



**Bluetooth<sup>®</sup> Low Energy**

## Table des matières

Bluetooth Low Energy versus Bluetooth Classic .....	3
Architecture du Bluetooth Low Energy .....	4
Couche Application (Application Layer) : .....	5
Couche Gestionnaire (Host Layer) : .....	5
Couche Contrôleur (Controller Layer) : .....	5
Les profils .....	6
Les services .....	7
Objectif : Fournir des informations sur le niveau de batterie du périphérique. ....	7
Caractéristiques : Niveau de batterie actuel, seuils d'alerte de batterie faible. ....	7
2. Service de Fréquence Cardiaque (Heart Rate Service) : .....	7
Objectif : Mesurer et transmettre la fréquence cardiaque d'un utilisateur. ....	7
Caractéristiques : Mesure de la fréquence cardiaque, support pour différentes zones de fréquence cardiaque. ....	7
3. Service d'Alerte d'Urgence (Immediate Alert Service) : .....	7
Objectif : Permettre au périphérique de déclencher des alertes d'urgence. ....	7
Caractéristiques : Niveau d'alerte (critique, grave, etc.), déclenchement d'alertes sonores ou visuelles.....	7
4. Service de Données d'Environnement (Environmental Sensing Service) : .....	7
Objectif : Mesurer des paramètres environnementaux tels que la température, l'humidité, etc. .....	7
Caractéristiques : Température actuelle, humidité relative, qualité de l'air. ....	7
5. Service de Positionnement (Location and Navigation Service) : .....	7
Objectif : Fournir des informations de positionnement et de navigation. ....	7
Caractéristiques : Coordonnées GPS, vitesse, direction. ....	7
Les caractéristiques .....	9
Les indications et notifications .....	10
Exemple concret avec un data logger de la marque Efento .....	11
1. Déchiffrement du cadre de publicité (advertisement frame) : .....	11
2. Déchiffrement du cadre de réponse de balayage (scan response frame) : .....	12

## Un Aperçu de la Technologie Bluetooth Low Energy

Le Bluetooth Low Energy (BLE) est une technologie de communication sans fil conçue pour les appareils à faible consommation d'énergie, tels que les appareils portables, les capteurs, les dispositifs médicaux et les objets connectés. Il a été introduit pour répondre à la demande croissante d'une connectivité sans fil économe en énergie pour des applications variées. BLE est une norme de communication sans fil qui permet aux appareils de se connecter et d'échanger des données de manière efficace, en minimisant la consommation d'énergie, ce qui en fait un choix idéal pour de nombreuses applications IoT (Internet of Things).

## La Nécessité de la Technologie Bluetooth Low Energy

Bluetooth Low Energy a été motivée par la nécessité de fournir une solution de communication sans fil économe en énergie pour les appareils alimentés par batterie. Les technologies de communication sans fil précédentes, telles que le Bluetooth classique, étaient souvent trop gourmandes en énergie pour de tels appareils, ce qui limitait leur durée de vie de la batterie et leur convivialité.

Avec l'essor de l'Internet des objets (IoT) et des appareils portables, il est devenu essentiel de disposer d'une technologie sans fil capable de maintenir des connexions durables tout en conservant la durée de vie de la batterie des appareils. BLE a répondu à cette exigence en offrant une communication sans fil fiable et efficace tout en réduisant considérablement la consommation d'énergie, ce qui en fait un choix incontournable pour de nombreuses applications sans fil.

## Bluetooth Low Energy versus Bluetooth Classic

Bluetooth Low Energy (BLE) diffère considérablement du Bluetooth classique en termes de consommation d'énergie, de portée et d'applications. Alors que le Bluetooth classique est principalement destiné aux applications nécessitant une bande passante élevée, comme la diffusion audio, BLE se concentre sur les applications nécessitant une faible consommation d'énergie et une transmission de données plus légères.

Les principales différences entre BLE et le Bluetooth classique incluent :

- Consommation d'énergie réduite : BLE est conçu pour minimiser la consommation d'énergie, ce qui le rend idéal pour les appareils alimentés par batterie.
- Portée réduite : BLE a une portée plus courte que le Bluetooth classique, ce qui le rend adapté aux connexions de proximité.
- Débit de données plus faible : BLE offre un débit de données plus faible que le Bluetooth classique, ce qui convient aux applications de transfert de données légères.
- Temps de connexion rapide : BLE permet des connexions rapides et de courte durée, ce qui économise de l'énergie.

## Architecture du Bluetooth Low Energy

L'architecture du Bluetooth Low Energy (BLE) est conçue de manière à minimiser la consommation d'énergie tout en permettant une communication efficace entre les appareils.

Les spécifications techniques du BLE :

Figure 1 - Core technical specification

Technical specifications	Bluetooth Low Energy	Bluetooth Classic
Power consumption	Rated power Consumption of 0.01 to 0.5 W	Rated power consumption of 1 W
Data rate and throughput	Physical data rate is 1 MBit/s with an effective application data throughput of 0.3 MBit/s	Physical data rate is 1-3 MBit/s with an effective application data throughput of 2.1 MBit/s

Latency (from a non-connected state)	6 ms	100 ms
Voice capable	No	Yes
Distance/range (theoretical max.)	>100 m	100 m
Pairing mandatory	No	Yes
Frequency (GHz)	2.4	2.4
Active slaves	Up to 7	Undefined and implementation dependent
Security	128-bit AES and application layer user defined	56/128-bit and application layer user defined
Network topology	Point-to-point and Star	Piconet, scatternet, and point-to-point
Frequency channels	40	79
Minimum total time to send data (det. battery life)	3 ms	100 ms
Current consumption	<15 mA	<30 mA
Logo		

Son architecture est généralement divisée en trois couches : l'application (Application layer), le gestionnaire (Host layer), et le contrôleur (Controller layer) :

<sup>1</sup> IoT Projects with Bluetooth Low Energy p. 14-15

### Couche Application (Application Layer) :

Cette couche est la plus haute dans l'architecture BLE et elle représente l'interface avec les applications utilisateur. Elle gère les profils et services applicatifs spécifiques à l'application qui définissent les fonctionnalités que le périphérique BLE peut fournir. Les profils standard comme le Generic Attribute Profile (GATT) et le Generic Access Profile (GAP) sont souvent utilisés dans cette couche.

### Couche Gestionnaire (Host Layer) :

Cette couche est responsable de la gestion des protocoles et des procédures qui facilitent la communication BLE entre les périphériques. Elle comprend plusieurs sous-couches, notamment :

- Protocole de l'Application (Attribute Protocol - ATT) : Gère les transferts de données entre les périphériques BLE.
- Profil d'Attribut Générique (Generic Attribute Profile - GATT) : Définit la structure des données et des procédures pour l'interaction entre les périphériques.
- Profil d'Accès Générique (Generic Access Profile - GAP) : Gère les aspects fondamentaux de la connexion, comme la découverte de périphériques et l'établissement de connexions.
- L2CAP (Logical Link Control and Adaptation Protocol) : Fournit un multiplexage et la segmentation des paquets de données pour le transport.

### Couche Contrôleur (Controller Layer) :

Cette couche est responsable de la gestion de la couche physique et de la liaison radio. Elle contrôle les opérations liées à la radio et gère la transmission et la réception des paquets BLE. Les aspects tels que la modulation, la fréquence, la puissance de transmission, etc., sont gérés à ce niveau.

L'architecture de BLE repose sur le principe de l'événement et de l'échange de données asynchrones, ce qui permet de maintenir une faible consommation d'énergie. Les connexions BLE sont établies et gérées de manière à minimiser l'utilisation de la batterie des appareils.

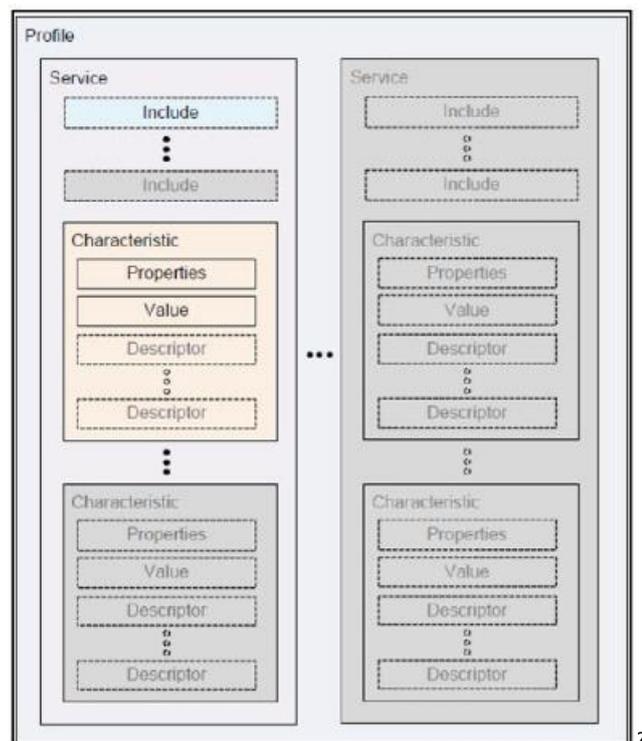
## Les profils

Les profils BLE sont des spécifications définissant la manière dont les appareils BLE interagissent dans un domaine d'application particulier. Ils décrivent les services et les caractéristiques qui doivent être pris en charge par les périphériques et les centraux pour assurer une communication cohérente. Les profils sont essentiels pour garantir que les appareils BLE de différents fabricants peuvent fonctionner ensemble de manière uniforme.

Certains exemples de profils BLE incluent le profil de santé (pour les dispositifs médicaux), le profil de surveillance (pour les montres connectées et les bracelets d'activité), et le profil de proximité (pour les appareils de suivi d'objets).

Les profils BLE simplifient le processus de développement d'applications en définissant des règles standardisées pour la communication, ce qui permet aux développeurs de créer des applications compatibles avec un large éventail d'appareils BLE.

Voici la représentation schématique d'un profil :



<sup>2</sup> IoT Projects with Bluetooth Low Energy (p.21)

## Les services

Les services dans le contexte de BLE sont des ensembles de caractéristiques liées qui définissent un domaine d'application spécifique. Les services contiennent des informations sur la manière dont les données doivent être organisées, lues, et échangées. Chaque service est identifié par un UUID (Universally Unique Identifier) qui le rend reconnaissable.

Ces services sont conçus pour simplifier la communication entre les périphériques en définissant des caractéristiques et des opérations standardisées. Chaque service a une fonction particulière et contient des caractéristiques qui représentent des données spécifiques ou des capacités du périphérique.

Exemples concrets de Services :

1. Service de Batterie (Battery Service) :

Objectif : Fournir des informations sur le niveau de batterie du périphérique.

Caractéristiques : Niveau de batterie actuel, seuils d'alerte de batterie faible.

2. Service de Fréquence Cardiaque (Heart Rate Service) :

Objectif : Mesurer et transmettre la fréquence cardiaque d'un utilisateur.

Caractéristiques : Mesure de la fréquence cardiaque, support pour différentes zones de fréquence cardiaque.

3. Service d'Alerte d'Urgence (Immediate Alert Service) :

Objectif : Permettre au périphérique de déclencher des alertes d'urgence.

Caractéristiques : Niveau d'alerte (critique, grave, etc.), déclenchement d'alertes sonores ou visuelles.

4. Service de Données d'Environnement (Environmental Sensing Service) :

Objectif : Mesurer des paramètres environnementaux tels que la température, l'humidité, etc.

Caractéristiques : Température actuelle, humidité relative, qualité de l'air.

5. Service de Positionnement (Location and Navigation Service) :

Objectif : Fournir des informations de positionnement et de navigation.

Caractéristiques : Coordonnées GPS, vitesse, direction.

Voici la représentation d'un service :

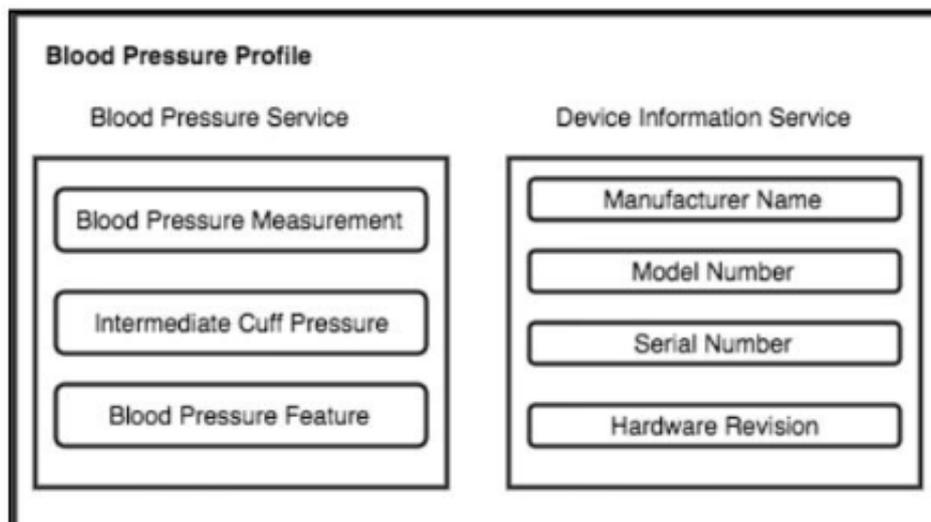


Figure 13: Services in a Blood Pressure Profile

## Les caractéristiques

Les caractéristiques sont des éléments de données au sein d'un service. Elles contiennent des informations spécifiques liées à ce service. Chaque caractéristique est identifiée par un UUID unique qui permet de la distinguer des autres caractéristiques au sein du même service.

Les caractéristiques peuvent être lues, écrites ou mises à jour par le client pour échanger des données avec le périphérique (serveur). Par exemple, une caractéristique au sein d'un service de suivi d'activité peut stocker la fréquence cardiaque actuelle.

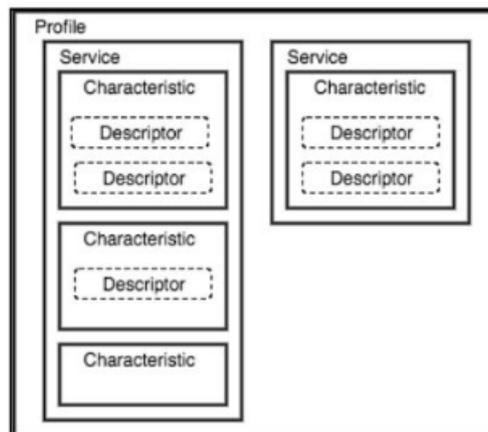


Figure 14: Relationship of Characteristics, Service, and Profile

4

Voici une image qui détail la structure d'une caractéristique :

Each characteristic as a whole comprises the following:

- **Characteristic declaration:** The characteristic declaration is an important part of a characteristic as it contains the UUID and properties of a characteristic. The properties of a characteristic are 8-bits lined together, which determine how the value of the characteristic can be used and how the descriptors can be accessed. You should already have seen the properties associated with each characteristic of the Blood Pressure Profile service by visiting the preceding link. We discuss the relevance of each of these properties as follows:
  - **Read:** If this bit is set, then it means that clients are allowed to read this characteristic's value.
  - **Write:** If this bit is set, then it means that clients are allowed to write (and receive a response) to this characteristic's value.
  - **Write without response:** If this bit is set, then it means that clients are allowed to write (without response) to this characteristic's value.
  - **Signed Write:** If this bit is set, then it means that clients are allowed to do a signed write to this characteristic's value.
  - **Notify:** One of the important ones. If set then the server will asynchronously notify the client whenever the value of the characteristic gets updated on the server. We will discuss this more in the next section. Also, if set, then the client configuration descriptor will exist. We shall discuss descriptors in detail shortly.
  - **Indicate:** Similar to notify, the only difference is that an indication requires an acknowledgement from the client. We will discuss this more in the next section. Also, if set, then the client configuration descriptor will exist. We shall discuss descriptors in detail shortly.
  - **Write auxiliaries:** If set, then the client can write to the characteristic user description descriptor.
  - **Broadcast:** If this bit is set, then it means that the value of this characteristic will be broadcasted, that is, placed in advertising packets.
  - **Extended properties:** Is set, then additional properties are defined in the characteristic extended properties descriptor, which also means that the characteristic extended properties descriptor shall exist. We shall discuss descriptors in detail shortly.

5

## Les indications et notifications

Les indications et les notifications sont des mécanismes utilisés pour informer le client de modifications importantes au sein des caractéristiques. Ils sont particulièrement utiles lorsque le client doit être averti de changements en temps réel. Les indications nécessitent une confirmation de réception par le client, tandis que les notifications n'en nécessitent pas.

Ces mécanismes sont couramment utilisés dans des applications telles que les moniteurs de fréquence cardiaque, les dispositifs de suivi en temps réel, ou les alertes d'événements. Par exemple, un capteur de température peut utiliser les notifications pour signaler un changement de température à l'application du client dès qu'il se produit.

---

<sup>5</sup> IoT Projects with Bluetooth Low Energy (p.26)

Ces éléments de communication (services, caractéristiques, indications et notifications) permettent une interaction flexible et efficace entre les appareils BLE, ce qui les rend adaptés à une variété d'applications, de la surveillance de la santé aux dispositifs de suivi d'objets.

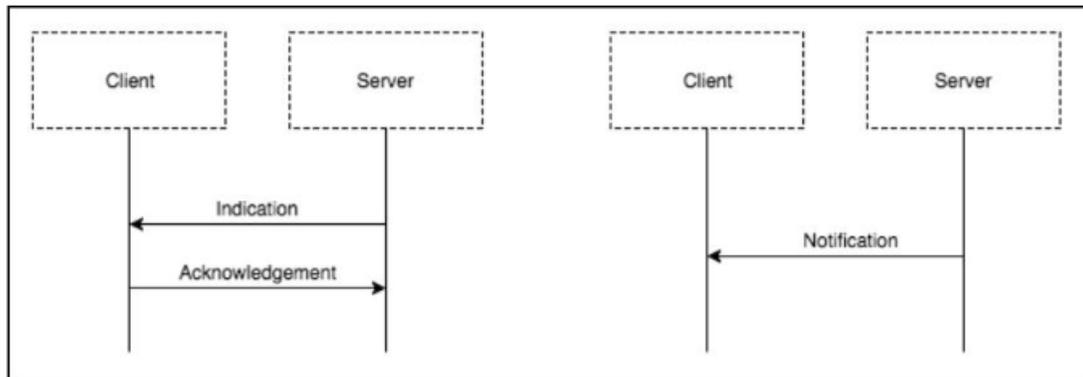


Figure 15: Indications versus notifications

6

### Exemple concret avec un data logger de la marque Efento

Ces éléments sont repris et traduits de la documentation officielle d'Efento.<sup>7</sup>

Le processus pour récupérer les valeurs des capteurs d'un data logger Efento utilisant Bluetooth Low Energy (BLE) implique de décoder à la fois le cadre de publicité (advertisement frame) et le cadre de réponse de balayage (scan response frame) pour obtenir les informations complètes sur l'état du capteur et ses mesures actuelles. Voici comment cela fonctionne :

#### 1. Déchiffrement du cadre de publicité (advertisement frame) :

Byte 1-2 : Le fabricant ID, "6C-02", identifie le fabricant comme Efento.

Byte 3 : La version des données de fabrication est définie à "03" pour les cadres de publicité BLE.

Byte 4-9 : Le numéro de série de l'appareil (BLE MAC address) est contenu ici.

Byte 10-11 : La version du firmware est indiquée sous forme de bits. Les bits 0-4 représentent la version à long terme (LTS), les bits 5-10 la version mineure, et les bits 11-15 la version majeure.

Byte 12 : Le byte de statut comprend plusieurs informations telles que le niveau de batterie, l'état de l'alimentation externe, le statut de chiffrement, la synchronisation de l'heure, les erreurs d'exécution et la connexion cellulaire.

Byte 13-16 : Timestamp de la mesure au format POSIX.

<sup>6</sup> IoT Projects with Bluetooth Low Energy (p.29)

<sup>7</sup> <https://getefento.com/library/efento-bluetooth-low-energy-sensors-reading-advertising-data/>

Byte 17-18 : Période de base de la mesure.

Byte 19-20 : Facteur de la période de mesure.

Byte 21-22 : Date de calibration (optionnelle).

Byte 23-24 : Checksum CRC16.

## 2. Déchiffrement du cadre de réponse de balayage (scan response frame) :

Byte 1-2 : Le fabricant ID, "6C-02", identifie le fabricant comme Efento.

Byte 3 : La version des données de fabrication est définie à "04" pour les cadres de réponse de balayage BLE.

Les octets suivants (4, 8, 12, 16, 20, 24) indiquent le type de mesure et la valeur de la mesure, le cas échéant. Les types de mesures possibles sont listés, avec des exemples, dans la documentation.

Pour récupérer les valeurs de mesure, le gateway doit donc décoder les informations contenues dans les deux cadres. Cela permet d'obtenir à la fois les informations sur le capteur, telles que la version du firmware, l'état de la batterie et la synchronisation de l'heure, ainsi que les mesures spécifiques prises par le capteur, telles que la température, l'humidité, la pression atmosphérique, etc.

Le processus consiste donc à extraire les données pertinentes de ces cadres en utilisant les spécifications de décodage fournies, puis à les stocker ou à les traiter selon les besoins de l'application. Cela permet au gateway de récupérer et de surveiller les mesures du capteur et de maintenir une communication efficace avec le data logger Efento.

## Bibliographie

Bhargava, M. (2017). *IoT Projects with Bluetooth Low Energy*. Packt.

Efento. (s.d.). *efento-bluetooth-low-energy-sensors-reading-advertising-data*. Récupéré sur <https://getefento.com/>: <https://getefento.com/library/efento-bluetooth-low-energy-sensors-reading-advertising-data/>