

Bluetooth Low Energy

Alasdair Allan, Don Coleman et Sandeep Mistry

Bluetooth Low Energy

Projets pour Arduino, Raspberry Pi et smartphones

Traduit de l'anglais par Gérard Samblancat

DUNOD

Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.

Le Code de la propriété intellectuelle du 1^{er} juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements

d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour

les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée. Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du

Centre français d'exploitation du droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).



© Dunod, 2017

11 rue Paul Bert, 92240 Malakoff

www.dunod.com

ISBN 978-2-10-076085-5

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

TABLE DES MATIÈRES

PRÉFACE	IX
Mais moi, j'aime aussi les liaisons série!	X
Construire un service fait maison	X
À qui s'adresse ce livre ?	XI
Ce que vous devriez déjà savoir	XI
Ce que vous allez apprendre	XI
Ce que vous trouverez dans ce livre	XI
Conventions	XII
Utilisation des programmes d'exemple	XII
Pour nous contacter	XIII
REMERCIEMENTS	XV
/>1 INTRODUCTION	1
Quelques concepts généraux sur le BLE	1
Protocoles et profils	2
UUID	4
Comment vérifier qu'un appareil est compatible Bluetooth LE	5
Que ne vous a-t-on pas encore dit sur le Bluetooth LE ?	7

/>2	BIEN DÉBUTER	9
	Arduino	9
	Brancher la carte	14
	Installation de la librairie BLE	15
	Utiliser Raspberry Pi	16
	Node.js	17
	Installer les librairies avec npm	18
	Régler les références aux librairies pour noble et bleno	19
	Installer PhoneGap	19
/>3	UN COMMUTATEUR D'ÉCLAIRAGE INTELLIGENT	23
	Qu'est-ce qu'un commutateur intelligent ?	23
	Le montage	23
	Démarrer la construction	24
	Un anti-rebond logiciel	33
	Construire un vrai interrupteur d'éclairage	33
	Ajouter le Bluetooth	36
	Conclusion	47
/>4	UNE SERRURE BLE	49
	Le service serrure	49
	Le montage	49
	Lock Software	52
	Initialisation	53
	Test de la serrure	59
	Une application mobile	61
	Améliorer la serrure	77
/>5	UNE SERRURE BLENO	79
	Le matériel	79
	Le logiciel	83
	Conclusion	90
/>6	UNE STATION MÉTÉO	91
	Le montage	92
	Les librairies utilisées	93
	Programmation	94
	Compilation et programmation	99
	Le moniteur série	99
	Utilisation du service	100

	Utilisation de nRF Master Control Panel sur Android	101
	PhoneGap	102
	Créer le Projet	102
	Démarrer l'application	114
	Pour aller plus loin	115
/>7	UNE LAMPE NEOPIXEL	117
	Le matériel	117
	Le logiciel	120
	Réaliser une application pour smartphone	130
	Créer le projet	130
	Améliorations	140
/>8	UNE TÉLÉCOMMANDE À SENSORTAG	155
	Le matériel	155
	Création du projet	157
	SensorTag et noble	157
	Une télécommande à SensorTag	164
	Une version simplifiée	167
	Étape suivante	168
	Le service Simple Key sur Arduino	168
/>9	HID SUR GATT	173
	HOGP et BLEPeripheral	173
	Un réglage de volume	174
	Conclusion	183
/>10	BEACONS	185
	Le matériel	185
	IBeacons	186
	Les beacons Eddystone et le web des objets	200
	Conclusion	205
/>11	DRONES	207
	Ce dont vous aurez besoin	207
	Premier test du drone	208
	Piloter votre Rolling Spider avec Node.js	208
	Conclusion	219

/>12 POUR ALLER PLUS LOIN	221
Sur Arduino	221
Quelques remarques matérielles	221
Autres ouvrages	224
/>13 APPAIRAGE D'APPAREILS HID SUR GATT	225
Appairage sur iOS	225
Appairage sur Android	228
Appairage sur OS X	232
INDEX	235
À PROPOS DES AUTEURS	237



PRÉFACE

Le principal moteur de l'explosion récente de ce qu'on appelle l'Internet des objets est certainement la technologie Bluetooth LE (ou BLE). Ce qui la rend si attirante est le fait d'être supportée par la majorité des smartphones, ce qui facilite bien le travail des concepteurs dans la fabrication de nouvelles applications. Vous pourrez pleinement profiter de la puissance et de l'interface de vos smartphones interagir avec votre environnement grâce à la technologie Bluetooth LE (basse consommation). Malgré la ressemblance du nom, le BLE est très différent de sa version classique. Vous verrez ici le plus important, et apprendrez comment construire vos propres appareils connectés. Sa très faible consommation et sa meilleure connectivité sur de plus grandes distances en font un outil parfait pour les concepteurs.

Avec ce livre, vous apprendrez à écrire des programmes et à utiliser des capteurs du commerce pour créer vos propres applications BLE ! En utilisant des modules peu onéreux comme ceux de la gamme AdafruitBluefruit LE, ce livre vous apportera de solides connaissances, tout en vous guidant pas à pas à travers plusieurs montages. C'est le complément idéal pour ceux qui ont déjà une expérience de la programmation sur Arduino ou Raspberry Pi, qui leur permettra d'étendre largement leurs possibilités.

Dans ce livre vous allez :

- Construire un interrupteur d'éclairage contrôlable à distance.
- Créer une serrure que seul votre smartphone pourra ouvrir.
- Construire une station météo capable de mesurer température, humidité et pression.
- Fabriquer une lampe à Leds NeoPixel dont on pourra changer la couleur.
- Piloter à distance le défilement d'images sur votre ordinateur avec un badge du type SensorTag.
- Faire un module Bluetooth pour ajuster le volume de votre PC ou de votre smartphone.
- Piloter un drone du type « Rolling spider » (Parrot) en utilisant votre PC et Node.js.

Ce livre vous montrera comment travailler avec la technologie Bluetooth à tous les niveaux de votre projet – carte microcontrôleur, smartphone, tablette, et PC. Vous pourrez faire communiquer tous ces composants entre eux, en utilisant des logiciels comme Arduino, Node.js ou PhoneGap; et vous pourrez aussi utiliser toute sorte de capteurs et composants externes.

Vous préférez peut-être voir les capteurs et les composants périphériques comme les serveurs d'une classique relation client/serveur. Les cartes contrôleurs sont alors les clients de la liaison Bluetooth LE, ils sont à l'écoute des informations issues des périphériques extérieurs.

MAIS MOI, J'AIME AUSSI LES LIAISONS SÉRIE !

La plupart, pour ne pas dire tous les modules Bluetooth LE disponibles actuellement pour les concepteurs (RedBearLabBLE mini et AdafruitBluefruit LE en l'occurrence) peuvent dans un souci de simplicité, s'utiliser comme des composants à accès série disposant d'une interface du type UART (avec file d'attente asynchrone et liaison bit par bit). En effet, ces modules radio imitent presque parfaitement les anciens ports série.

La couche Serial over Bluetooth LE rend le passage à la nouvelle version Bluetooth LE plus facile, surtout pour les gens qui ont déjà utilisé le protocole SPP (*Serial Port Profile*). Mais son principal inconvénient est de transmettre les données brutes et en vrac. Il faudra encore lui adjoindre un protocole d'encapsulation des données.

Le Bluetooth LE permet aux concepteurs de construire des applications génériques capables de se déclarer toutes seules sur le réseau. Si leurs données et leurs descripteurs sont correctement définis, vous pourrez utiliser ses services sans documentation (comme avec l'ampoule Smartbotic dans l'application LightBlue iOS).

Cela diffère complètement d'autres composants SPP, pour lesquels il faut d'abord bien connaître tous les détails du protocole afin de pouvoir modifier le bon octet qui permettra d'allumer la LED ou de changer sa couleur. Ces modules nécessitent d'avoir une vraie documentation, et même des bibliothèques spécifiques pour faire des choses aussi simples qu'allumer une LED.

Dans notre futur exemple de tableau d'affichage, nous allons créer un tableau (le service), sur lequel on pourra imaginer un post-it. Ce sera ce qu'on appelle une `characteristic` dans le langage du Bluetooth LE. On pourra ensuite lire ou venir modifier sa valeur pour savoir si la LED est allumée ou non.

CONSTRUIRE UN SERVICE FAIT MAISON

Il y a encore peu de temps, la conception d'un service Bluetooth LE était réellement difficile pour le commun des mortels. Mais on trouve maintenant des outils de plus en plus simples et efficaces capables de faire le plus gros du travail.

C'est pourquoi nous avons décidé de nous intéresser à une platine radio (le module de chez NordicSemiconductor) et de présenter un ensemble complet d'outils qui vous permettront d'utiliser les liaisons radios pour vos propres applications Arduino. Nous avons choisi ces modules au vu de leur bonne disponibilité, et du choix de librairies déjà disponibles.

Ce livre en est le résultat.

À QUI S'ADRESSE CE LIVRE ?

Vous trouverez dans cet ouvrage les informations nécessaires pour construire et utiliser des composants et capteurs Bluetooth LE. Il ne s'agit pas seulement d'une description formelle des protocoles utilisés mais d'une aide pratique et concrète.

CE QUE VOUS DEVRIEZ DÉJÀ SAVOIR

Ce livre se veut une introduction à l'utilisation des modules Bluetooth LE. Il nécessite bien sûr quelques connaissances techniques, mais il vous guidera pas à pas, à travers toutes les étapes: de l'installation à l'utilisation de tous les logiciels utiles; y compris la prise en main de Arduino, Raspberry Pi, Node.js, et PhoneGap.

CE QUE VOUS ALLEZ APPRENDRE

Ce livre vous guidera dans la construction d'une série d'objets connectés: lampe, serrure, drone... Après avoir construit vos premiers prototypes, vous verrez comment les améliorer et apprendre à en faire des composants connectés grâce au BLE.

CE QUE VOUS TROUVEREZ DANS CE LIVRE

Chapitre 1 – Introduction

Ce chapitre vous présentera les principaux standards et concepts qu'il vous faudra connaître avant de commencer à travailler avec les composants BLE.

Chapitre 2 – Bien débiter

Vous verrez ici comment installer les outils logiciels nécessaires au développement.

Chapitre 3 – Un commutateur d'éclairage intelligent

Ce chapitre vous montrera comment construire un commutateur d'éclairage intelligent qui vous permettra de piloter la lumière à distance grâce au BLE. Ce commutateur sera également capable de vous envoyer une notification en cas de changement d'état.

Chapitre 4 – Une serrure BLE

Vous apprendrez ici à fabriquer une serrure qui ne pourra être ouverte que par votre smartphone toujours par Bluetooth LE. Pour cela, vous verrez

aussi comment écrire l'application nécessaire (pour iOS ou Android) en utilisant le langage PhoneGap.

Chapitre 5 – Serrure version Bleno

Ce chapitre revisite la précédente serrure, mais cette fois en utilisant Node.js (d'où le nom de Bleno) sur une platine Raspberry Pi. Le service BLE sera le même que pour la première version, et l'application de commande sera donc toujours compatible.

Chapitre 6 – Une station météo

Vous y trouverez les détails de la construction d'une station météo avec liaison Bluetooth LE, capable de mesurer température, humidité et pression.

Chapitre 7 – Une lampe NeoPixel

Ce chapitre abordera la fabrication d'une lampe RGB (multicolore) télécommandée depuis votre smartphone, en utilisant une carte Arduino et une lampe du type NeoPixel à 16 leds.

Chapitre 8 – Une Télécommande à SensorTag

Vous apprendrez comment transformez un petit module SensorTag (de Texas Instruments) en télécommande pour déclencher par exemple un processus sur votre PC lorsqu'un bouton est appuyé.

Chapitre 9 – Un composant HID over GATT

La norme GATT (*General Attribute Profiles*) définit un certain nombre d'appareils types, utilisables avec Bluetooth LE. Vous apprendrez à construire un appareil du type *Human Interface Device* (un périphérique d'entrée typique comme un clavier) pour faire une télécommande de volume.

Chapitre 10 – Beacons

Ce chapitre est dédié à l'utilisation de ce nouveau type de balises. Vous verrez comment en créer une avec Node.js, et comment les utiliser avec un smartphone.

Chapitre 11 – Drones

Apprenez à piloter un drone du type «Rolling Spider» (Parrot) avec votre liaison BLE et Node.js sur votre PC.

Chapitre 12 – Pour aller plus loin

Vous y trouverez un ensemble de pistes vers d'autres composants avancés, en rapport avec les chapitres précédents.

CONVENTIONS

L'abréviation BLE signifie Bluetooth Low Energy.

UTILISATION DES PROGRAMMES D'EXEMPLE

Ce livre est fait pour vous aider à atteindre vos objectifs. De manière générale, vous pourrez utiliser les exemples de code fournis à l'intérieur, dans vos propres applications. Vous n'avez pas besoin de demander une autorisation si

vous n'utilisez que quelques morceaux de programme. Par contre, si vous comptez vendre ou distribuer plusieurs exemples tirés de ce livre, vous aurez besoin d'une autorisation. Répondre à une question en citant juste un exemple du livre est autorisé. Recopier plusieurs exemples de codes tirés du livre, pour les incorporer dans votre projet nécessite une autorisation.

Tous les exemples du livre sont également disponibles sur GitHub (<https://github.com/MakeBluetooth>).

Même si ce n'est pas obligatoire, nous apprécions les citations (titre du livre, auteurs, collection... par exemple « Copyright 2016 Alasdair Allan et Don Coleman, Dunod 978-2-10-076085-5 »).

Si vous pensez que l'utilisation que vous faites de ce livre nécessite une autorisation particulière, n'hésitez pas à écrire à bookpermissions@makemedia.com.

POUR NOUS CONTACTER

Pour adresser à l'éditeur toutes vos questions et remarques concernant ce livre :

Make:
1160 Battery Street East, suite 125
San Francisco, CA 94111
Tél. : 707-639-1355 (à l'international).

Nous avons créé une page Web pour ce livre. Vous y trouverez toute une liste d'erratas, d'exemples et de nouvelles informations. L'adresse est <http://bit.ly/make-BT>.

Le réseau Make inspire, informe, et rassemble une communauté grandissante de gens pleins de ressources capables de réaliser des projets incroyables dans leur cave ou leur garage. Make vous permet et vous encourage à customiser et à détourner selon vos souhaits toute forme de technologie.

Les lecteurs de Make croient toujours qu'un certain progrès est possible pour nous, notre environnement, notre éducation, ou le monde entier. Ce ne sont plus de simples lecteurs, mais un mouvement mondial que constituent les makers.

Pour plus d'informations en ligne sur Make :

- Make : magazine (<http://makezine.com/magazine>)
- Maker Faire (<http://makerfaire.com>)
- Makezine.com (<http://makezine.com>)
- Maker Shed (<http://makershed.com>)

Pour tout commentaire ou question technique sur cet ouvrage, envoyez vos messages par mail à bookquestions@oreilly.com.



REMERCIEMENTS

REMERCIEMENTS D'ALASDAIR ALLAN

Ce livre est déjà mon neuvième ouvrage, mais ce n'est pas seulement le mien. Car même si tous les livres sont différents, on ne les écrit jamais vraiment tout seul. C'est pourquoi je tiens à remercier mes co-auteurs Don Coleman et Sandeep Mistry, ainsi que mes éditeurs de chez Make : Brian Jepson et Roger Stewart pour m'avoir toujours accompagné.

REMERCIEMENTS DE DON COLEMAN

J'aimerais remercier ma femme, Meghan, mon fils Liam pour leur support et leur patience, pour toutes ces heures que j'ai passées à travailler sur ce livre. Tom Igoe m'a beaucoup aidé avec ses idées, et toutes les premières versions de codes qu'il a pu tester avec ses étudiants. Le travail de Guan Yang sur le module Arduino nRF8001 a été un solide point de départ pour utiliser le Bluetooth LE sur Arduino sans avoir besoin d'outil propriétaire. Brian Jepson, Roger Stewart et l'équipe de Make méritent d'être salués pour leur patience et leur obstination à publier ce livre. J'ai également apprécié l'aide de mes collègues de Chariot Solutions LLC. Plus récemment, ça a été génial de travailler avec mes co-auteurs Alasdair et Sandeep.

REMERCIEMENTS DE SANDEEP MISTRY

Ce livre est le premier ouvrage auquel j'ai pu participer. Aussi j'aimerais remercier mes co-auteurs Alasdair Allan et Don Coleman, ainsi que mes éditeurs Brian Jepson et Roger Stewart qui m'ont guidé tout au long de la rédaction grâce à leurs retours éclairés.



CHAPITRE 1

INTRODUCTION

Au deuxième chapitre, nous verrons comment installer les outils nécessaires à l'écriture de nos programmes et au fonctionnement des services à distance. Mais avant de commencer, parlons un peu de la technologie BLE et du vocabulaire technique qui va avec. Vous pourrez passer ensuite au troisième chapitre qui constituera déjà une solide introduction à l'utilisation de BLE et Arduino. Après cela, vous serez en mesure d'attaquer tous les autres projets décrits dans ce livre.

QUELQUES CONCEPTS GÉNÉRAUX SUR LE BLE

Le BLE distingue toujours deux sortes de composants : les périphériques simples ou « esclaves », et les composants « maîtres » (ou central). Les composants périphériques sont souvent des capteurs, de petites dimensions et faible consommation. Les composants maîtres sont le plus souvent des smartphones, ou des tablettes. Ils peuvent également fonctionner en tant que périphérique.

Les périphériques esclaves ont deux modes de fonctionnement : soit en mode diffusion, soit en liaison directe vers un composant maître. Le mode diffusion (*broadcast*) constitue une des principales différences entre le BLE et le Bluetooth classique ; il permet à un « esclave » d'envoyer des données à destination de n'importe quel périphérique assez proche.

CONSEIL

La portée réelle d'un composant BLE dépend de la puissance du module radio. Mais comme une forte puissance nécessite davantage de batteries, le BLE a plutôt une portée réduite. Même s'il est parfaitement possible d'avoir une portée de 30 mètres, on va le plus souvent de 2 à 5 mètres.

Cela signifie qu'un périphérique BLE esclave n'a pas forcément besoin d'être appairé avec le maître pour pouvoir commencer à transmettre. Avec le BLE nous

parlerons plutôt de périphériques « connectés » et plus « appairés » comme avec le Bluetooth 2.1. En mode diffusion ou *broadcast*, les périphériques envoient périodiquement des messages de signalement (*advertising packets*) à destination des autres composants qui observent le réseau et pourraient en avoir besoin.

Un « message de signalement » décrit le périphérique qui l'émet ainsi que ses caractéristiques. Mais il peut aussi contenir des informations personnalisées, telles que des valeurs physiques mesurées.

La diffusion de données depuis un périphérique esclave est parfaite pour, par exemple, une station météo dont les données ne sont pas sensibles. Mais ce mode de diffusion ne propose aucune confidentialité. Pour transmettre des données plus personnelles, il faudra utiliser une connexion spécifique entre les composants maîtres et esclaves.

En effet, les connexions sont privées et un périphérique esclave ne peut être connecté qu'à un seul maître à la fois à un instant donné. Lorsqu'un maître se connecte à un esclave, celui-ci arrête d'émettre ses messages de signalement. Les autres composants ne peuvent alors ni le voir, ni demander à s'y connecter, tant que la première connexion n'est pas terminée. Si un esclave ne peut avoir qu'un seul maître, un maître peut par contre avoir plusieurs esclaves à la fois.

CONSEIL

La norme Bluetooth 4.1 a supprimé la limitation fixant qu'un esclave ne peut être connecté qu'à un seul maître. On peut maintenant aller plus loin et un esclave peut être connecté à plusieurs maîtres simultanément. Hélas, nombreux sont les chipsets encore limités à une seule connexion.

Si vous voulez échanger des données entre composants maître et esclave, vous devrez de toute façon établir une connexion.

PROTOCOLES ET PROFILES

Au-dessus de tous les protocoles qui constituent le BLE, la norme définit ce qu'elle appelle des profils. Il y a des profils standards utilisés par tous les composants BLE comme les *Generic Access Profile* et *Generic Attribute Profile*. Et il y a des profils dédiés à des usages spécifiques comme le *Heart Rate Profile* (pour les capteurs de rythme cardiaque).

/ LE GAP

Le *Generic Access Profile* (GAP) définit les rôles des composants maîtres et esclaves, comme celui des messages de signalement, ou la découverte du réseau.

Il y a deux méthodes de signalement dans le GAP : l'envoi de messages de signalement, ou l'envoi d'une réponse au scan du réseau (*scan response packets*).

Alors que ces deux types de messages ont le même format composé de 31 octets de données. Seuls les messages de signalement sont obligatoires et envoyés à intervalles réguliers (plus la période est longue, plus la consommation est faible). Quand ils les reçoivent les autres composants peuvent demander l'émission d'un message de réponse au scan (*scan response packet*) avec d'éventuelles données supplémentaires.

La diffusion de messages de signalement personnalisés est la méthode utilisée par l'iBeacon et la norme Eddystone que vous retrouverez au chapitre 10.

À chaque fois qu'une connexion sera établie avec un composant esclave, les liaisons se feront grâce à un service GATT avec ses caractéristiques. Les messages de signalement cesseront tant que la connexion ne sera pas terminée.

/ LE GATT

Le *Generic Attribute Profile* définit comment le BLE transfère les données entre maîtres et esclaves dans les deux sens. Il définit des profils qui regroupent un ensemble de services. Chaque service a des caractéristiques qui correspondent aux données.

Les modes de fonctionnement peuvent changer quand on passe du GAP au GATT. Le GATT dispose de deux modes de fonctionnement : client ou serveur.

Au risque de faire vieux jeu, les périphériques esclaves sont appelés « serveurs GATT » et les maîtres (plus puissants) sont des « clients GATT ». Souvenez-vous : le serveur détient les données, le client veut les données. Toutes les connexions sont initiées par les clients.

À chaque nouvelle connexion, un client peut demander une liste des services fournis par le serveur. Le maître détient en effet en permanence une liste des services (pas forcément exhaustive) complétée grâce aux messages de signalement.

/ SERVICES ET CARACTÉRISTIQUES

Les services permettent d'encapsuler les données en unités logiques correspondant aux différentes valeurs ou caractéristiques. Les caractéristiques sont comme des variables qui contiennent les données associées à chaque service. Les deux entités sont identifiées par un unique identifiant que l'on appelle UUID. Une caractéristique est définie par son nom et par sa valeur effective (la donnée réelle).

Les caractéristiques peuvent aussi contenir des descripteurs pour des paramètres supplémentaires. Chaque ensemble formé par une déclaration, une valeur et d'éventuels paramètres supplémentaires forme une caractéristique.

Les caractéristiques peuvent être définies en mode lecture seule, ou en écriture (*read* ou *write*). Un client déclenche une lecture en faisant d'abord une demande de lecture, avant de se voir retourner la valeur de la caractéristique.

Les valeurs des caractéristiques sont écrites au moyen d'une demande d'écriture (*write request*). Lorsqu'une écriture est faite, le serveur renvoie une confirmation. Il y a encore une autre méthode : la commande d'écriture (*write command*)

qui force une écriture, sans confirmation en retour. C'est une écriture sans réponse, ni confirmation.

Il existe encore deux autres méthodes: la notification (*notify*) et l'indication (*indicate*). Toutes les deux sont initiées par le serveur lorsqu'une valeur change. Un client qui s'inscrit à l'avance sera ainsi prévenu et recevra une notification avec la nouvelle valeur de la caractéristique. L'indication fonctionne de la même façon, sauf que le client doit en accuser réception.

Chaque caractéristique peut utiliser plusieurs méthodes, comme par exemple `read`, `write` et `notify`.

UUID

Dans presque tous les services, le Bluetooth utilise des identifiants uniques (*Universally Unique Identifiers* ou UUID). Les services qui sont officiellement approuvés par le Bluetooth Special Interest Group ont tous un numéro unique (UUID) codé sur 16 bits. Tous les autres services ainsi que les opérations sur les caractéristiques doivent utiliser un UUID codé sur 128 bits. Celui-ci pourra être généré par des outils comme `uuidgen`, comme ci-après:

```
> $ uuidgen  
437121E5-A6F8-43F9-8F8F-4AB73D6CC3EB
```

Par commodité, vous ne verrez dans ce livre que des UUID 16 bits. Cela ne respecte pas tout à fait la norme, mais dans les exemples qui suivent, il sera plus facile d'écrire DCF8 que 391CDCF8-4BD4-4507-BE23-B57DFD1F870B. Le Tableau 1.1 donne la liste des familles d'UUID 16 bits officiellement approuvés.

UUID	USAGE
00xx	Descripteurs de nom
18xx	Services
27xx	Unités
28xx	Déclarations
29xx	Descripteurs
2Axx	Caractéristiques

Tableau 1.1 Liste officielle des familles d'UUID 16bits reconnus

C'est ainsi que dans les exemples utilisant des LED nous réutiliserons le service utilisé par la lampe RoboSmart de chez SmartBotics. Pour les boutons poussoirs, nous réutiliserons le service Simple Key de Texas Instruments. Pour les thermomètres par contre nous créerons des UUID 16 bits.

C'est une bonne solution d'utiliser les UUID des services courants qui correspondent à vos besoins. Mais si vous préférez en créer de nouveaux, il vaut mieux qu'ils soient codés sur 128 bits. Allez donc voir sur le web le site Bluetooth Developer pour plus d'informations (<http://developer.bluetooth.org>).