

Easy and Wireless Home Automation



ZCL Library V4.3

User Manual

Révision 1.0

Document : UM_LIBZCL_20140115_001_01_00

GUARANTEE

The device supplied to the buyer and/or the recipient is guaranteed by CLEODE against any malfunctions originating from a design and/or manufacturing flaw, for a period of twelve (12) months following delivery. The buyer and/or recipient is (are) responsible for proving the existence of the said defects or flaws. This warranty is applicable in accordance with articles 1641 to 1648 of the French Civil Code and in compliance with the French statutory warranty.

The warranty covers the replacement free of charge of devices and parts affected by a design and/or manufacturing flaw excluding conspicuous defects in the device that are covered by the buyer and/or the recipient.

In order to invoke the warranty, the buyer must immediately send written notice to CLEODE of the flaws that it attributes to the device. It must enable CLEODE to have access to the device to observe these defects and repair them. The warranty provided by CLEODE is strictly limited to the equipment provided and shall only have for effect the replacement or repair, at CLEODE's expense, on its own premises, of all devices or parts that are not functioning as a result of defects or flaws. CLEODE reserves the right to modify the devices in order to comply with the warranty.

The warranty does not apply to replacement or repairs that may result from normal wear and tear of devices, systems or products, damage or accidents resulting from negligence, failure to supervise or maintain, or incorrect use of the devices, systems and/or products.

The maintenance service is provided by CLEODE with all reasonable care possible and in compliance with the current state of the arts. The exchange of parts or repairs performed under the warranty cannot result in extending the length of the warranty. In no event can the unavailability of the device due to servicing give rise to compensation for any reason whatsoever. The seller is released from all obligations relating to the warranty if the product or device has been modified without prior written consent, or if original parts have been replaced by parts which it has not manufactured without prior consent. If unforeseen damage is caused by the device, it is expressly agreed that the seller can only be liable for the reimbursement of monies received for the purchase of the device if it has been destroyed. Under no circumstances can the seller be held liable for indirect or contingent damage. The seller is released from any liability and the buyer waives following a defect, incorrect usage, incorrect maintenance or normal wear of the device sold.

SOMMAIRE

| | | |
|-------------|--|-----------|
| I | PRESENTATION | 6 |
| I.1 | FONCTIONNALITIES OF THE ZCL LIBRARY | 6 |
| I.2 | TECHNICAL CHARACTERISTICS | 6 |
| I.2.1 | DELIVERABLES PROVIDED | 6 |
| I.3 | FIRST USE | 7 |
| I.3.1 | EXAMPLE APPLICATION IN C++ | 7 |
| I.3.2 | IMPLEMENTATION | 7 |
| I.3.3 | EXAMPLE APPLICATION IN JAVA | 8 |
| II | FEATURES | 10 |
| II.1 | MANAGEMENT LIFE CYCLE OF ZCL API | 10 |
| II.1.1 | START ZCL API | 10 |
| II.1.2 | STOP THE ZCL API | 10 |
| II.1.3 | TEST IF THE UBEE IS STARTED | 10 |
| II.2 | EVENTS OF ZIGBEE OBJECTS | 11 |
| II.2.1 | RECORD OF THE « CALLBACK » | 11 |
| II.2.2 | DELETE THE « CALLBACK » | 11 |
| II.3 | MANAGEMENT OF THE ZIGBEE OBJECTS IN THE NETWORK | 11 |
| II.4 | NOTIFICATIONS OF THE ZIGBEE OBJECTS | 11 |
| II.5 | SEND/RECEIVE ZCL COMMAND | 12 |
| II.6 | SEND BINDING REQUESTS | 12 |
| II.7 | MANAGEMENT OF THE NETWORK PARAMETERS | 12 |
| II.8 | MANAGEMENT OF THE AUTHORIZED OBJECTS | 12 |
| II.9 | UTILITARY FUNCTIONS | 12 |

TABLEAU DES REVISIONS

| Version | Auteur(s) | Description de la version | Date |
|---------|-----------|---------------------------|------------|
| 1.0 | CLEODE | Initial version | 15/01/2014 |

GLOSSARY

| | |
|------------------|--|
| UBee Nano | Cleode Coordinator |
| ZPlug | Cleode Power Outlet |
| ZRC | Cleode Remote Control and temperature sensor |
| API | Application Programming Interface |
| JNI | Java Native Interface |
| PAN | Personal Area Network |

REFERENCE DOCUMENTS

| | | |
|----------|---------------------------------|----------------------------------|
| 1 | UM_UBEE_NANO_20120502_001_01_00 | User Manual of the UBee Nano Key |
|----------|---------------------------------|----------------------------------|

I PRESENTATION

I.1 FONCTIONNALITIES OF THE ZCL LIBRARY

The ZCL library is intended to facilitate the development of application that interfaces with the UBEE. Nano key It provides services for the following features:

- Managing the life cycle of the ZCL library.
- Reception of association events, of update objects Zigbee.
- Reading objects in the Zigbee network.
- Permission objects to append in the Zigbee network
- Subscription to receive notifications objects Zigbee.
- Sending and receiving ZCL commands.
- Creating and deleting bindings between objects Zigbee.

I.2 TECHNICAL CHARACTERISTICS

The ZCL library is a dynamic library written in C ++. It can be used in a multi-threaded. The user never has to release objects from the library.

She runs under the following operating systems:

- Windows XP and more
- Windows CE 6.0 and more
- Linux 2.6.19 and more

She also includes a connector to interface with the world via JNI Java for Windows XP and Linux.

I.2.1 DELIVERABLES PROVIDED

The ZCL library is provided as binary files, as well as all files "*.h" needed to compile, a user manual in PDF file, a source file example in C ++ (testZCL.cpp) and a source file example in java (TestZCL.java).

- Under Windows with msvc:
Files « zcl.dll » et « zcl.lib ».
- Under Windows with gcc
Files « zcl.dll » « zcl.a » and « zcl.def ».
- Under Linux
File « libzcl.so »

The delivery also includes a jar file "ZCL-connector.jar" for the java interface.

I.3 FIRST USE

I.3.1 EXAMPLE APPLICATION IN C++

The best way to implement the library ZCL is to compile and run the file "testZCL.cpp" supplied with the library. It helps to understand by example the use of the library ZCL.

This is a console application that shows a simple use of the library:

- Subscription to receive events Zigbee objects that are in the Zigbee network or that will appear later.
- Start the ZCL library and discover the network if a UBee™ key is detected into a USB port.
- List the objects present in the Zigbee network.
- Send the commands and display the ZCL responses.
- Display the help menu by typing 'h'.

I.3.2 IMPLEMENTATION

1) Install the drivers of the UBee™ key on Windows or Linux machine
Cf. document [1]. Contact CLEODE support in case of problem.

2) Compile the example file « testZCL.cpp »

- Under Windows with msvc
Copy the files zcl.dll and zcl.lib compiled with msvc in the test directory.

```
cl /I..\Includes /EHsc /TestZCL.exe testZCL.cpp zcl.lib
```

- Under Windows with gcc version < 4.0
Copy the file zcl.dll compiled with gcc in the test directory.

```
g++ -I..\Includes -O0 -Wall -L. -lzcl -o TestZCL.exe testZCL.cpp
```

- Under Windows with gcc version >= 4.0
Copy the file zcl.dll compile with gcc in the test directory.

```
g++ -I..\Includes -Wall -Wl,--enable-auto-import -o TestZCL.exe testZCL.cpp  
-L. -lzcl
```

Or

```
g++ -I..\Includes -Wall -static-libgcc -static-libstdc++  
-Wl,--enable-auto-import -o TestZCL.exe testZCL.cpp -L. -lzcl
```

- Under Linux with gcc

Copy the files libzcl.so and run.sh compiled with gcc in the test directory.

```
g++ -I..\Includes -O0 -Wall -o TestZCL testZCL.cpp -L. -lzcl -lpthread
```

3) Insert the UBee Nano Key in a USB Port of the computer.

- Under windows :
Verify in the « Device Manager (COM port and LPT) », the presence of the UBEE Nano key as a " USB Serial Port "Manufacturer" FDTI. "
- Under Linux :
Verify by the command « dmesg » that the key is associated with a ttyUSBx.

4) Start the console application

- Under windows : TestZCL.exe
- Under Linux : ./run.sh

If the UBEE Nano key is detected, the console will display:

```
Ubee Started
Tap 'q' to quit the program, 'h' for help
```

Otherwise, the console will display :

```
Ubee NOT Started
Tap 'q' to quit the program, 'h' for help
```

5) Add a ZPlug and a in the network ZRC to test all the commands ZCL.

I.3.3 EXAMPLE APPLICATION IN JAVA

The example in Java is a graphical application that shows the network zigbee objects in a table and the notifications received from these objects.

It is possible to change the description of location by double-clicking in the appropriate field and confirming with "Return".

A double-click on the field "model identification" of a ZPlug sends a command "Toggle" to change the state of ZPlug.

1. Install the drivers of the Ubee Nano key on Windows or Linux machine
Cf. document [1]. Contact CLEODE support in case of problem.
2. Compile the example file « TestZCL.java ».
 - Under Windows
Copy the files zcl-connector.jar and zcl.dll (compiled with msvc ou gcc) in the test directory.

```
javac -cp zcl-connector.jar TestZCL.java
```


- Under Linux

Copy the files `zcl-connector.jar` and `libzcl.so` in the test directory.

```
javac -cp zcl-connector.jar TestZCL.java
```

6) Insert the UBee Nano Key in a USB Port of the computer.

- Under windows :

Verify in the « Device Manager (COM port and LPT) », the presence of the UBEE Nano key as a " USB Serial Port "Manufacturer" FDTI. "

- Under Linux :

Verify by the command « `dmesg` » that the key is associated with a `ttyUSBx`.

7) Start the application

- Under windows :

```
java -cp .;zcl-connector.jar TestZCL
```

- Under Linux :

```
java -Djava.library.path=. -cp .:zcl-connector.jar TestZCL
```

II FEATURES

Full functionalities of the library are available as HTML documentation:

- <http://www.cleode.fr/cppdoc/index.html>
or
- <http://www.cleode.fr/javadoc/index.html>

This document describes a quick and non-exhaustive overview of its features.

II.1 MANAGEMENT LIFE CYCLE OF ZCL API

The ZCL library provides functions to manage the lifecycle of the ZCL Api :

- Start ZCL Api and try to start the UBEE,
- Stop the ZCL API and UBEE Nano ,
- Test if UBEE Nano is started.

II.1.1 START ZCL API

Resume :

Starts ZCL library i.e. opens a communication channel with the UBEE key if detected on a USB port and searches the objects in the Zigbee network.

Definition:

```
static bool zcl::startUbee
    (void*)(UbeeEvent event) pCallback, const char * portname = NULL)
```

II.1.2 STOP THE ZCL API

Resume :

Stops the ZCL library i.e. closes the communication channel with the UBee key if the library is in the started state and deletes the subscription to receive events Zigbee objects.

Definition:

```
bool stopUbee ();
```

II.1.3 TEST IF THE UBEE IS STARTED

Resume:

Test if the Ubee is started.

Definition:

```
bool isUbeeStarted ()
```

II.2 EVENTS OF ZIGBEE OBJECTS

The ZCL library offers the ability to subscribe at the events such as associating, updating and deleting of objects Zigbee.

A "Create" event is generated at the association of the object with the coordinator or ZigBee router or in the discovery phase to initialize the ZCL library.

A "Update" event is generated when an update information of a zigbee object.

A "Delete" event is generated when an object is no longer managed by the ZCL library.

II.2.1 RECORD OF THE « CALLBACK »

Resume:

Register a « callback » to receive the events such as associating, updating and deleting of the objets Zigbee.

To record a new « callback », it is necessary to delete the previous « callback » (cf. II.2.2).

Definition :

```
bool zcl::registerCallback
    (void*)(zcl::NodeEvent event, const ZNode &node, zcl::Reason) pCallback)
```

II.2.2 DELETE THE « CALLBACK »

Resume:

Delete the record of the « callback » to receive the events such as associating, updating and deleting of the objets Zigbee.

Definition:

```
void unregisterCallback ()
```

II.3 MANAGEMENT OF THE ZIGBEE OBJECTS IN THE NETWORK

The ZCL library provides functions to find the objects in the Zigbee network as :

- the number of zigbee objects in the network,
- the list of objects in the network,
- the search of a zigbee object in the network,
- Require a zigbee object to leave the network
- ...

II.4 NOTIFICATIONS OF THE ZIGBEE OBJECTS

The ZCL library provides the ability to receive notifications about a zigbee object by a subscription with the network address of this object. The notifications are :

- ReportAttributes
- ClusterSpecificCommand,

- ...

The ZCL library also offers the functionality to unsubscribe.

II.5 SEND/RECEIVE ZCL COMMAND

The ZCL library provides functions to send ZCL commands and wait for responses on a response receiver.

The functions are:

- To discover one or more attributes of an object,
- To read one or more attributes of an object,
- To write one or more attributes of an object,
- To read the notification configuration of one or more attributes of an object,
- To configure the notification of one or more attributes of an object,
- To send a specific command on an object,
- To send a ZCL frame to an object,
- ...

II.6 SEND BINDING REQUESTS

The ZCL library provides functions to issue queries for binding:

- To create a binding on a cluster between 2 objects
- To delete a binding on a cluster between 2 objects
- To create a binding between an object and a group,
- To delete a binding between an object and a group,
- To obtain the binding list of one or more objects
- ...

II.7 MANAGEMENT OF THE NETWORK PARAMETERS

The ZCL library manages the informations of the zigbee network such as the channel, the PanID.

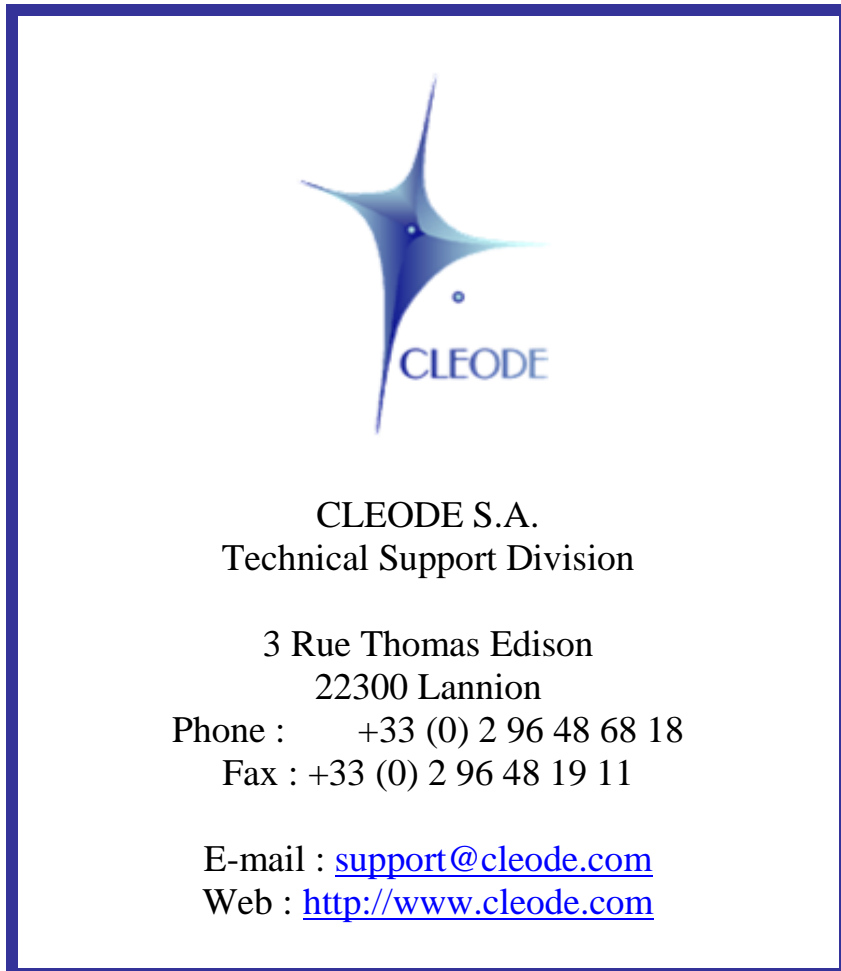
II.8 MANAGEMENT OF THE AUTHORIZED OBJECTS

The ZCL library can manage a list of network objects. This feature allows the return or not of an object in the network. When this list is not initialized, this function does not work.

II.9 UTILITY FUNCTIONS

The ZCL library provides utility functions such as get the version of the library, see traces of debug or modify the license directory, ...

All requests for support should be directed to the following postal address or e-mail to support@cleode.com.



End of User Manual