



Installation and configuration of ZigbeetoMqtt on Windows 11

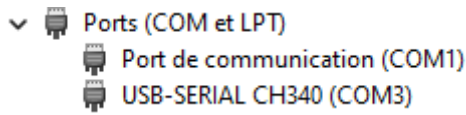
In first time, you must have an Zigbee Usb Adaptater compatible.

<https://www.zigbee2mqtt.io/guide/adapters/>

For me i use a Zigbee 3.0 USB adapter based on Silicon Labs EFR32MG21 with EZSP v8 from Aliexpress, which run fine.



When you plug it you must find the good COM number.



For me COM3...

Download & Install Node.js

Node.js is a library specialized in network and written in javascript.

You must download the last version here and execute : <https://nodejs.org/en>

Download & install zigbee2mqtt










This link is a good guide for installing.

https://www.zigbee2mqtt.io/guide/installation/05_windows.html

in my configuration the location of the zigbee2mqtt is here :

c:\Users\<name user>\documents\zigbee2mqtt

in this directory you must configure data\configuration.yaml

	log
	configuration.example Type : Fichier source Yaml
	configuration Type : Fichier source Yaml
	coordinator_backup Type : Fichier source JSON
	database Type : Data Base File
	database.db.backup Type : Fichier BACKUP
	state Type : Fichier source JSON
	stm32WB5 - backup Type : Fichier source JavaScript
	stm32WB5 Type : Fichier source JavaScript

You must modify your **configuration.yaml** :

```

homeassistant: false
frontend: {}
mqtt:
  server: mqtt://myserver.com:1883
  base_topic: zigbee2mqtt
  user: myuser
  password: mypassword
  force_disable_retain: false
serial:
  adapter: ezsp
  port: \\.\COM3
advanced:
  network_key:
    - 217
    - 145
    - 238
    - 190
    - 25
    - 53
    - 239
    - 2
    - 252
    - 99
    - 160
    - 200
    - 244
    - 205
    - 243
    - 38
  pan_id: 23633
  log_level: debug
  log_syslog:
    app_name: Zigbee2MQTT
    eol: /n
  host: localhost
  localhost: localhost

```

```

path: /dev/log
pid: process.pid
port: 514
protocol: udp4
type: '5424'
channel: 15
homeassistant_legacy_entity_attributes: false
legacy_api: false
legacy_availability_payload: false
cache_state: false
output: attribute
external_converters:
  - stm32WB5.js
device_options:
  legacy: false
availability:
  active:
    # Time after which an active device will be marked as offline in
    # minutes (default = 10 minutes)
    timeout: 5
  passive:
    # Time after which a passive device will be marked as offline in
    # minutes (default = 1500 minutes aka 25 hours)
    timeout: 1500
devices:
  # fix here address of your own card
  '0x0080e125006efe86':
    friendly_name: STM32WB5MM-DK
    availability:
      #every minute
      timeout: 1
passlist:
  # fix here address of your own card
  - '0x0080e125006efe86'

```

In the mqtt part you specify the address of your server, the main topic to use, the username for this topic and the password. You can configure if necessary the certificates used for security.

In **advanced** : section you must add

```

channel: 15
homeassistant_legacy_entity_attributes: false
legacy_api: false
legacy_availability_payload: false
cache_state: false
output: attribute

```

Channel is the channel that zigbeetomqtt will listen to in my example, channel 15.

cache_state= false is very important, this prevents zigbee2mqtt from transmitting the olds values of the attributes each time with the one you are querying.

Output : attribute – generate one topic by attribute read.

```

external_converters:
  - stm32WB5.js

```

The **stm32WB5.js** file must be added in this same directory to reference the stm32wb5mm-dk card.

This configuration file will tell zigbee2mqtt:

fzL: List of attribute converters from the Zigbee network.

tzL: List of converters allowing you to transmit an attribute to a cluster id and attribute id or to query.

fz and tz: references the attributes of the standard.

e: Provision of attributes and display mode in the zigbee2mqtt interface.

My copy of this file is joined to this project.

Zigbee2mqtt have an web interface : zigbee-frontend. In the distribution this interface is compiled, but here it is necessary to add some attributes informations.

In file zigbee2mqtt/node_modules/zigbee-hersman/dist/zcl/definition/cluster.js

You must modify Cluster and add some attributes to Cluster 1026 :

In file zigbee2mqtt/node_modules/zigbee-herdsman-converters/lib/expose.js

You must modify and add

```
msTemperatureMeasurement: {
  ID: 1026,
  attributes: {
    measuredValue: { ID: 0, type: dataType_1.default.int16 },
    minMeasuredValue: { ID: 1, type: dataType_1.default.int16 },
    maxMeasuredValue: { ID: 2, type: dataType_1.default.int16 },
    tolerance: { ID: 3, type: dataType_1.default.uint16 },
    minPercentChange: { ID: 16, type: dataType_1.default.unknown },
    minAbsoluteChange: { ID: 17, type: dataType_1.default.unknown },
    BatteryVoltage: { ID: 0x0101, type: dataType_1.default.uint16 },
    DelayRefreshMeas: { ID: 0x0102, type: dataType_1.default.uint16 },
    Luminosity: { ID: 0x0300, type: dataType_1.default.uint16 },
    Pressure: { ID: 0x0301, type: dataType_1.default.uint16 },
    Humidity: { ID: 0x0302, type: dataType_1.default.uint16 },
    TemperatureEnv: { ID: 0x0303, type: dataType_1.default.int16 },
    AirQuality: { ID: 0x0304, type: dataType_1.default.uint8 },
    Co2: { ID: 0x0305, type: dataType_1.default.uint16 },
    Tvoc: { ID: 0x0306, type: dataType_1.default.uint16 },
    R1: { ID: 0x0307, type: dataType_1.default.uint32 },
    R2: { ID: 0x0308, type: dataType_1.default.uint32 },
    R3: { ID: 0x0309, type: dataType_1.default.uint32 },
    R4: { ID: 0x030A, type: dataType_1.default.uint32 },
    WhiteLuminosity: { ID: 0x030B, type: dataType_1.default.uint16 },
    TankLevel: { ID: 0x030C, type: dataType_1.default.uint8 },
    PlantTemperature1: { ID: 0x0310, type: dataType_1.default.int16 },
    PlantMACDS1: { ID: 0x0311, type: dataType_1.default.uint64 },
    PlantSoilMoisture1: { ID: 0x0312, type: dataType_1.default.uint8 },
    PlantWaterAlimentation1: { ID: 0x0313, type: dataType_1.default.uint16 },
    PlantWaterAlimentationStatel: { ID: 0x0314, type: dataType_1.default.uint8 },
    PlantTemperature2: { ID: 0x0320, type: dataType_1.default.int16 },
    PlantMACDS2: { ID: 0x0321, type: dataType_1.default.uint64 },
    PlantSoilMoisture2: { ID: 0x0322, type: dataType_1.default.uint8 },
    PlantWaterAlimentation2: { ID: 0x0323, type: dataType_1.default.uint16 },
    PlantWaterAlimentationState2: { ID: 0x0324, type: dataType_1.default.uint8 },
    PlantTemperature3: { ID: 0x0330, type: dataType_1.default.int16 },
    PlantMACDS3: { ID: 0x0331, type: dataType_1.default.uint64 },
    PlantSoilMoisture3: { ID: 0x0332, type: dataType_1.default.uint8 },
    PlantWaterAlimentation3: { ID: 0x0333, type: dataType_1.default.uint16 },
    PlantWaterAlimentationState3: { ID: 0x0334, type: dataType_1.default.uint8 },
    PlantLight: { ID: 0x0340, type: dataType_1.default.uint8 },
    sprutTemperatureOffset: { ID: 0x6600, type: dataType_1.default.int16, manufacturerCode: manu
  },
},
```

2 files are joins in the project.

database and state Files of Zigbee2mqtt

In the file **database** you can find all entry of devices joined with Zigbee2mqtt.

If you want to make a new join procedure for a device, you must delete the correspondent line in this file.

```
{ "id":1, "type": "Coordinator", "ieeeAddr": "0xf4b3b1fffe45ceaf", "nwkAddr": 0, "manufId": 0, "epList": [1, 242], "endpoi
{ "id":2, "type": "Router", "ieeeAddr": "0x0080e125006efe86", "nwkAddr": 28282, "manufId": 4311, "manufName": "STMicroelectr
{ "id":3, "type": "EndDevice", "ieeeAddr": "0xa4c138f3fedcb468", "nwkAddr": 25061, "manufId": 4098, "manufName": "_TZ3000
```

And restart zigbee2mqtt and make a new join.

In the file **stat**, you can find in JSON format, the last value of exposed attributes for all devices.

```
{
  "0x0080e125006efe86": {
    "temperature": 24.87,
    "BatteryVoltage": 4.142,
    "TemperatureEnv": 20.93,
    "linkquality": 216,
    "Luminosity": 0,
    "PlantTemperature1": 17.87,
    "Pressure": 994,
    "Co2": 417,
    "Tvoc": 31,
    "R1": 564816,
    "R2": 1,
    "R3": 1658138,
    "R4": 56621,
    "PlantMACDS1": "0x286cfc56b5013ca6",
    "PlantSoilMoisture1": 0,
    "PlantWaterAlimentation1": 0,
    "Humidity": 51,
    "WhiteLuminosity": 269,
    "AirQuality": 1,
    "PlantLight": 0,
    "PlantWaterAlimentation2": 0,
    "PlantWaterAlimentation3": 0,
    "PlantWaterAlimentationState1": 0,
    "PlantMACDS2": "0x28da1956b5013c1b",
    "PlantTemperature2": 20.93,
    "PlantSoilMoisture2": 82.95,
    "PlantWaterAlimentationState2": 0,
    "PlantTemperature3": 18,
    "PlantMACDS3": "0x28e90156b5013ced",
    "PlantSoilMoisture3": 0,
    "PlantWaterAlimentationState3": 0,
    "DelayRefreshMeas": 20
  },
}
```

Modify Zigbee-frontend

Zigbee2mqtt originally has a frontend interface, this interface is configured with standard Clusters and attributes. But if you wish to display and manage additional attributes in the drop-down menus, I will show you a procedure for updating the Front-end, which is compiled in the base distribution.

You must download the github zigbee2mqtt-frontend and compile the dist directory which will have to be placed in the zigbee2mqtt frontend plug-in.

Download the GitHub.

<https://github.com/nurikk/zigbee2mqtt-frontend>

you must update with the two precedents files.

- ***zigbee2mqtt-frontend/node_modules/zigbee-hersman/dist/zcl/definition/cluster.js***
- ***zigbee2mqtt-frontend/node_modules/zigbee-herdsman-converters/lib/expose.js***

In command mode, you compile with **npm run build** in zigbee2mqtt-frontend directory.

You must copy the complete directory ***zigbee2mqtt-frontend/dist*** to the directory to ***zigbee2mqtt/node_modules/zigbee2mqtt-frontend/dist***

Now for starting, zigbeetomqtt write :

npm start in cmd mode from install directory.

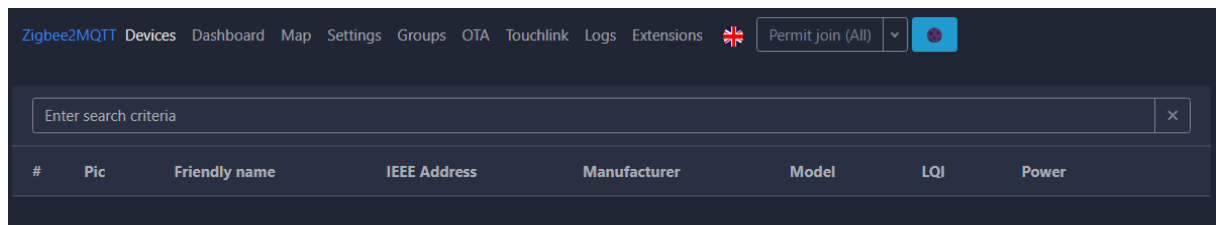
When Zigbee2mqtt start, it try to contact the usb device and open the network. (Use CTRL + C for stopping or close the window).

```
Zigbee2MQTT:debug 2024-02-13 11:10:48: Loaded state from file C:\Users\mag\Documents\zigbee2mqtt\data\state.json
Zigbee2MQTT:info 2024-02-13 11:10:48: Logging to console and directory: 'C:\Users\mag\Documents\zigbee2mqtt\data\log\2024-02-13.11-10-48' filename: log.txt
Zigbee2MQTT:debug 2024-02-13 11:10:48: Removing old log directory 'C:\Users\mag\Documents\zigbee2mqtt\data\log\2024-01-31.15-26-11'
Zigbee2MQTT:info 2024-02-13 11:10:48: Starting Zigbee2MQTT version 1.33.2 (commit #unknown)
Zigbee2MQTT:info 2024-02-13 11:10:48: Starting zigbee-herdsman (0.21.0)
Zigbee2MQTT:debug 2024-02-13 11:10:48: Using zigbee-herdsman with settings: '{"adapter":{"concurrent":null,"delay":null,"disableLED":false},"backupPath":"C:\\Users\\mag\\Documents\\zigbee2mqtt\\data\\coordinator_backup.json","databaseBackupPath":"C:\\Users\\mag\\Documents\\zigbee2mqtt\\data\\database.db.backup","databasePath":"C:\\Users\\mag\\Documents\\zigbee2mqtt\\data\\database.db","network":{"channelList":[15],"extendedPanID":[221,221,221,221,221,221,221,221],"networkKey":["HIDDEN","panID":23633],"serialPort":{"adapter":"ezsp","path":"\\\\\\\\.\\\\COM3"}}}'
Zigbee2MQTT:info 2024-02-13 11:10:50: zigbee-herdsman started (resumed)
Zigbee2MQTT:info 2024-02-13 11:10:50: Coordinator firmware version: '{"meta":{"maintrel":"3 ","majorrel":"6","minorrel":"10"},"product":8,"revision":"6.10.3.0 build 297"},"type":"EZSP v8"}'
Zigbee2MQTT:debug 2024-02-13 11:10:50: Zigbee network parameters: {"channel":15,"extendedPanID":221,"panID":23633}
Zigbee2MQTT:info 2024-02-13 11:10:50: Currently 2 devices are joined:
Zigbee2MQTT:info 2024-02-13 11:10:50: STM32WB5MM-DK (0x0080e125006efe86): STM32WB - STMicroelectronics Plantation feeding management - PFM - Elektor Project Contest 2024 (Router)
Zigbee2MQTT:info 2024-02-13 11:10:50: 0xa4c138f3fedcb468 (0xa4c138f3fedcb468): ERS-10TZBVB-AA - TuYa Smart button (EndDevice)
Zigbee2MQTT:info 2024-02-13 11:10:50: Zigbee: disabling joining new devices.
Zigbee2MQTT:info 2024-02-13 11:10:51: Connecting to MQTT server at mqtt://vps1.sophe.com:1883
Zigbee2MQTT:debug 2024-02-13 11:10:51: Using MQTT login with username: dev
Zigbee2MQTT:info 2024-02-13 11:10:51: Connected to MQTT server
Zigbee2MQTT:info 2024-02-13 11:10:51: MQTT publish: topic 'zigbee2mqtt/bridge/state', payload '{"state":"online"}'
Zigbee2MQTT:info 2024-02-13 11:10:51: Started frontend on port 0.0.0.0:8080
Zigbee2MQTT:info 2024-02-13 11:10:51: Zigbee2MQTT started!
```

Now you can have access to web gui interface of zigbee2mqtt on :

<http://localhost:8080>

when you want to make a new clean join for endpoint, you must erase the correspondant line in database.



For forcing the card stm32wb5mm-dk to join the coordinator, you must press the button one (at left) and power on. You count 3 seconds and release the button.

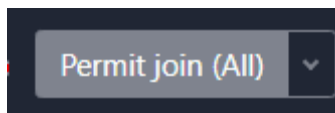
A screen appears :



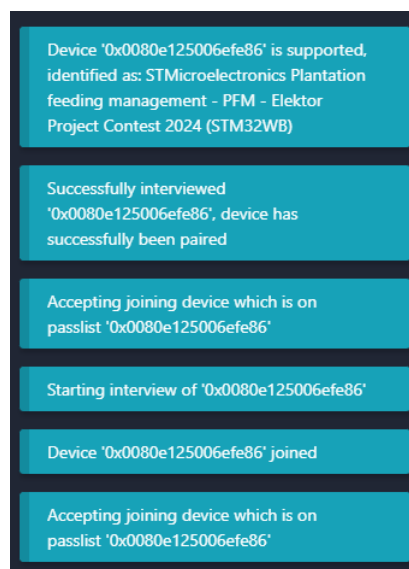
And the box try to rejoin the coordinator every 500ms.

```
[M4 APPLICATION] ZbStartup Callback (status = 0xca)
[M4 APPLICATION] Startup failed, attempting again after a short delay (500 ms)
[M4 APPLICATION] Network config : APP_STARTUP_CENTRALIZED_ROUTER
[M0] [00000000.519][PLATFORM] ZbNlmeResetReq : NLME-RESET.request (warmStart = 0)
[M0] [00000000.018][PLATFORM] zb_startup_join_nwk_disc : Attempting network discovery. Scans = 3, Duration = 4
[M0] [00000000.019][PLATFORM] nwk_scan_req : MLME-SCAN.request (wpan0): type=1, page=0, mask=0x00008000, dur=4
[M0] [00000000.035][PLATFORM] nwk_handle_beacon_ind : BEACON addr16=0x0000, epid=0xdddddddddddd, ch=15, pan=0x5c51, depth= 0, lqi=213, cost=1, pjoin=0
[M0] [00000000.290][PLATFORM] nwk_scan_req : MLME-SCAN.request (wpan0): type=1, page=0, mask=0x00008000, dur=4
[M0] [00000000.334][PLATFORM] nwk_handle_beacon_ind : BEACON addr16=0x0000, epid=0xdddddddddddd, ch=15, pan=0x5c51, depth= 0, lqi=213, cost=1, pjoin=0
[M0] [00000000.558][PLATFORM] nwk_scan_req : MLME-SCAN.request (wpan0): type=1, page=0, mask=0x00008000, dur=4
[M0] [00000000.599][PLATFORM] nwk_handle_beacon_ind : BEACON addr16=0x0000, epid=0xdddddddddddd, ch=15, pan=0x5c51, depth= 0, lqi=213, cost=1, pjoin=0
```

You can click on Permi Join



Join events



New device added

Zigbee2MQTT

Devices

Dashboard

Map

Settings

Groups

OTA

Touchlink

Logs

Extensions

Disable join (All) 03:51

Enter search criteria

#	Pic	Friendly name	IEEE Address	Manufacturer	Model	LQI	Power	
1		0x0080e125006efe86	0x0080e125006efe86 (0x79D4)	STMicroelectronics	STM32WB	N/A	↓	<div><div></div><div></div><div></div></div>

You can rename the device for fix the Topic path of mqtt :

Click on 

Rename device

STM32WB5MM-DK

Friendly name

STM32WB5MM-DK

Rename device

Click on new device name in list :

2		STM32WB5MM-DK	0x0080e125006efe86 (0x6E7A)	STMicroelectro
---	---	---------------	--------------------------------	----------------

Zigbee2MQTT

Devices

Dashboard

Map

Settings

Groups

OTA

Touchlink

Logs

Extensions

Permit join (All)

STM32WB5MM-DK

About

Exposes

Bind

Reporting

Settings


Settings (specific)

State

Clusters

Scene

Dev console



Friendly name

STM32WB5MM-DK

Description

Last seen

N/A

Availability

Disabled

Device type

Router

Zigbee Model

STM32WB

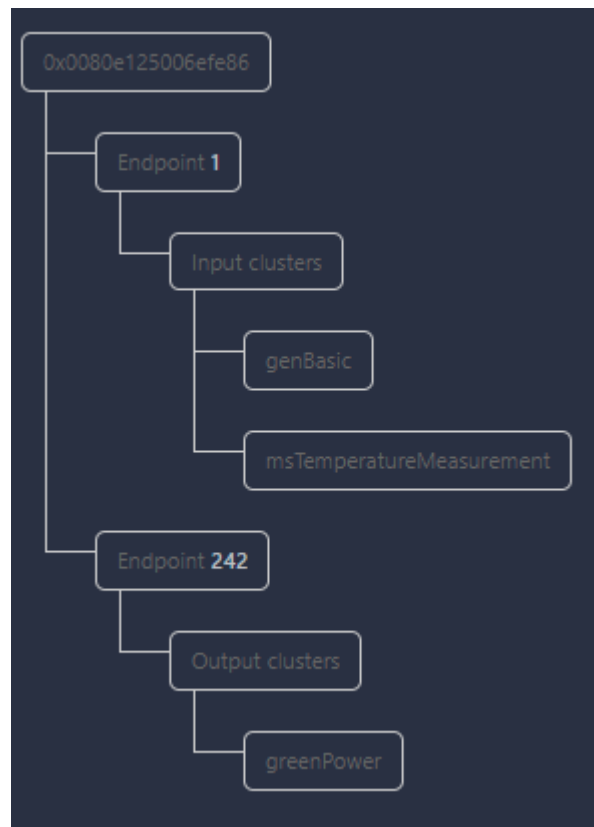
Exposes Menu : View the list of attributes and last values

STM32WB5MM-DK		
About	Exposes	Bind Reporting Settings Set
Temperature		24.91 °C
Measured temperature value		
TemperatureEnv		20.93 °C
Environment Temperature		
BatteryVoltage		4.142 V
Measured Voltage of Battery		
DelayRefreshMeas		20 Sec.
State of Minimum Tank level		
Humidity		47 %
Measured relative humidity		
Luminosity		0
Global luminosity		
WhiteLuminosity		269
Global White luminosity		
Pressure		1004 hPa
The measured atmospheric pressure		
AirQuality		1
Air quality index		

CO2		417 ppm
The measured CO2 (carbon dioxide) value		
TVOC		31 ppb
Total Volatile Organic Compounds value		
R1		564816 ohms
Resistance 1 value		
R2		1 ohms
Resistance 2 value		
R3		1658138 ohms
Resistance 3 value		
R4		56621 ohms
Resistance 4 value		
PlantTemperature1		17.87 °C
Temperature Plant 1		
MAC		0x286cfc56b5013ca6
MAC Address of DS18B20 Plant 1		
PlantSoilMoisture1		0 %
Measured soil moisture Plant 1		
PlantWaterAlimentation1		0 ms
Time of Water Quantity to delivery value Plant 1		
PlantWaterAlimentationState1		0
Current State of Water Alimentation Plant 1		

PlantTemperature2		20.93 °C
Temperature Plant 2		
MAC		0x28da1956b5013c1b
MAC Address of DS18B20 Plant 2		
PlantSoilMoisture2		82.95 %
Measured soil moisture Plant 2		
PlantWaterAlimentation2		0 ms
Time of Water Quantity to delivery value Plant 2		
PlantWaterAlimentationState2		0
Current State of Water Alimentation Plant 2		
PlantTemperature3		18 °C
Temperature Plant 3		
MAC		0x28e90156b5013ced
MAC Address of DS18B20 Plant 3		
PlantSoilMoisture3		0 %
Measured soil moisture Plant 3		
PlantWaterAlimentation3		0 ms
Time of Quantity to delivery value Plant 3		
PlantWaterAlimentationState3		0
Current State of Water Alimentation Plant 3		
PlantLight		0
State of Plant light		
Linkquality		216 lqi
Link quality (signal strength)		

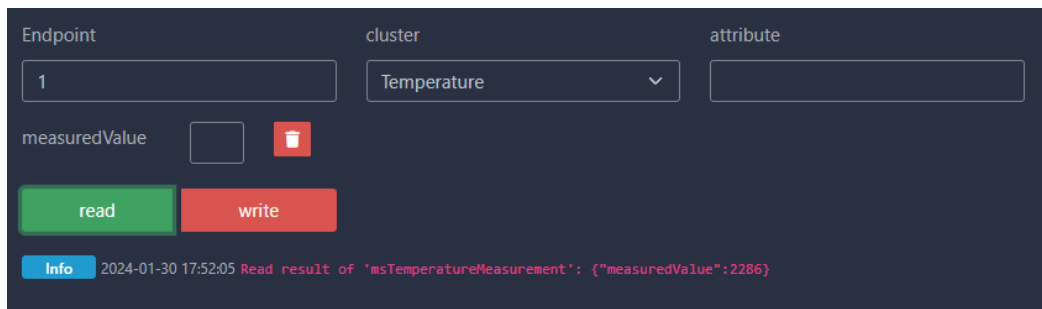
Clusters Menu



Dev ConsoleMenu

In this console you can read or write value of attributes

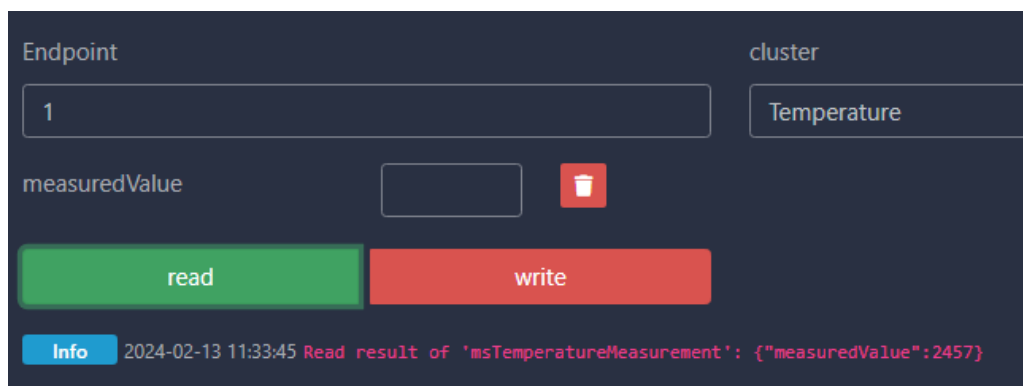
For example : Read attribute **Endpoint :1 Cluster :Temperature attribute : MeasuredValue**



The screenshot shows the Dev Console interface with the following fields and actions:

- Endpoint:** 1
- cluster:** Temperature
- attribute:** (empty)
- measuredValue:** (empty)
- Buttons:** read (green), write (red)
- Info:** 2024-01-30 17:52:05 Read result of 'msTemperatureMeasurement': {"measuredValue":2286}

If you click on Read



The screenshot shows the Dev Console interface with the following fields and actions:

- Endpoint:** 1
- cluster:** Temperature
- measuredValue:** (empty)
- Buttons:** read (green), write (red)
- Info:** 2024-02-13 11:33:45 Read result of 'msTemperatureMeasurement': {"measuredValue":2457}

The answer in console windows :

```
Zigbee2MQTT:debug 2024-02-13 11:33:45: Received MQTT message on 'zigbee2mqtt/STM32WB5MM-DK/1/set' with data '{"read":{"attributes":["measuredValue"],"cluster":"msTemperatureMeasurement","options":{}}}'
Zigbee2MQTT:debug 2024-02-13 11:33:45: Publishing 'set' 'read' to 'STM32WB5MM-DK'
Zigbee2MQTT:debug 2024-02-13 11:33:45: Received Zigbee message from 'STM32WB5MM-DK', type 'readResponse', cluster 'msTemperatureMeasurement', data '{"measuredValue":2457}' from endpoint 1 with groupID 0
Zigbee2MQTT:info 2024-02-13 11:33:45: Read result of 'msTemperatureMeasurement': {"measuredValue":2457}
Zigbee2MQTT:info 2024-02-13 11:33:45: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/temperature', payload '24.57'
Zigbee2MQTT:info 2024-02-13 11:33:45: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/linkquality', payload '212'
```

You can observe that :

- A mqtt request has sent to the MQTT Topic: **zigbee2mqtt/STM32WB5MM-DK/1/set**
- with JSON payload :
`{"read":{"attributes":["measuredValue"],"cluster":"msTemperatureMeasurement","options":{}}}`
- Zigbee2mqtt transform in : **Publishing 'set' 'read' to 'STM32WB5MM-DK'**
- Zigbee2mqtt receive :
Received Zigbee message from 'STM32WB5MM-DK', type 'readResponse', cluster 'msTemperatureMeasurement', data '{"measuredValue":2457}' from endpoint 1 with groupID 0
- Zigbee2mqtt send 2 topics for Mqtt server :
MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/temperature', payload '24.57'
MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/linkquality', payload '212'

You can Read multiples attributes :

Endpoint: 1, cluster: Temperature

measuredValue, Pressure, Humidity

read, write

Info 2024-02-13 11:43:52 Read result of 'msTemperatureMeasurement': {'measuredValue':2496,'Pressure':9940,'Humidity':5100}

1 read of 3 attributes generate 4 topics, one by attribute and linkquality attribute.

```
Zigbee2MQTT:debug 2024-02-13 11:43:52: Received MQTT message on 'zigbee2mqtt/STM32WB5MM-DK/1/set' with data '{"read":{"attributes":["measuredValue","Pressure","Humidity"],"cluster":"msTemperatureMeasurement","options":{}}}'
Zigbee2MQTT:debug 2024-02-13 11:43:52: Publishing 'set' 'read' to 'STM32WB5MM-DK'
Zigbee2MQTT:debug 2024-02-13 11:43:52: Received Zigbee message from 'STM32WB5MM-DK', type 'readResponse', cluster 'msTemperatureMeasurement', data '{"Humidity":5100,"Pressure":9940,"measuredValue":2496}' from endpoint 1 with groupID 0
Zigbee2MQTT:info 2024-02-13 11:43:52: Read result of 'msTemperatureMeasurement': {'measuredValue':2496,'Pressure':9940,'Humidity':5100}
Zigbee2MQTT:info 2024-02-13 11:43:52: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/temperature', payload '24.96'
Zigbee2MQTT:info 2024-02-13 11:43:52: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/Humidity', payload '51'
Zigbee2MQTT:info 2024-02-13 11:43:52: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/Pressure', payload '994'
Zigbee2MQTT:info 2024-02-13 11:43:52: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/linkquality', payload '224'
```

Writing attributes

In this example, i write 20 in DelayRefreshMeas, this attribute fix the delay in seconds between measures.

Endpoint: 1, cluster: Temperature

DelayRefreshMeas: 20

read, write

Info 2024-02-13 11:46:53 Wrote '{"DelayRefreshMeas":20}' to 'msTemperatureMeasurement'

```
Zigbee2MQTT:debug 2024-02-13 11:46:53: Received MQTT message on 'zigbee2mqtt/STM32WB5MM-DK/1/set' with data '{"write":{"cluster":"msTemperatureMeasurement","options":{},"payload":{"DelayRefreshMeas":20}}}'
Zigbee2MQTT:debug 2024-02-13 11:46:53: Publishing 'set' 'write' to 'STM32WB5MM-DK'
Zigbee2MQTT:info 2024-02-13 11:46:53: Wrote '{"DelayRefreshMeas":20}' to 'msTemperatureMeasurement'
```

You can observe that :

- A mqtt request has sent to the MQTT Topic: **zigbee2mqtt/STM32WB5MM-DK/1/set**
- with JSON payload :
`{"write":{"cluster":"msTemperatureMeasurement","options":{},"payload":{"DelayRefreshMeas":20}}}`
- Zigbee2mqtt transform in : **Publishing 'set' 'write' to 'STM32WB5MM-DK'**
- Zigbee2mqtt send to ZigBee network :
Wrote '{"DelayRefreshMeas":20}' to 'msTemperatureMeasurement'

Test reporting

The image shows two screenshots of a web interface for configuring reporting for a specific endpoint (1). The interface has tabs: About, Exposes, Bind, Reporting, Settings, Settings (specific), State, Clusters, Scene, and Dev console. The Reporting tab is active. It displays fields for Endpoint (1), Cluster (Temperature), Attribute (measuredValue), Min rep interval (30), max rep interval (360), and Min rep change (0). There is an 'Apply' button. The bottom screenshot shows the same interface but with a 'Disable' button added next to the 'Apply' button.

After apply

- A mqtt request has sent to the MQTT Topic:
zigbee2mqtt/bridge/request/device/configure_reporting
- with JSON payload :
`{"attribute":"measuredValue","cluster":"msTemperatureMeasurement","id":"STM32WB5MM-DK/1","maximum_report_interval":3600,"minimum_report_interval":60,"reportable_change":0,"transaction":"okbmi-8"}`

In response

- A mqtt request has sent to the MQTT Topic:
zigbee2mqtt/bridge/request/device/configure_reporting
- with JSON payload :
`'{"data":{"attribute":"measuredValue","cluster":"msTemperatureMeasurement","id":"STM32WB5MM-DK/1","maximum_report_interval":3600,"minimum_report_interval":60,"reportable_change":0},"status":"ok","transaction":"okbmi-8"}`

It is the same payload sent but "status" added with "ok" or "error".

Report every 30s

```
Zigbee2MQTT:debug 2024-02-13 12:03:04: Received Zigbee message from 'STM32WB5MM-DK', type 'attributeReport', cluster 'msTemperatureMeasurement', data '{"65534":1,"measuredValue":2528}' from endpoint 1 with groupID 0
Zigbee2MQTT:info 2024-02-13 12:03:04: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/temperature', payload '25.28'
Zigbee2MQTT:info 2024-02-13 12:03:04: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/linkquality', payload '216'
Zigbee2MQTT:debug 2024-02-13 12:04:04: Received Zigbee message from 'STM32WB5MM-DK', type 'attributeReport', cluster 'msTemperatureMeasurement', data '{"65534":1,"measuredValue":2530}' from endpoint 1 with groupID 0
Zigbee2MQTT:info 2024-02-13 12:04:04: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/temperature', payload '25.3'
Zigbee2MQTT:info 2024-02-13 12:04:04: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/linkquality', payload '216'
Zigbee2MQTT:debug 2024-02-13 12:05:04: Received Zigbee message from 'STM32WB5MM-DK', type 'attributeReport', cluster 'msTemperatureMeasurement', data '{"65534":1,"measuredValue":2530}' from endpoint 1 with groupID 0
Zigbee2MQTT:info 2024-02-13 12:05:04: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/temperature', payload '25.3'
Zigbee2MQTT:info 2024-02-13 12:05:04: MQTT publish: topic 'zigbee2mqtt/STM32WB5MM-DK/linkquality', payload '216'
```

CONCLUSION

It is very easy now to interact with the Zigbee Device using Mqtt Request.

The stm32wb55 processor is a good solution which correctly manages the Zigbee standard.

The next part is now to configure and use Node-RED application for automation, data storage and visualisation of informations.