

La domotique facile et sans fil



Librairie ZCL V4.3

Manuel utilisateur

Révision 1.0

Document : MU_LIBZCL_20140115_001_01_00

GARANTIE

Le matériel fourni à l'acquéreur et/ou au destinataire est garanti par la société CLEODE contre tout défaut de fonctionnement provenant d'un risque de conception et/ou de fabrication, pendant une période de douze (12) mois après la livraison, à charge pour l'acquéreur et/ou le destinataire de prouver lesdits défauts ou vices. Cette garantie est appliquée conformément aux articles 1641 à 1648 du Code Civil et en respect de la garantie légale. La garantie couvre le remplacement gratuit des appareils et des pièces affectés d'un vice de conception ou d'un défaut de fabrication à l'exception des vices apparents du matériel couverts par l'acquéreur et/ou le destinataire.

Pour pouvoir invoquer le bénéfice de la garantie, l'acheteur doit aviser la société CLEODE par écrit et sans retard, des vices qu'il impute au matériel. Il doit donner à la société CLEODE toute facilité pour procéder à la constatation de ces vices et pour y porter remède. Les garanties de la société CLEODE sont strictement limitées au matériel fourni et ne doivent avoir pour effet que le remplacement ou la réparation à ses frais, dans ses ateliers, de toutes les pièces mises hors service par suite de défauts ou vices, la société CLEODE se réservant de modifier les dispositifs en vue de satisfaire à ces garanties.

La garantie ne s'applique pas aux remplacements ni aux réparations qui résulteraient de l'usure normale des appareils et machines, matériels et produits, de détériorations ou accidents provenant de négligence, défaut de surveillance ou d'entretien, ou d'utilisation défectueuse des appareils, machines, matériels et/ou produits.

Le service de maintenance est fourni par la société CLEODE avec tout le soin raisonnablement possible en l'état de la technique.

Les échanges de pièces ou réparations faites au titre de la garantie ne peuvent pas avoir pour effet d'en prolonger la durée. En aucun cas, l'immobilisation d'un appareil ne pourra donner droit à une indemnité pour quelque cause que ce soit. Le vendeur est dégagé de toute obligation relative à la garantie si des modifications sont apportées à la fourniture sans son consentement exprès, ou si des pièces étrangères à sa fabrication ont été substituées à son insu à des pièces d'origine. En cas de dommage imprévisible causé par le matériel vendu par le vendeur, il est expressément convenu que le vendeur ne pourra être tenu au maximum qu'au remboursement du prix perçu pour l'achat du matériel dans le cas où il serait détruit. En aucun cas le vendeur ne pourra être tenu pour responsable du préjudice indirect ou éventuel. Le vendeur se dégage de toute responsabilité et l'acheteur renonce à tout recours contre lui si un accident ou des dommages directs ou indirects sont causés à l'acheteur, suite à un défaut, à un mauvais usage, à un mauvais entretien ou à une usure normale du matériel vendu.

SOMMAIRE

I	PRÉSENTATION	6
I.1	FONCTIONNALITÉS DE LA LIBRAIRIE ZCL	6
I.2	CARACTÉRISTIQUES TECHNIQUES	6
I.2.1	LIVRABLES FOURNIS	6
I.3	PREMIÈRE UTILISATION	7
I.3.1	APPLICATION EXEMPLE EN C++	7
I.3.2	ETAPES DE MISE EN OEUVRE	7
I.3.3	APPLICATION EXEMPLE EN JAVA	9
II	DESCRIPTION DES FONCTIONNALITÉS	10
II.1	GESTION DU CYCLE DE VIE DE L'API ZCL	10
II.1.1	DÉMARRER L'API ZCL	10
II.1.2	STOPPER L'API ZCL	10
II.1.3	TESTER SI L'UBEE NANO EST DÉMARRÉE	10
II.2	ÉVÉNEMENTS CONCERNANT LES OBJETS ZIGBEE	11
II.2.1	ENREGISTREMENT DE LA « CALLBACK »	11
II.2.2	SUPPRESSION DE LA « CALLBACK »	11
II.3	GESTION DES OBJETS ZIGBEE PRÉSENTS DANS LE RÉSEAU	11
II.4	NOTIFICATION DES OBJETS ZIGBEE DU RÉSEAU	12
II.5	ÉMISSION/RÉCEPTION DE COMMANDE ZCL	12
II.6	ÉMISSION DE REQUÊTES DE BINDING	12
II.7	GESTION DES PARAMÈTRES RÉSEAU	12
II.8	GESTION DES OBJETS AUTORISÉS	13
II.9	FONCTION UTILITAIRES	13

TABLEAU DES REVISIONS

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

GLOSSAIRE

UBee Nano	Coordinateur Zigbee de Cleode
ZPlug	Prise de courant télécommandable Zigbee de Cleode
ZRC	Télécommande et capteur de température Zigbee de Cleode
API	Application Programming Interface
JNI	Java Native Interface
PAN	Personal Area Network

DOCUMENTS DE REFERENCE

1	MU_UBEE_NANO_20120502_001_01_00	Manuel utilisateur de la clef UBee Nano

I PRESENTATION

I.1 FONCTIONNALITES DE LA LIBRAIRIE ZCL

La librairie ZCL a pour but de faciliter le développement d'application qui s'interface avec la clef UBeep Nano. Elle propose des services pour les fonctionnalités suivantes :

- Gestion du cycle de vie de la librairie ZCL.
- Réception des événements d'association, de mise à jour des objets Zigbee.
- Lecture des objets Zigbee présents dans le réseau.
- Autorisation des objets Zigbee à rentrer dans le réseau
- Abonnement pour recevoir les notifications des objets Zigbee.
- Emission et réception de commandes ZCL.
- Création et suppression de bindings entre objets Zigbee.

I.2 CARACTERISTIQUES TECHNIQUES

La librairie ZCL est une librairie dynamique écrite en C++. Elle est utilisable dans un environnement multi-thread.

L'utilisateur n'a jamais à libérer des objets provenant de la librairie.

Elle fonctionne actuellement sous les systèmes d'exploitation suivants :

- Windows XP et plus
- Windows CE 6.0 et plus
- Linux 2.6.19 et plus

Elle intègre également un connecteur pour s'interfacer avec le monde Java via JNI pour les systèmes Windows XP et Linux.

I.2.1 LIVRABLES FOURNIS

La librairie ZCL est fournie sous la forme de fichiers binaires pour la librairie, ainsi que de l'ensemble des fichiers « *.h » nécessaire à la compilation, d'un manuel utilisateur au format PDF et d'un fichier source d'exemple en c++ (testZCL.cpp) et un fichier source d'exemple en java (TestZCL.java).

- Sous Windows avec msvc:
Fichiers « zcl.dll » et « zcl.lib ».
- Sous Windows avec gcc
Fichiers « zcl.dll » « zcl.a » et « zcl.def ».
- Sous Linux
Fichier « libzcl.so »

La livraison comprend également un fichier Jar « zcl-connector.jar » pour s'interfacer avec le monde Java.

I.3 PREMIERE UTILISATION

I.3.1 APPLICATION EXEMPLE EN C++

La meilleure façon de mettre en œuvre la librairie ZCL est de compiler et d'exécuter le fichier « testZCL.cpp » fourni avec la librairie. Il permet de comprendre par l'exemple l'utilisation de la librairie ZCL.

C'est une application console qui montre de façon simple l'utilisation de la librairie :

- S'enregistrer pour recevoir les événements des objets Zigbee qui sont déjà présents dans le réseau Zigbee ou qui apparaîtront ultérieurement.
- Démarrer la librairie ZCL et découvrir le réseau si une clef UBee Nano est détectée sur un port USB.
- Lister les objets Zigbee présents dans le réseau.
- Envoyer les différentes commandes ZCL et afficher leurs réponses.
- Afficher le menu d'aide en tapant 'h'.

I.3.2 ETAPES DE MISE EN OEUVRE

1) Installer les drivers de la clef UBee Nano sur la machine Windows ou Linux
Cf. document [1]. Contactez le support de CLEODE en cas de problème.

2) Compiler le fichier d'exemple « testZCL.cpp »

- Sous Windows avec msvc
Copier les fichiers zcl.dll et zcl.lib compilé avec msvc dans le répertoire de test.

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

- Sous Windows avec gcc version < 4.0
Copier le fichier zcl.dll compilé avec gcc dans le répertoire de test.

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

- Sous Windows avec gcc version >= 4.0
Copier le fichier zcl.dll compilé avec gcc dans le répertoire de test.

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

ou bien

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

- Sous Linux avec gcc
Copier les fichiers libzcl.so et run.sh compilé avec gcc dans le répertoire de test.

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

3) Insérer la clef UBee Nano dans un port USB de la machine.

- Sous windows :
Vérifier dans le « Gestionnaire de périphériques (Port COM et LPT) », la présence de la clef UBee Nano comme un « USB Serial Port » du fabricant « FDTI ».
- Sous Linux :
Vérifier par la commande « dmesg » que la clef est bien associée à un ttyUSBx.

4) Lancer l'application console

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

Si la clef UBee Nano est détectée, la console affichera :

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

Sinon, la console affichera :

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

5) Mettre en fonctionnement une ZPlug et une ZRC pour tester l'ensemble des commandes ZCL.

I.3.3 APPLICATION EXEMPLE EN JAVA

L'exemple en Java est une application graphique qui présente dans une table les objets Zigbee découverts sur le réseau et au-dessous les notifications reçues de ces objets Zigbee.

Il est possible de changer la description de localisation en double-cliquant dans le champ correspondant et en validant par la touche « Return ».

Un double-click sur le champ « model identifier » d'une ZPlug émet un commande « Toggle » qui permet de changer l'état de la ZPlug.

1. Installer les drivers de la clef UBee Nano sur la machine Windows ou Linux
Cf. document [1]. Contactez le support de CLEODE en cas de problème.
2. Compiler le fichier d'exemple « TestZCL.java ».
 - Sous Windows
Copier les fichiers zcl-connector.jar et zcl.dll (compilé avec msvc ou gcc) dans le répertoire de test.

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

- Sous Linux
Copier les fichiers zcl-connector.jar et libzcl.so dans le répertoire de test.

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

3. Insérer la clef UBee Nano dans un port USB de la machine.
 - Sous windows :
Vérifier dans le « Gestionnaire de périphériques (Port COM et LPT) », la présence de la clef UBee Nano comme un « USB Serial Port » du fabricant « FDTI ».
 - Sous Linux :
Vérifier par la commande « dmesg » que la clef est bien associée à un ttyUSBx.

4. Lancer l'application

- Sous windows :

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

- Sous Linux :

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

II DESCRIPTION DES FONCTIONNALITES

Les fonctionnalités complètes de la librairie sont disponibles sous forme de documentation HTML.

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

La suite de ce document permet seulement un aperçu rapide et non exhaustif de ses fonctionnalités.

II.1 GESTION DU CYCLE DE VIE DE L'API ZCL

La librairie ZCL fournit des fonctions pour gérer le cycle de vie de l'Api ZCL :

- Démarrer l'api ZCL et essayer de démarrer l'UBee Nano,
- Stopper l'api ZCL et l'UBee Nano,
- Tester si l'Ubee Nano est démarrée.

II.1.1 DEMARRER L'API ZCL

Résumé :

Démarré la librairie ZCL, c'est-à-dire ouvre un canal de communication avec la clef UBee Nano si elle est présente et détectée sur un port USB et démarre la découverte des objets Zigbee présent dans le réseau.

Définition :

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

II.1.2 STOPPER L'API ZCL

Résumé :

Stoppe la librairie ZCL, c'est-à-dire ferme le canal de communication avec la clef UBee Nano si la librairie est dans l'état démarré et supprime l'enregistrement de la « callback permettant de recevoir les événements des objets Zigbee.

Définition :

```
bool stopUbee ();
```

II.1.3 TESTER SI L'UBEE NANO EST DEMARREE

Résumé :

Retourne `true` si l'UBee est démarrée.

Définition :

```
bool isUbeeStarted ();
```

II.2 EVENEMENTS CONCERNANT LES OBJETS ZIGBEE

La librairie ZCL offre la possibilité d'enregistrer une « callback » (et une seule) pour recevoir les événements d'association, de mise à jour et de disparition des objets Zigbee.

Un événement « Create » est généré pour l'association d'un objet Zigbee au coordinateur ou à des routeurs du réseau ou dans la phase de découverte à l'initialisation de la librairie ZCL.

Un événement « Update » est généré lors d'une mise à jour des informations d'un objet Zigbee.

Un événement « Delete » est généré lorsqu'un objet Zigbee n'est plus géré par la librairie ZCL.

II.2.1 ENREGISTREMENT DE LA « CALLBACK »

Résumé :

Enregistre une « callback » pour recevoir les événements d'association, de mise à jour et de disparition des objets Zigbee.

Pour enregistrer une nouvelle « callback », il est nécessaire de supprimer la précédente « callback » (cf. II.2.2).

Définition :

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

II.2.2 SUPPRESSION DE LA « CALLBACK »

Résumé :

Supprime l'enregistrement de la « callback permettant de recevoir les événements d'association, de mise à jour et de disparition des objets Zigbee.

Définition :

```
void unregisterCallback()
```

II.3 GESTION DES OBJETS ZIGBEE PRESENTS DANS LE RESEAU

La librairie ZCL fournit des fonctions pour connaître les objets Zigbee présents dans le réseau comme :

- le nombre d'objets Zigbee présents dans le réseau,
- la liste des d'objets Zigbee présents dans le réseau
- la recherche un objet Zigbee présent dans le réseau par son adresse réseau,
- forcer un objet Zigbee à quitter le réseau,
- ...

II.4 NOTIFICATION DES OBJETS ZIGBEE DU RESEAU

La librairie ZCL offre la possibilité de recevoir des notifications concernant un objet Zigbee du réseau en s'abonnant par l'adresse réseau de cet objet. Les notifications sont de types :

- ReportAttributes
- ClusterSpecificCommand,
- ...

La librairie ZCL offre de plus la possibilité de désabonnement.

II.5 EMISSION/RECEPTION DE COMMANDE ZCL

La librairie ZCL fournit des fonctions pour émettre des commandes ZCL et attendre les réponses sur un récepteur de réponse.

Les commandes possibles sont :

- découvrir un ou plusieurs attributs d'un objet ;
- lire un ou plusieurs attributs d'un objet ;
- écrire un ou plusieurs attributs d'un objet,
- lire la configuration de notification d'un ou plusieurs attributs,
- configurer la notification d'un ou plusieurs attributs d'un objet,
- émettre une commande spécifique sur un objet,
- émettre une trame ZCL vers un objet,
- ...

II.6 EMISSION DE REQUETES DE BINDING

La librairie ZCL fournit des fonctions pour émettre des requêtes de binding pour :

- créer un binding sur un Cluster entre 2 objets,
- supprimer un binding sur un Cluster entre 2 objets,
- créer un binding entre un objet Zigbee et un groupe,
- supprimer le binding entre un objet Zigbee et un groupe,
- récupérer la liste des bindings d'un ou de plusieurs objets,
- ...

II.7 GESTION DES PARAMETRES RESEAU

La librairie ZCL permet de gérer les informations liées au réseau zigbee comme le canal, l'identifiant.

II.8 GESTION DES OBJETS AUTORISES

La librairie ZCL permet de gérer une liste d'objets d'un réseau. Cette fonctionnalité permet d'autoriser ou de refuser la rentrée d'un objet dans le réseau géré par l'Ubee. Lorsque cette liste n'est pas initialisée, cette fonctionnalité est inopérante.

II.9 FONCTION UTILITAIRES

La librairie ZCL fournit des fonctions utilitaires comme afficher la version de la librairie, afficher les traces de debug ou changer le répertoire de licence,

Toute demande de support doit être adressée à l'adresse postale suivante ou par courrier électronique à support@cleode.com :



CLEODE S.A.
Service Support et Maintenance

3 Rue Thomas Edison
22300 Lannion
Tél : +33 (0) 2 96 48 68 18
Fax : +33 (0) 2 96 48 19 11

E-mail : support@cleode.com
Web : <http://www.cleode.com>

Fin du manuel utilisateur