RG)



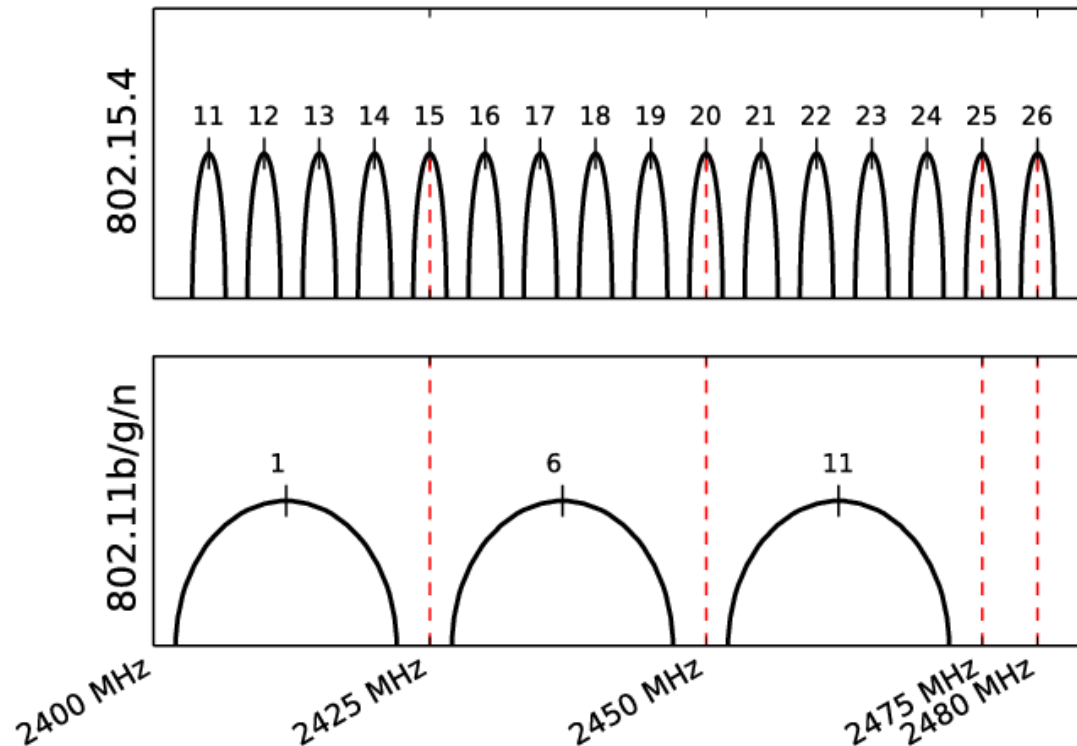
Zigbee End Device





How about noisy or crowded environment?

- Your network might not be alone at particular IEEE 802.15.4 channel (Zigbee, Thread, proprietary)
- There is an overlap between some channels of IEEE 802.15.4 and IEEE 802.11b/g/n
- All this traffic may have impact on the solution robustness and user experience



1) Select well the channel (if possible)

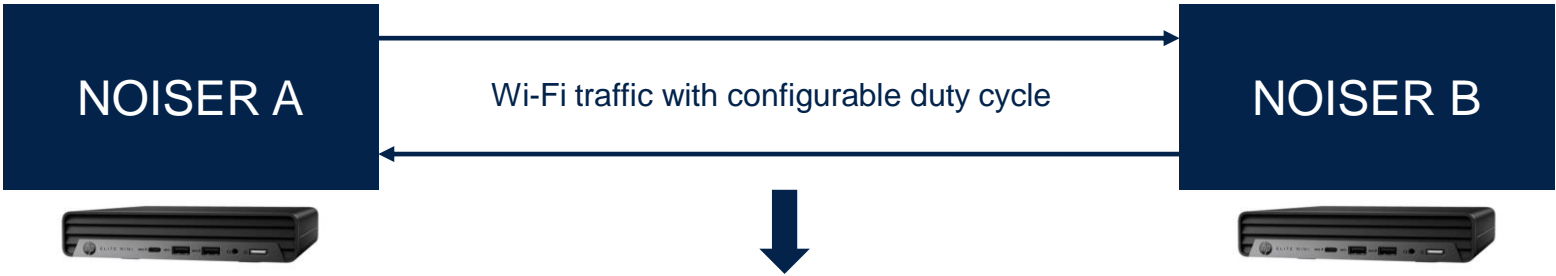
2) Increase the output power and improve RF performances(if possible)



Advanced sniffer
IEEE 802.15.4 (all channels)
+
IEEE 802.11 traffic

Larger network test setup

IEEE 802.11 Channel 1

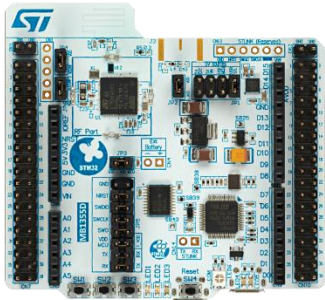


IEEE 802.15.4 Channel 12

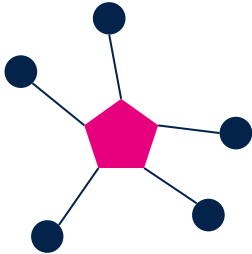
1x NUCLEO-WB55RG

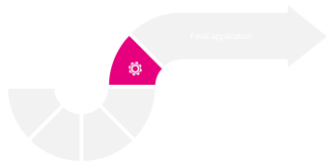
60x MB1293 (P-NUCLEO-WB55RG)

B



+





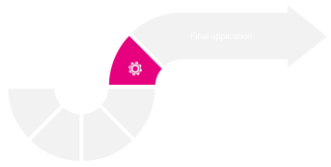
Zigbee Frequency/Channels Agility

- Optional feature in some Zigbee stacks
- Can be used to trigger a change of the channel in case of issues in communication are detected
- Can be used to select an empty (the least crowded) channel when the network is formed

STM32CubeWB SDK (STM32CubeWBA SDK – coming soon)

Projects\P-NUCLEO-WB55.Nucleo\Applications\Zigbee\

- Zigbee_OnOff_ChannelsAgility_SED
- Zigbee_OnOff_ChannelsAgility_ZC
- Zigbee_OnOff_ChannelsAgility_ZR



Memory impacts

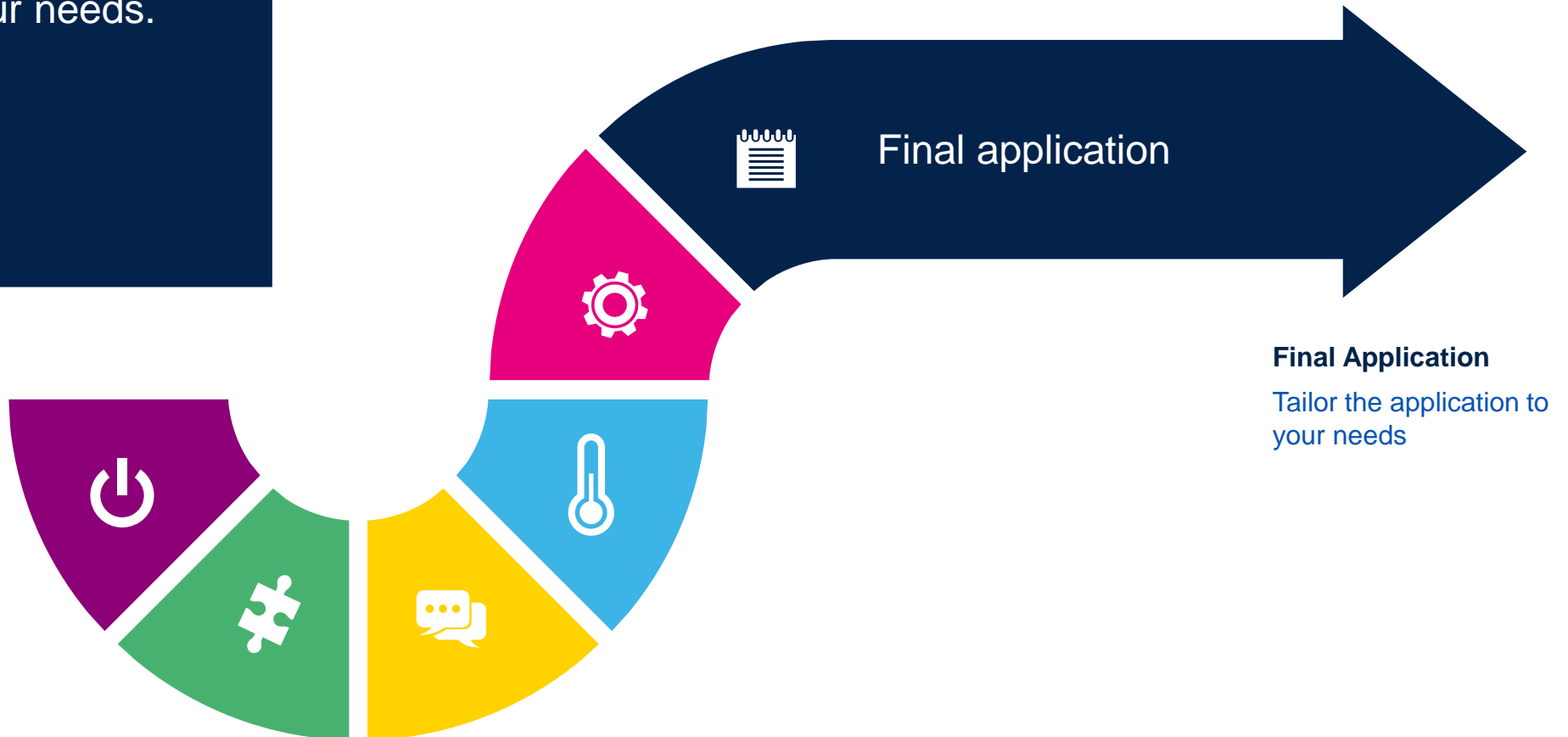
- Coordinator and Router needs large amounts of SRAM and FLASH
 - SRAM for all the network data, tables and SED parent buffers
 - FLASH to store the persistence data

AN5492 – Persistent data management ZigBee® and non-volatile memory in STM32WB Series

Let's start your Zigbee journey with STM32WBx today!

Final application

Add your own code to tailor the application to your needs.



Final Application
Tailor the application to
your needs

Time for Survey

Workshop survey

- Please provide us with your feedback about this workshop.
- Your inputs are very valuable and important for us.



https://www.surveymonkey.com/r/Zigbee_DM_WS_24

Thank you!



Takeaways

Full-featured Zigbee ecosystem

During development of your new Zigbee product you will take advantage of rich and continuously expanding ecosystem around STM32

Quick setup and prototyping

You do not need any prior knowledge of STM32 to get started with the STM32 Zigbee solution

Benefits of STM32 platform DNA

Many I/Os, lot of peripherals and general features and more security at lower power consumptions

Field proven Zigbee stack solution

The CubeMX package contains more than 50 ready-made Zigbee examples and is still growing

Enjoy ease of use of ST wireless ecosystem

In case you need, ST's strong wireless support team is here to help!

Let's start your Zigbee journey with STM32WBx today!

Links

Useful links

- STM32 Developer Zone - www.st.com/stm32-dev-zone
- Github repositories STMicroelectronics - STM32 Hotspot - github.com/stm32-hotspot
- Getting Started with STM32WB-WBA and Zigbee STM32 MCU Wiki pages:
 - wiki.st.com/stm32mcu/wiki/Connectivity:Getting_Started_with_Zigbee
- STM32WB Getting Started Video Series :
 - www.youtube.com/playlist?list=PLnMKNibPkDnG9JRe2fbOOpVpWY7E4WbJ-
- Nucleo boards product pages:
 - www.st.com/en/evaluation-tools/nucleo-wb55rg.html
 - www.st.com/en/evaluation-tools/nucleo-wba55cg.html
- AN5506 - Getting started with Zigbee® on STM32WB series:
 - www.st.com/resource/en/application_note/an5506-getting-started-with-zigbee-on-stm32wb-series-stmicroelectronics.pdf

Q&A



Thank you

© STMicroelectronics - All rights reserved.

The STMicroelectronics corporate logo is a registered trademark of the STMicroelectronics group of companies. All other names are the property of their respective owners.



life.augmented