**Le BLE (Bluetooth Low Energy)**

[](https://blog.groupe-sii.com/wp-content/uploads/2015/11/image007.png)

**1. Introduction**

En guise de préambule, il convient de préciser que la technologie **Bluetooth Low Energy** est également appelée Bluetooth LE, BLE ou encore plus couramment **Bluetooth Smart**.

Cette technologie est apparue en 2010 avec la sortie de la version 4.0 du ***Bluetooth Core Specification***. Le Bluetooth Low Energy est une alternative au Bluetooth classique, avec une réduction du coût et de la consommation en puissance, tout en conservant une portée de communication équivalente. Ce qu’il faut entendre par « Bluetooth classique » sont toutes les versions du Bluetooth parues avant la *Core Specification 4.0*, autrement appelées Bluetooth BR/EDR (*Basic Rate / Enhanced Data Rate*).

Le Bluetooth Low Energie cherche à adresser des appareils à faible puissance de calcul, à faible coût de production et dont l’espérance de vie est à maximiser. L’idée est donc de réduire leur temps d’activité par rapport à des appareils basés sur le Bluetooth classique. Les performances brutes telles que le début de transmission en seront donc tout naturellement affectées.

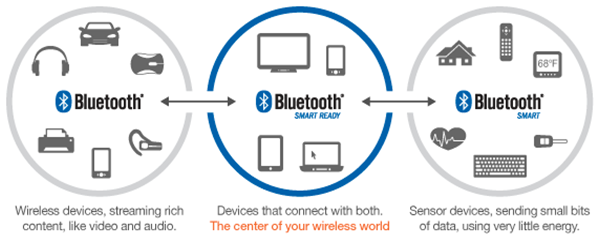
**Ce qu’apporte la Bluetooth Core Specification 4.0…**

Le changement apparaît avec les appareils dotés de cette nouvelle version de Bluetooth.

* Tous les appareils basés sur le Bluetooth classique continuent de fonctionner tel qu’ils pouvaient le faire avant mais ces appareils ne sont en revanche pas compatible avec des appareils Bluetooth Low Energy.
* Les appareils Bluetooth Low Energy ne sont pas compatibles avec les appareils basés sur le Bluetooth classique. Il n’y a donc aucune rétrocompatibilité avec le Bluetooth Low Energy. Les appareils Bluetooth Low Energy sont ainsi appelé également *single-mode*.
* Un nouveau type d’appareils, appelé **Bluetooth Smart Ready**, permet de faire le lien entre ces deux univers. Un tel appareil est capable de communiqué avec des appareils basés sur le Bluetooth classique et avec des appareils Bluetooth Low Energy. Ces appareils sont ainsi également appelés *dual-mode*.

Ces compatibilités sont illustrées sur la figure ci-dessous :

|  |  |  |
| --- | --- | --- |
| **Technical Specification** | **Classic Bluetooth** | **Bluetooth Low Energy** |
| Distance/Range | 100 m (330 ft) | 50 m (160 ft) |
| Over the air data rate | 1–3 Mbit/s | 1 Mbit/s |
| Application throughput | 0.7–2.1 Mbit/s | 0.27 Mbit/s |
| Active slaves | 7 | Not defined; implementation dependent |
| Security | 56/128-bit and application layer user defined | 128-bit AES with Counter Mode CBC-MAC and application layer user defined |
| Robustness | Adaptive fast frequency hopping, FEC, fast ACK | Adaptive frequency hopping, Lazy Acknowledgement, 24-bit CRC, 32-bit Message Integrity Check |
| Latency (from a non-connected state) | Typically 100 ms | 6 ms |
| Total time to send data (det.battery life) | 100 ms | 3 ms, <3 ms |
| Voice capable | Yes | No |
| Network topology | Scatter-net | Star-bus |
| Power consumption | 1 as the reference | 0.01 to 0.5 (use case dependent) |
| Peak current consumption | <30 mA | <15 mA |

[](https://blog.groupe-sii.com/wp-content/uploads/2015/11/BLE-2-Compatibilites.png)

*Figure 1 : Bluetooth 4.0 avec les appareils Bluetooth Smart Ready et Bluetooth Smart*

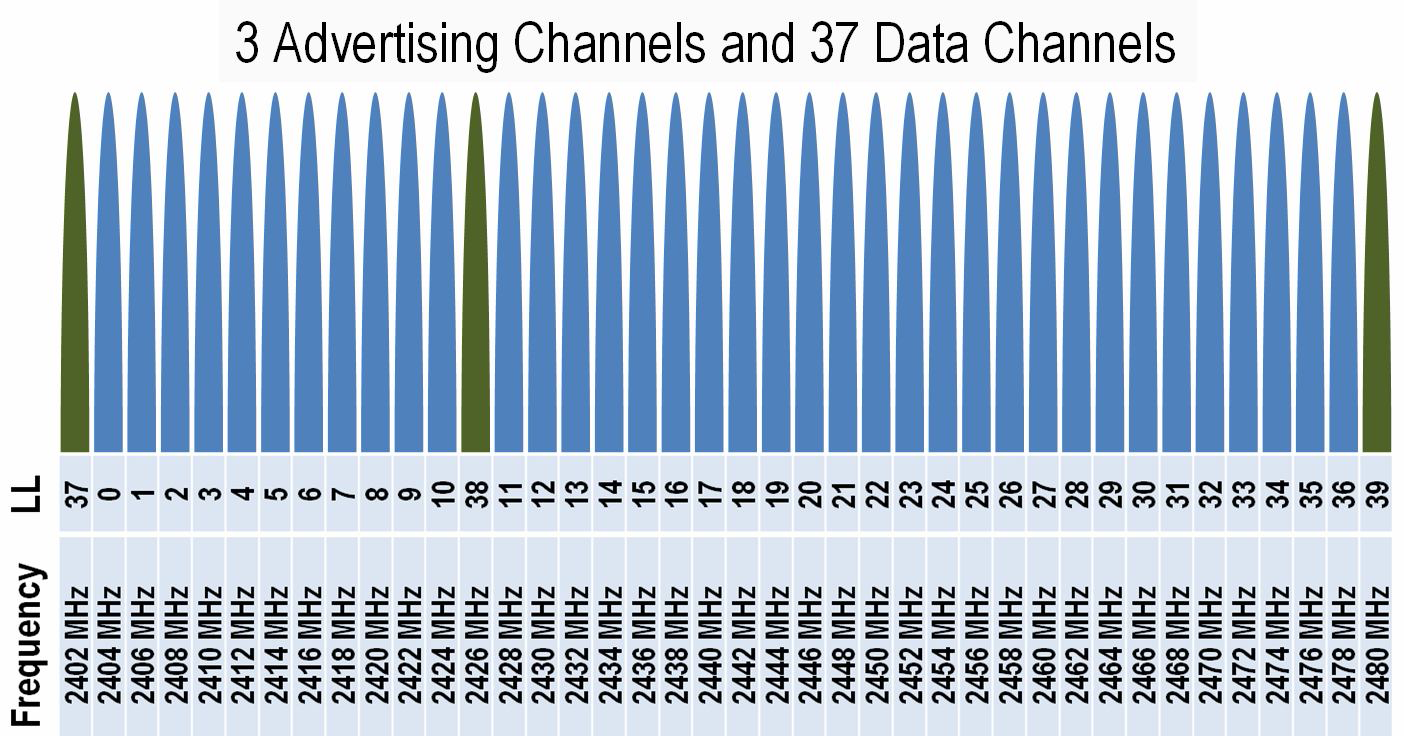
La technologie radio Bluetooth Low Energy

Le tableau ci-dessous illustre quelques différences de performance entre le Bluetooth classique et le Bluetooth Low Energy.

*Table 1: Comparaisons entre le Bluetooth classique et le Bluetooth Low Energy*

La technologie Bluetooth Low Energy opère dans la bande libre ISM 2.4 GHz (usage Industriel, Scientifique, Médical). Cette technologie est basée sur une radio à saut de fréquence pour combattre l’interférence et les évanouissements fréquentiels et fournit de nombreuses porteuses FHSS (*Frequency Hopping Spread Spectrum*). Une modulation simple (*Gaussian Frequency Shift Keying* – GFSK) est choisir pour réduire la complexité de la radio.

Deux modes d’accès sont utilisés : *Frequency Division Multiple Access* (FDMA) et *Time Division Multiple Access* (TDMA) pour un multiplexage en temps et en fréquence. 40 canaux physiques sont alloués ; chacun d’eux espacés de 2 MHz et utilisés selon le FDMA. Trois d’entre eux sont des canaux dits d’*avertisement* (on peut voir cela comme de la signalisation) et tous les autres sont des canaux de données. En revanche, le Bluetooth classique utilise 80 canaux séparés de 1 MHz.

[](https://blog.groupe-sii.com/wp-content/uploads/2015/11/image009.png)

*Figure 2: Distribution des canaux Bluetooth Low Energy*

Un mode TDMA est adopté pour synchroniser les appareils et ainsi les faire communiquer ensemble à des instants bien précis. La ressource physique qu’est l’accès à une fréquence radio est subdivisée en unités de temps appelé ***events***. Les données sont transmises dans des trames positionnées dans ces *events*. Les *events* peuvent être soit des ***advertising events***, soit des ***connection events***.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RF Channel | RF Center Frequency | Channel Type | Data Channel Index | Advertising Channel Index |
| 0 | 2402 MHz | Advertising channel |  | 37 |
| 1 | 2404 MHz | Data channel | 0 |  |
| 2 | 2406 MHz | Data channel | 1 |  |
| … | … | Data channel | … |  |
| 11 | 2424 MHz | Data channel | 10 |  |
| 12 | 2426 MHz | Advertising channel |  | 38 |
| 13 | 2428 MHz | Data channel | 11 |  |
| 14 | 2430 MHz | Data channel | 12 |  |
| … | … | Data channel | … |  |
| 38 | 2478 MHz | Data channel | 36 |  |
| 39 | 2480 MHz | Advertising channel |  | 39 |

*Table 2: Index des canaux du Bluetooth Low Energy*

**2. Vue protocolaire du Bluetooth Low Energy**

Cette section a pour but de présenter succinctement la technologie Bluetooth Low Energy avec sa vue par couche protocolaire. Les couches les plus pertinentes seront détaillées.

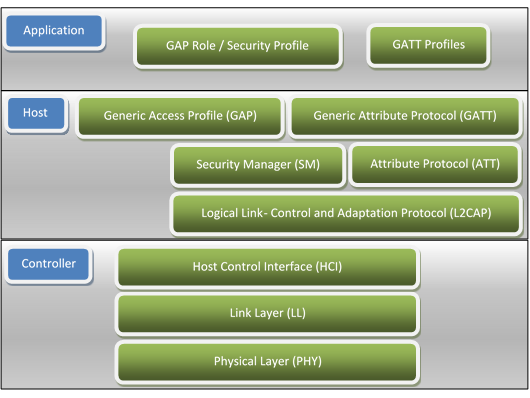
La pile protocolaire Bluetooth a toujours été séparée en deux parties : *Controller* et *Host*.

* Le Controller est en charge des basses couches protocolaires qui s’occupent de l’accès au canal radio avec tous les ajustements nécessaires (synchronisation fréquence et temps), de l’émission et de la réception des trames. Le Controller est souvent un microcontrôleur doté d’une radio Bluetooth.
* Le Host est en charge des hautes couches protocolaires qui aboutissent sur l’application et la définition des profils qui indiquent l’usage que va faire l’appareil du Bluetooth Low Energy. Le Host est en général un microprocesseur doté d’une plus grande puissance de calcul et entièrement contrôlé par l’application.
* Le Host et le Controller communiquent au travers de l’interface HCI (Hardware Controller Interface) via une communication série.

Toutefois, on peut trouver des implémentations qui ne comportent qu’un seul microprocesseur doté d’une radio Bluetooth et dans ce cas la couche HCI n’a plus d’intérêt.

Le *Host* gère tous les aspects liés à la sécurisation du lien au niveau de la couche *Security Manager* (SM). Le support de toutes les données exposées par un appareil est géré par la couche *Attribute Protocol* (ATT). Les couches *Generic Access Profile* (GAP) et *Generic Attribute Protocol* (GATT) sont fondamentales.

* La couche GAP est responsable de l’établissement du lien et de la supervisation de connexion entre deux appareils.
* La couche GATT quant à elle apporte un ensemble de sous-procédures basées sur la couche ATT pour permettre d’orchestrer toute la gestion des données supportées par un appareil (voir section GATT ci-après).

[](https://blog.groupe-sii.com/wp-content/uploads/2015/11/BLE-3-Couches-protocolaires.png)

*Figure 3: Les couches protocolaires Bluetooth Smart pour des appareils single-mode*

**Différents concepts fondamentaux**

* La couche Link Layer (LL) introduit deux modes opposés : **maître** et **esclave**.
  + Le maître est responsable de la synchronisation entre les deux appareils connectés. Il définit donc le pattern de saut de fréquence ainsi que l’horloge nécessaire pour le cadencement.
  + L’autre appareil est alors l’esclave.
* La couche GATT introduit les rôles de **serveur** et de **client** (voir section GATT ci-après).
* La couche GAP introduit 4 rôles : ***broadcaster***, ***scanner*** (ou***observer***), ***peripheral***, ***central***. Le *broadcaster* s’oppose au *scanner*, le *peripheral* s’oppose au *central*. Ce qu’il est fondamental de comprendre avec le Bluetooth Low Energy, c’est qu’il y a deux façons pour que deux appareils communiquent ensemble. Soit le mode dit connecté, soit le mode *advertising*. C’est ainsi que l’on fait le lien avec les deux types d’*events* et les deux types de canaux radios introduits précédemment. En **mode *advertising***, un appareil émet des trames sur les 3 canaux d’avertising et ces trames sont accessibles à tout autre appareil à l’écoute sur ces 3 canaux. En **mode connecté**, un lien est établi entre deux appareils et eux seuls peuvent alors communiquer ensemble sur des canaux connus négociés ; le lien peut éventuellement être authentifié et encrypté pour accroître la sécurité.
  + Le rôle de *Broadcaster* est destiné à des applications qui ne font qu’émettre. Les appareils supportant ce rôle envoient des événements d’*avertising* pour diffuser des données. Ce rôle ne supporte pas le mode connecté.
  + Le rôle de *Scanner* est dédié à des applications qui ne font que recevoir. Il s’agit du rôle dual à l’*Observer*. Un appareil basé sur ce rôle reçoit des données diffusées lors d’événements d’*advertising*. Ce rôle ne supporte pas le mode connecté.
  + Le rôle *Peripheral* est destiné à des appareils qui supporte une (ou plusieurs) connexions et sont moins complexes et plus contraints qu’un appareil avec le rôle *Central*. Un appareil avec un rôle *Peripheral* a besoin d’un *Controller* esclave (par opposition à maître).
  + Le rôle Central supporte plusieurs connexions avec différents appareils Peripheral ; un tel appareil est l’initiateur des connexions et a besoin d’un *Controller* maître. Il est doté de fonctionnalités plus complexes et plus coûteuses que le *Peripheral*.

**3. Le protocole GATT**

Avec la technologie Bluetooth Low Energy, un appareil Bluetooth Smart Ready (le maître) va établir un lien avec un appareil Bluetooth Smart (l’esclave).

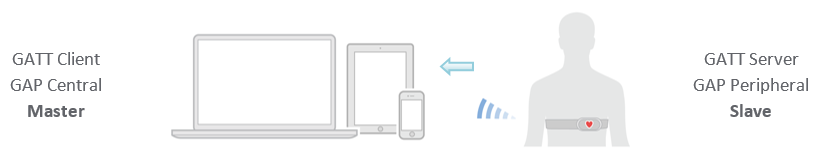
Le plus souvent, l’esclave est un serveur GATT et le maître est client GATT. Autrement dit, le serveur expose à un client des données, appelées *attributes*. Les *attributes*, tels que définis par le protocole ATT sont formatés soit en ***services***, soit en ***characteristics***.

Un *service* peut contenir une collection de *characteristics*.

Un *characteristic* contient une description, une unique valeur et des descripteurs qui qualifient la valeur.

La valeur incluse dans le *characteristic* est la valeur que le client cherche en pratique à récupérer, comme par exemple le nombre de battements cardiaques relevés par un capteur placé près du cœur.

Ainsi, si le *service* est l’agrégat de plusieurs *characteristics* faisant référence à des valeurs proches, on peut alors voir le service comme un usage de l’appareil : Blood Pressure, Heart Rate, Location ou encore Navigation, etc. De nombreux services sont standardisés

[](https://blog.groupe-sii.com/wp-content/uploads/2015/11/BLE-3-GAT.png)

*Figure 4: Communication Bluetooth Low Energy entre un client et un serveur qui héberge le service GATT*

Le fait d’utiliser des *services* standardisés permet à des appareils Bluetooth de différents fabricants de communiquer ensemble car ils ont une définition commune des valeurs qu’ils échangent sur les ondes. Toutefois, une application est libre de définir ses propres *services*.

Un client peut envoyer une requête au serveur pour aller lire ou écrire une valeur au sein d’un *characteristic*d’un *service*. On parle alors de *Write Command*, *Write Request*, *Read Request*.

Mais le serveur peut aussi prendre l’initiative d’envoyer de lui-même cette valeur. On parle alors de *Notification* ou d’*Indication*.

Pour résumer :

**Deux appareils connectés communiqués au travers d’*attributes*. Les *attributes* sont hébergés dans un serveur GATT ; un client GATT va aller les lire et/ou les écrire.**

**Les *attributes* décrivent soit des services, soit des *characteristics*. Les *services* sont des collections de valeurs contenues dans des *characteristcs*.**