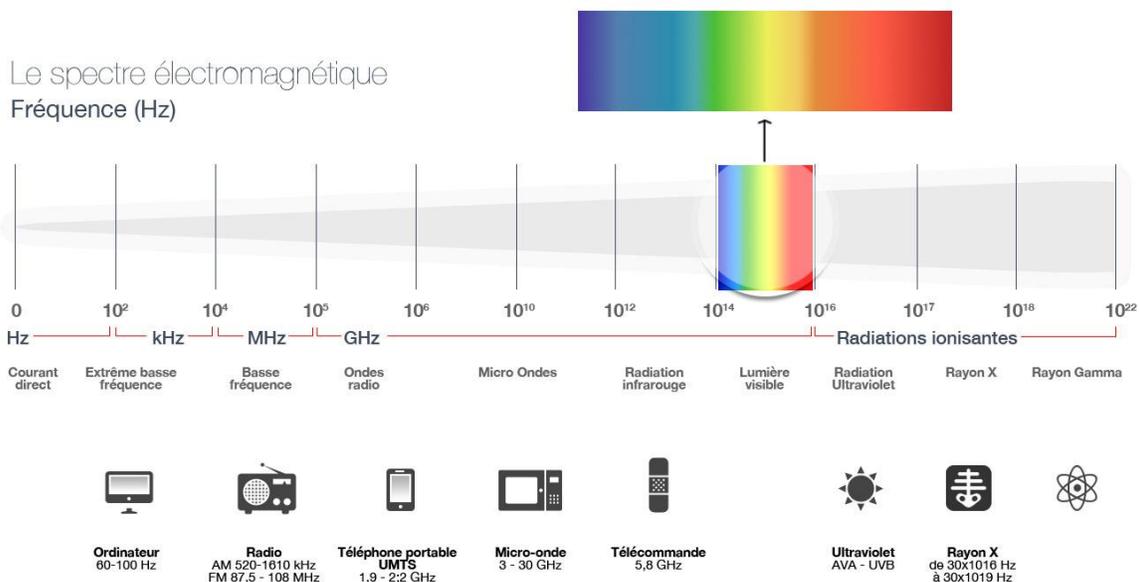


Communications sans fil

IOC - MU4IN109

Objectifs

Permettre à des équipements informatiques de communiquer sans lien physique hormis des ondes électromagnétiques.



IOC - MU4IN109

<https://electromagnetique.info/le-spectre-electromagnetique/>

Remplacement des fils par des ondes

L'objectif des liaisons sans fils est de remplacer les fils par des ondes.

- Avantages
 - coût d'installation,
 - mobilité des équipements,
 - flexibilité du réseau
 - autonomie
- Inconvénients
 - performance,
 - sécurité,
 - disponibilité (interférences)
 - autonomie

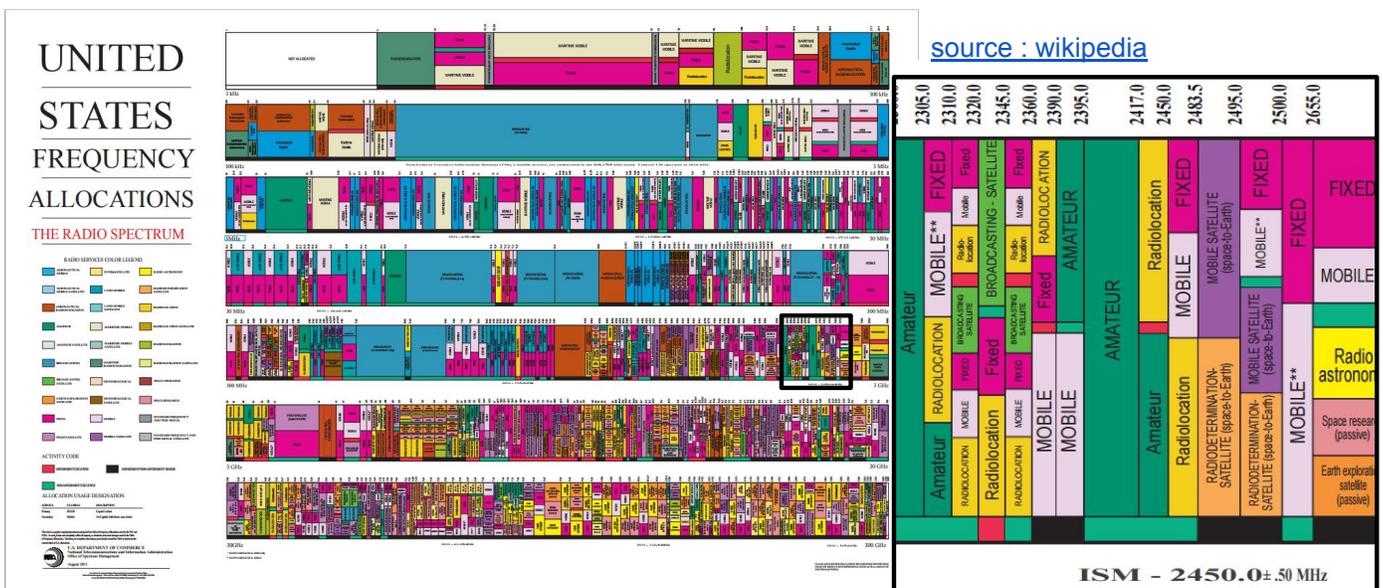
IOC - MU4IN109

3

Usage des radiofréquences

Les fréquences sont très utilisées pour la communication et sont très règlementées

→ un mauvais usage est un délit = amende + confiscation + prison !



IOC - MU4IN109

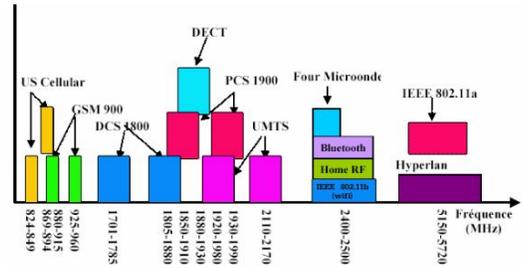
ISM : Industriels
Scientifiques et Médical

4

Bandes ISM

Fréquences **libres** pour les usages
Industriels **S**cientifiques et **M**édicaux

- pas de demandes aux autorités pour l'usage
- MAIS niveau de puissance limité
- et fréquences différentes suivant les continents (voire les pays)



Frequency range	Bandwidth	Center frequency	Availability
6.765 MHz - 6.795 MHz	30 kHz	6.780 MHz	Subject to local acceptance
13.553 MHz - 13.567 MHz	14 kHz	13.560 MHz	Worldwide
26.957 MHz - 27.283 MHz	326 kHz	27.120 MHz	Worldwide
40.660 MHz - 40.700 MHz	40 kHz	40.680 MHz	Worldwide
433.050 MHz - 434.790 MHz	1.74 MHz	433.920 MHz	Region 1 only and subject to local acceptance (within the amateur radio 70 cm band)
902.000 MHz - 928.000 MHz	26 MHz	915.000 MHz	Region 2 only (with some exceptions)
2.400 GHz - 2.500 GHz	100 MHz	2.450 GHz	Worldwide
5.725 GHz - 5.875 GHz	150 MHz	5.800 GHz	Worldwide
24.000 GHz - 24.250 GHz	250 MHz	24.125 GHz	Worldwide
61.000 GHz - 61.500 GHz	500 MHz	61.250 GHz	Subject to local acceptance
122.000 GHz - 123.000 GHz	1 GHz	122.500 GHz	Subject to local acceptance
244.000 GHz - 246.000 GHz	2 GHz	245.000 GHz	Subject to local acceptance

IOC - MU4IN109

"Radio Regulations, Edition of 2012".

5

Technologies existantes ...

- WIFI
- Bluetooth
- Lora
- Zigbee
-

Combien d'autres ?

IOC - MU4IN109

6

Technologies existantes en vrai

Ondes radiofréquences

- WiFi, WiMax
- Bluetooth classic et Low Energy
- ZigBee
- ANT+
- LoRaWan
- Sigfox
- Z-Wave
- NRF24
- Enocean
- GSM (2G), UMTS (3G), LTE (4G), 5G
- NB-IoT
- RF4CE
- Nike+

Ondes lumineuses

- irDA
- LiFi

Ondes magnétiques

- RFID
- NFC

et plein d'autres !

Critères de comparaison

Aucune technologie — existante — n'est meilleure ou pire qu'une autre, il faut définir précisément les besoins de l'application en fonction des critères distinctifs des technologies.

- Portée : distance entre les équipements
- Immunité face aux obstacles
- Bande passante requise (quantité de données)
- Energie disponible (autonomie sur batterie)
- Latence de connexion ou d'émission
- Robustesse des équipements et maintenance
- Robustesse du réseau (quelle conséquence si un nœud disparaît)
- Sécurité des transactions
- Réseaux et canaux radio utilisés
- Coût et disponibilité des équipements

Portée

[LP]WAN [Low Power] Wide Area Network (World Wide)

→ p. ex. GSM, UMTS, ...

MAN Metropolitan Area Network (50km)

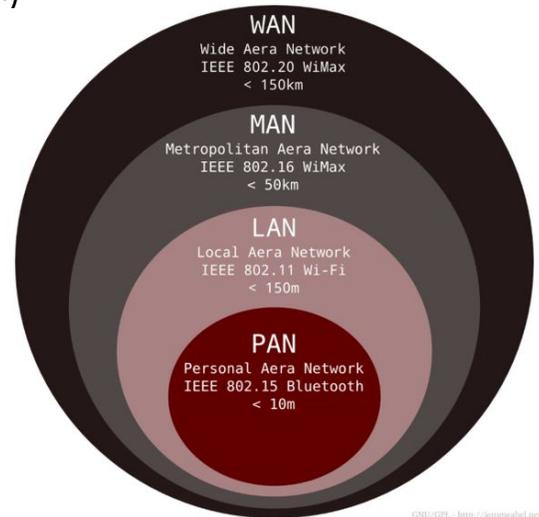
→ p. ex. LoRaWan, WiMax, ...

LAN Local Area Network (100m)

→ p. ex. WiFi, ZigBee, ...

PAN Personal Area Network (10m)

→ p. ex. Bluetooth, Nike+, ...



GNU/GPL - <http://promotehal.net>

IOC - MU4IN109

9

Comparaison

Technical capabilities	Low Power Wide Area Networks (LPWAN)							Short Range Networks				
	LoRaWAN	Neul	NWave	SigFox	Weightless -N	Weightless -P	Cellular	BLE	WiFi	Thread	ZigBee	Z-Wave
Range (km/m)	2-5 urban; 15 suburban; 45km rural	up to 10km	up to 10km	up to 10km urban; 50km rural	5km	2km	35km GSM; 200km 3G/4G	80m	50m	Mesh	100m/Mesh	30m/Mesh
Deep Indoor Performance	Yes	ISM yes, Whitespace no	Yes	Yes	Yes	Yes	No	No	No	No	-	-
Freq. Band	Varies, Sub-GHz	ISM or Whitespace	Sub-GHz	Frequency independent; 868/902MHz	Sub-GHz	Sub-GHz	900/1800/1900/2100MHz	2.4GHz	2.4GHz	2.4GHz	915MHz/2.4GHz	900MHz
ISM?	Yes	Yes, depends on base-station	Yes	Yes	Yes	Yes	Depends	Yes	Yes	Yes	Yes	Yes
Fully Bi-Directional	Yes, depends on mode	Yes	No	No	Uplink only	Yes	Yes	Yes	Yes	-	Yes	Yes
Data Rate	0.3 - 50 kbps adaptive	10 - 100kbps	100bps	10 - 1000bps	30kbps - 100kbps	up to 100kbps adaptive	35-170kbps GSM/ 3 - 10Mbps LTE	< 1mbps	600Mbps max	-	250kbps	10 - 100kbps
Power Profile	Low	Low	Low	Low	Low	Low	Medium	High	High	Low	Low	Low
Authentication	Yes	-	Yes	Yes	Yes	Yes	High security, back by major telecoms	Trusted devices problematic	Yes	Yes	Yes	Yes
E2E Encryption	Yes	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Over the Air Software Upgrades	Yes	-	No	No	No	Yes	Yes	Yes	Yes	-	Yes	Yes
Supports sensors moving between hubs	Yes	-	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes, mesh-based	Yes, mesh-based
Location Aware	Yes	-	No	No	No	-	Yes	No	Yes	-	-	-
Operational Model	Public or private	-	Public or private	Public	Public or private	Public or private	Public or private	Public or private	Public or private	Private/WiFi backbone	Public or private	Public or private
Standard	LoRaWAN	Weightless	Weightless	No	Weightless	Weightless	GSM, LTE etc	Bluetooth 4.0	IEEE802.11	Thread, based on 6LoWPAN IEEE802.15.4	ZigBee	Z-Wave
Scalability	Yes	Yes	Yes	Yes	Limited	Yes	Yes	Yes	Yes	Yes	Yes	Limited

IOC - MU4IN109

<https://www.objetconnecte.com/comment-choisir-meilleur-reseau-pour-son-projet-0312/>

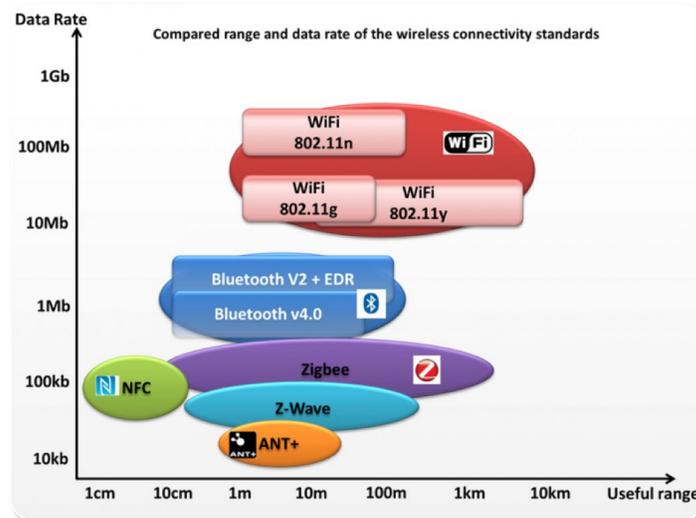
10

Comparaison

Les caractéristiques ne sont pas indépendantes.

- la portée est fonction de la consommation énergétique
- à puissance d'émission égale la portée est inversement proportionnelle à la fréquence, et la bande passante dépend de la fréquence.

Sur ce schéma ne figure pas l'empreinte énergétique.

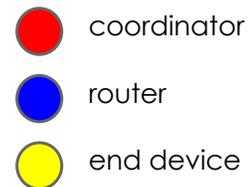


IOC - MU4IN109

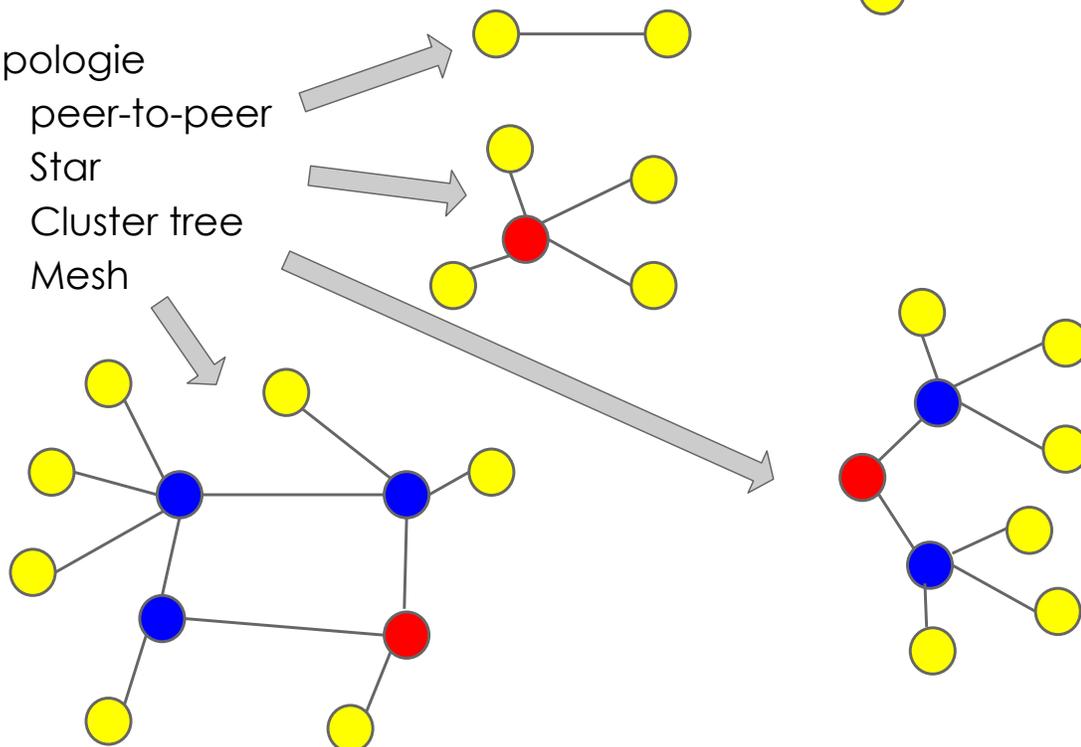
<http://www.wifilabs.com/2016/06/21/quelle-technologie-radio-pour-les-objets-connectes-deuxieme-partie/>

11

Types de réseau



- Topologie
 - peer-to-peer
 - Star
 - Cluster tree
 - Mesh



IOC - MU4IN109

12

Solutions grand public

ZigBee → réseau d'objets

- protocole simple (code 4ko à 32ko)
- 64 kilo noeuds : star, cluster, mesh
- autonomie plusieurs années
- 250kb/s
- portée 100m

Bluetooth → remplacement de câbles

- protocole complexe (code 250ko)
- 7 noeuds actifs : star, scatternet
- autonomie en semaines
- 0.7 à 2 Mb/s
- portée 10m

Wi-Fi → internet

- protocole très complexe (1Mo)
- 32 noeuds : star + roaming
- autonomie en heures
- 11Mb/s à x100Mb/s
- portée 100m

Bluetooth Low Energy → IOT

- protocole complexe (code 250ko)
- autonomie en années
- P-à-P ou Star nb de noeuds illimité
- découverte de services
- 300 kb/s
- portée 10m à 100m

LoRaWAN → IOT grandes distances

- 5 à 15km
- autonomie en années
- Star nombre de noeuds illimité
- 50 kb/s

Besoins pour un système domotique

Communications locales Raspberry Pi ↔ capteur/actionneur

- Faible coût
 - Quelques euros par noeud
- Basse consommation
 - Fonctionnement sur batterie
- Faible latence - Faible débit moyen
 - (re)connexion rapide d'un noeud
 - Le débit n'est pas une contrainte (si pas de vidéo)
- Simplicité de programmation
 - Le capteur ou l'actionneur doit être facilement programmable si possible à chaud sans reconfiguration
- Mise en réseau (si le lieu d'implantation est vaste)
 - L'accès au capteur à l'actionneur peut passer par des routeurs

ZigBee

IOC - MU4IN109

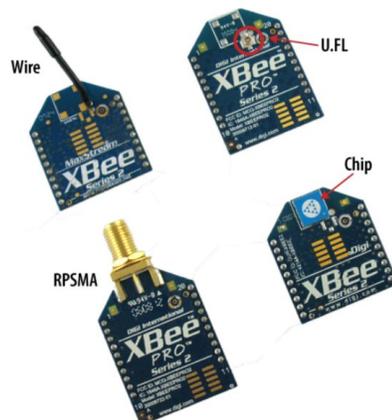
Présentation ZigBee - XBee



- Protocole de communication à très basse consommation pour les réseaux de dimension personnelle en radio fréquence
- (Wireless Personal Area Networks : WPAN).
- Norme [IEEE 802.15.4](#) (2003) : 868MHz - 915MHz - 2.4GHz
- ZigBee Alliance (2004)

XBee™

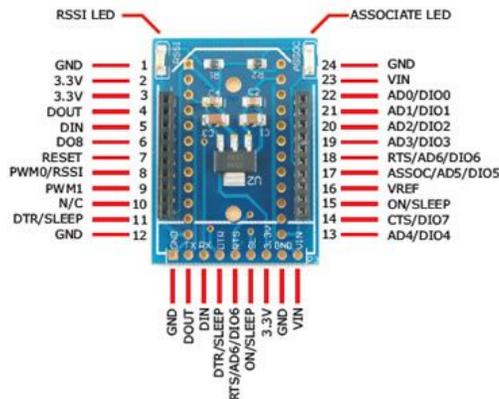
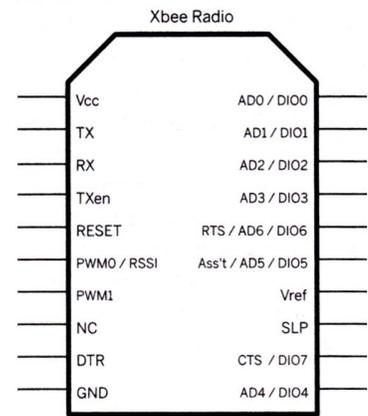
- Modules fabriqués par l'entreprise [Digi International](#)
- 2 types de modèles
 - 1 / 2 mW **XBee** (20\$)
 - 100 / 500 mW **XBee-PRO** (30\$)



IOC - MU4IN109

Module XBee

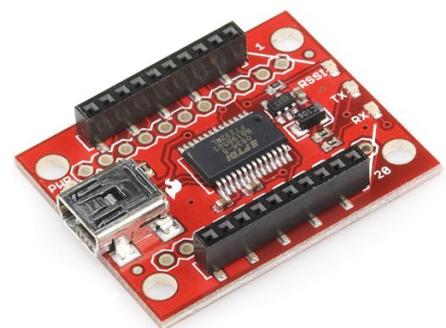
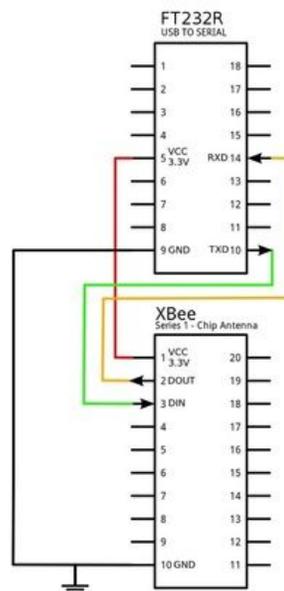
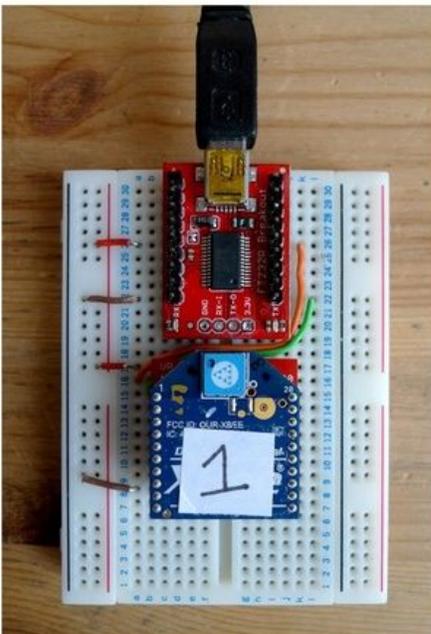
- Protocole rs232 de **9600** à 115200 bauds
- 4 fils (minimum)
 - TX (data out)
 - RX (data in)
 - Vcc (2.8 à 3.6V)
 - GND
- 7 GPIO Num-Ana
- 2 PWM



Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

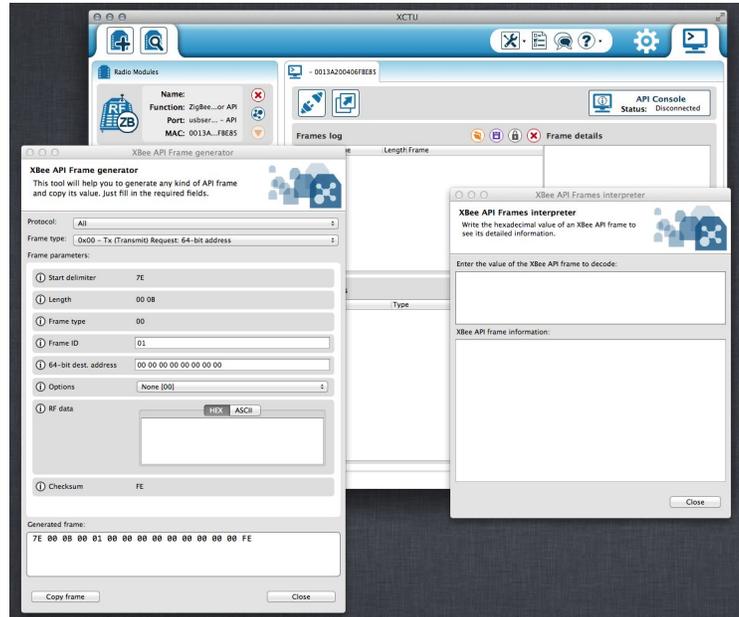
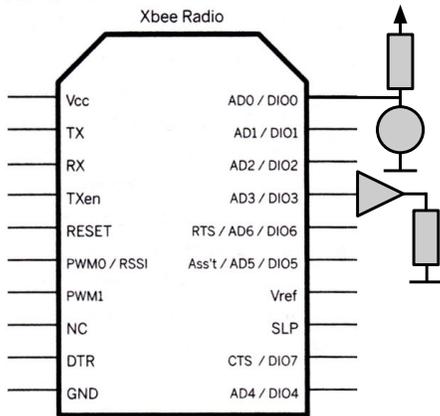
Configuration par USB

Comme pour l'Arduino, la "programmation" se fait depuis un ordinateur hôte via un câble USB utilisée par le protocole RS232 (série). On peut ainsi configurer l'usage de ses GPIOs et PWM.



XCTU : Configuration des nœuds

- Le logiciel XCTU de Digi International (windows / mac) permet de le commander et de **changer le firmware** (impossible à chaud) : Coordinateur, router, end-device et programmation des ports
- Il permet la programmation des end-devices pour demander la lecture périodique de capteurs et la commandes de leds



IOC - MU4IN109

19

Commandes AT

Les **Commandes AT**, constituent un langage de commandes développé à l'origine pour le modem [Hayes Smartmodem 300](https://fr.wikipedia.org/wiki/Commandes_Hayes) (https://fr.wikipedia.org/wiki/Commandes_Hayes). Ce jeu de commandes s'est ensuite retrouvé dans tous les [modems](#) produits et dans beaucoup d'équipement réseau.

Pour les modems, les commandes permettent :

- de composer un numéro de téléphone
- de commander le raccordement du modem à la ligne (décrocher le téléphone)
- de connaître l'état de la ligne : tonalité d'invitation à transmettre, ligne occupée...
- de spécifier le type de transmission et le protocole de liaison à utiliser
- etc.

Pour les modules Xbee, les commandes permettent :

- de configurer les modules (usage de leur GPIO, numéro d'identification)
- d'envoyer des trames (c'est ce que fera l'éventuel MCU connecté)
- de lire l'état interne du module ou des GPIO
- etc...

IOC - MU4IN109

20

Commandes AT ZigBee

- Etablir une communication rs232 9600bauds avec le module
 - Taper +++ → attendre OK // mode commande
 - ATMY1234 → attendre OK // programmation de l'adresse
 - ATMY → 1234 // vérification de la prog.
- [Commandes AT des modules XBEE](#)

Commande AT	Nom	Description	Exemple
ATCN	Command end	Fin du mode commande	ATCN
ATND	Node Discover	Découvrir les XBee présents sur le réseau	ATND
ATID	Pan ID	Définir le numéro d'identification du réseau	ATID 35
		Lire le numéro d'identification du réseau	ATID
ATNI	Node Identifier	Définir le nom de nœud	ATNI E35_Ordi
		Lire le nom de nœud	ATNI
ATWR	Write	Sauvegarder la configuration en mémoire non-volatile. Permet de conserver la configuration même si le XBee n'est plus alimenté.	ATWR

IOC - MU4IN109

21

Frame maker de Digi International

Pour envoyer un paquet depuis un MCU, il faut produire un paquet avec l'identifiant de l'émetteur, l'adresse du destinataires, la longueur, le message, un CRC, etc.

L'outil **Frame maker** permet d'aider à les construire sans erreur.

Notez que l'overhead est de près de 20 octets, c'est beaucoup...

FieldName	FieldValue	Data Type	Description
Delimiter	7E	Byte	Start Delimiter
Length	0014	Word	Number of bytes between length and checksum fields.
API	17	Byte	Remote AT Command
FrameID	01	Byte	Identifies the UART data frame for the host to match with a subsequent response. If zero, no response is requested.
64DestAddr	0013A200606F8D6F	EU164	Destination 64-bit (MAC/EUI64) address. The following addresses are also supported: 0x0000000000000000 - Reserved for the coordinator. 0x000000000000FFFF - Broadcast address
16DestAddr	FFFF	NWK16	Destination 16-bit network address, if known. Use 0xFFFF if the address is unknown, or if sending a broadcast. Other reserved addresses: 0xFFFFC - broadcast to all routers; 0xFFFFD - broadcast to all non-sleepy devices; 0xFFFFF - broadcast to all devices including sleepy ED.
CmdOptions	02	Byte	0x02 - Apply changes on remote device. NOTE: If this bit is not set, an AC (or WR+FR) command must be sent before changes will take effect. All other bits must be set to zero.
AT Cmd	D3	ATCmd	Command name of two ASCII characters.
AT CmdData	5	Variable	If present, set the register to this value. If absent, get the value of the register. String values should be terminated with a zero byte.
Checksum	bc	Byte	0xFF minus 8-bit sum of bytes between the length and checksum fields.
Packet	7E 00 10 17 01 00 7D 33 A2 00 60 6F 8D 6F FF FE 02 44 33 05 BC		message utile 2 octets → 21 octets

IOC - MU4IN109

22

Bluetooth Low Energy

IOC - MU4IN109

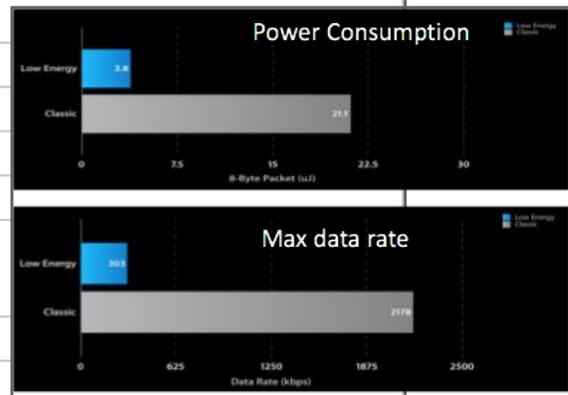
Bluetooth Low Energy

- Le BLE est une norme différente du Bluetooth classique. Les usages sont très différents, mais les clients BT sont souvent compatibles avec les deux normes.
- Objectif du BLE : fonctionner très longtemps sur une pile bouton plusieurs mois ou plusieurs années
 - Courant de pic < 15 mA (800 mA en WiFi)
 - Courant moyen < 5 μ A
 - paquets courts (~20B) => moins de mémoire
- BLE a été conçu pour être simple, bas coût et robuste.
- BLE fonctionne sur la bande ISM de 2,4 GHz sur 40 bandes espacés chacune de 2MHz
- Utilise FHSS (Frequency-Hopping Spread Spectrum) (Hedy Lamarre)
- Vitesse de transmission réelle : ~10kB/s
- Portée 30m à 50m théorique mais 3 à 5m en pratique
- Temps de transfert min d'un paquet 3 ms contre 100ms pour le BT



Bluetooth versus Bluetooth Low Energy

Technical specification	Classic Bluetooth technology	Bluetooth low energy technology
Radio frequency	2.4GHz	2.4GHz
Distance/Range	~10-100 meters	~10-100 meters
Symbol rate	1-3Mbps	1Mbps
Application throughput	0.7 – 2.1Mbps	305kbps
Nodes/Active slaves	7	Unlimited
Security	56 to 128 bit	128-bit AES
Robustness	FHSS	FHSS
Latency (from not connected state to send data)	100+ ms	<6ms
Government regulation	Worldwide	Worldwide
Certification body	Bluetooth SIG	Bluetooth SIG
Voice capable	Yes	No
Network topology	Point-to-point, scatternet	Point-to-point, star
Power consumption	1 (reference value)	0.01 to 0.5 (use case dependent)
Service discover	Yes	Yes
Profile concept	Yes	Yes



IOC - MU4IN109

<https://www.bluetooth.com/bluetooth-technology/radio-versions>

25

Rôles : Clients / Serveurs

Le rôle d'un objet définit comment il communique, il en existe 4.

- **Broadcaster**

Si c'est un serveur et transmet ces données périodiquement à tous ceux qui sont dans l'entourage. L'objet n'accepte pas les connexions entrantes.

- **Observer (scanner)**

S'il se contente d'observer les données envoyées par un broadcaster et de les utiliser. C'est un client passif car il ne peut pas se connecter aux serveurs.

- **Peripheral (slave)**

(S'il s'agit d'un esclave qui accepte les connexions d'un maître. C'est lui qui envoie périodiquement des paquets chercher un maître.)

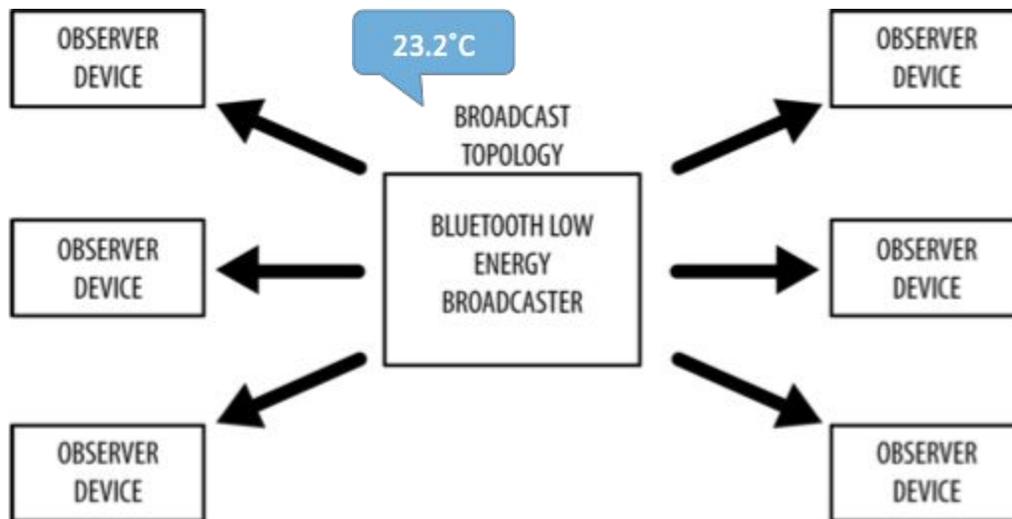
- **Central (master)**

S'il agit comme un client maître qui cherche régulièrement des esclaves dans son entourage. Quand il en trouve, il s'y connecte et commence à échanger des données.

IOC - MU4IN109

26

Topologie du réseau : Broadcast

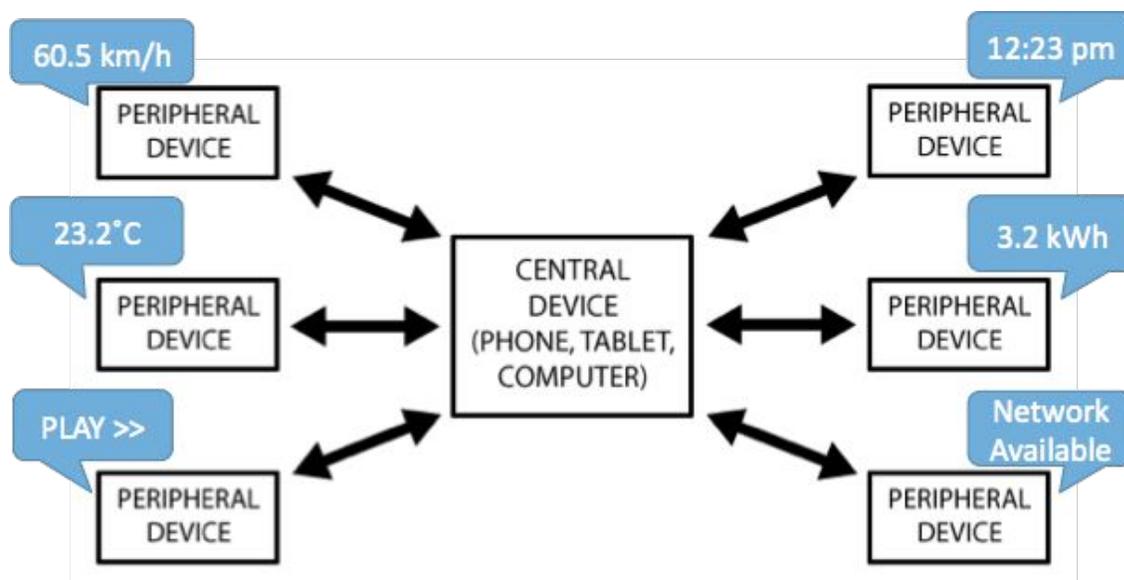


- Les broadcasters envoient périodiquement à tous les observers des Advertising Packets pour diffuser des informations.
- Les observers scannent régulièrement les Advertising Packets pour lire les informations.

IOC - MU4IN109

27

Topologie du réseau : Connected



- Un broadcaster peut changer de rôle et devenir un peripheral pour communiquer avec un central de manière chiffrée et permanente.
- Un central device peut communiquer avec plusieurs peripheral devices et avec BLE 4.1, un peripheral device peut communiquer avec plusieurs central devices

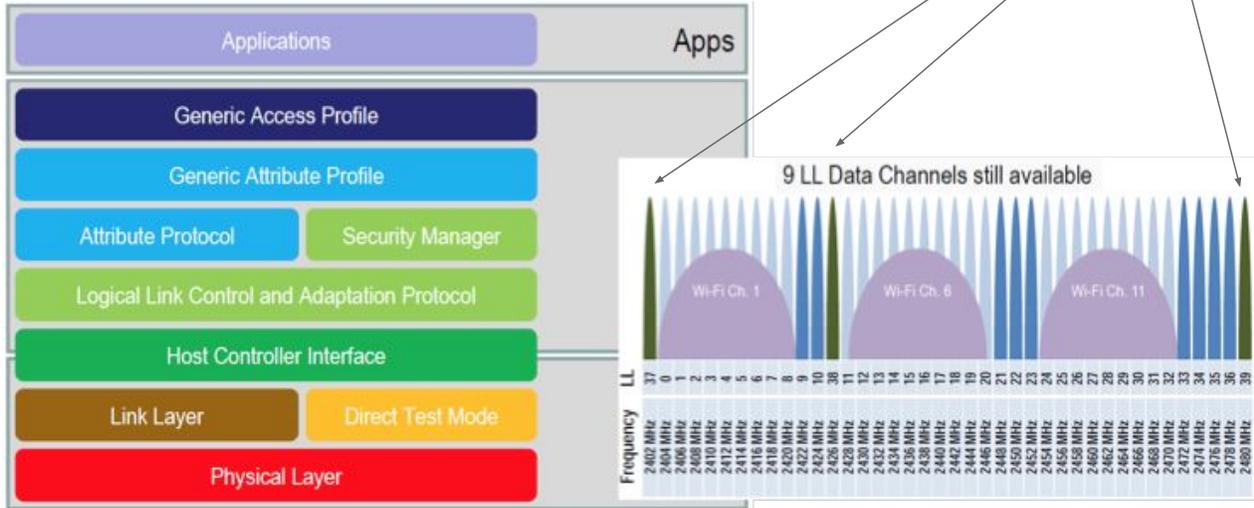
IOC - MU4IN109

28

Bluetooth Low Energy : Canaux

Physical Channels

- Comme le BT partage la bande ISM avec le WIFI les canaux advertising ont été choisis pour réduire les collisions
- Il reste 9 canaux disponibles : les canaux d'advertising sont les 37, 38 et 39.



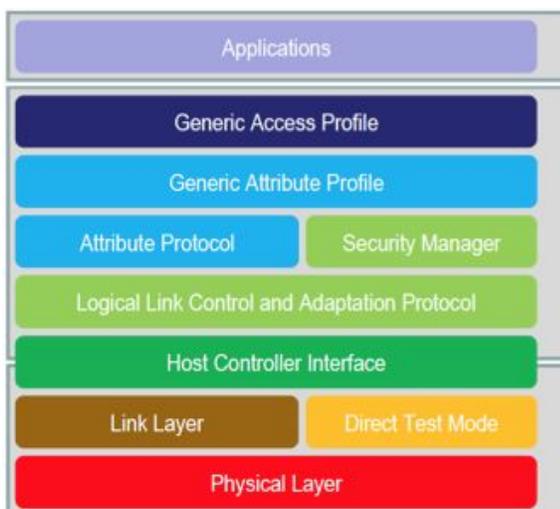
IOC - MU4IN109

29

Bluetooth Low Energy : Protocol

Couches protocolaires

- Elles gèrent de la connexion à l'envoi des paquets sur réseaux
- mots clés : faible consommation, standardisation, et sécurité



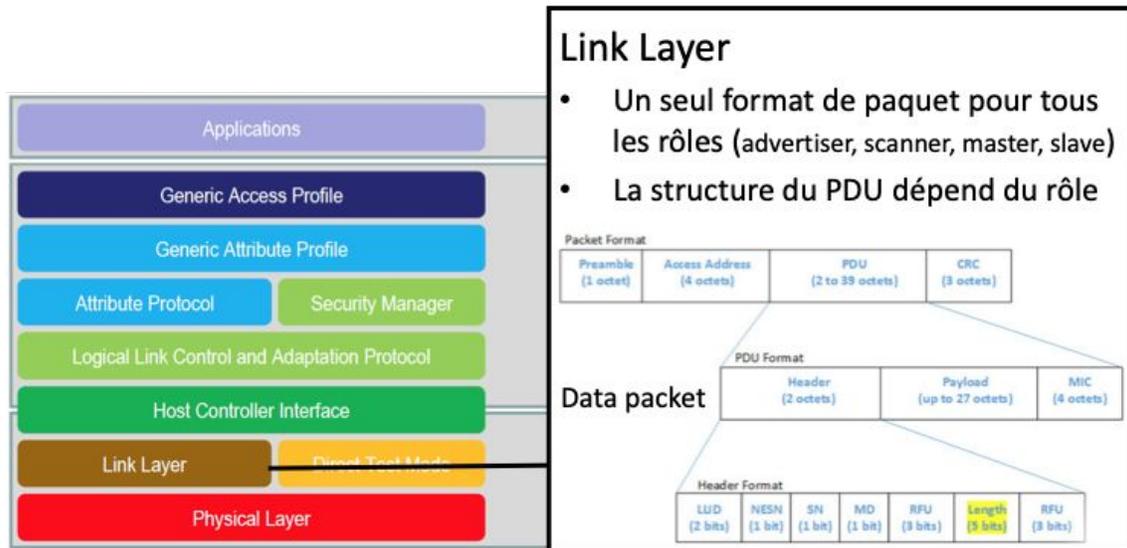
GAP	Protocole de connexion
GATT	Gestion standardisé des services et des caractéristiques
ATT	Protocole de transport des données
SM	Chiffrement des données
L2CAP	segmentation et réassemblage des paquets, gestion qualité de service
LL	Encapsulation en paquets
PHY	Gestion de la couche physique

IOC - MU4IN109

30

Couche Liaison (LL)

- On remarque que pour le BLE, les paquets sont identifiés par leur numéro de transaction (Access Address) et pas par le couple d'identifiants [émetteur-destinataire] (comme c'est le cas p.ex. pour ZigBee et IP) C'est une solution qui permet d'économiser des octets.
- Ce numéro est créé aléatoirement au moment de la connexion entre un master et un slave.

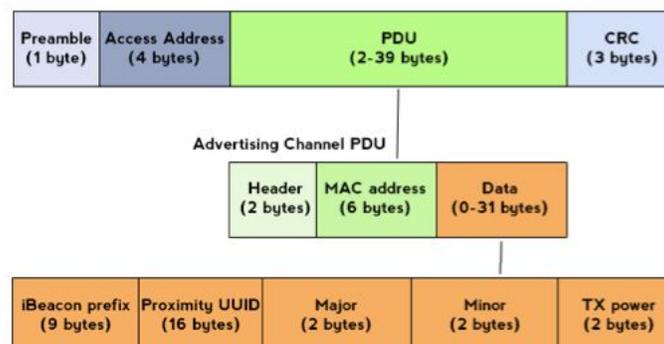


IOC - MU4IN109

31

Advertising Packet

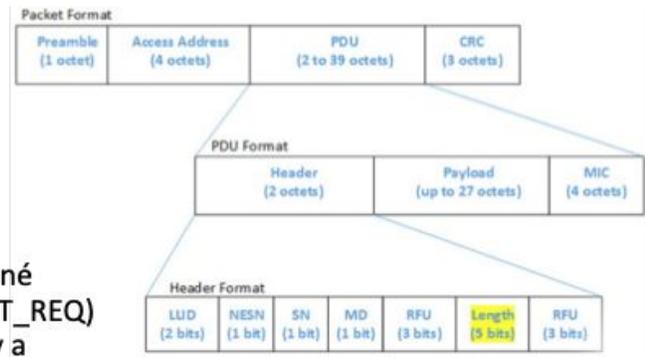
- Preamble 0xAA (0x55?)
- Access Address : **0x8E89BED6** pour tous les advertising packets.
- PDU: Advertising packets (canaux 37,38,39)
- CRC: 3 octets calculé sur le PDU
- Le payload (PDU) contient ce qui identifie le composant
 - Header : ... length...
 - MAC (BDA): identifiant 48b unique
 - Des éventuelles données (nécessairement publiques) ici c'est un [iBeacon](#) (système de positionnement en intérieur)



IOC - MU4IN109

32

Data Packet en mode connecté



- Preamble : 0xAA
- Access Address : nombre aléatoire donné dans le paquet de connexion (CONNECT_REQ) Ce n'est pas n'importe quel nombre il y a des contraintes sur les séquences de bits.
- PDU : Data
- CRC: 3 octets calculé sur le PDU
- Header →
- MIC : Message Integrity Check
- Payload : dépend du type de contenu...
- LLID : type de paquet : vide, data, ctrl
- NESN, SN : numéro de séquence
- MD : (More Data) pour dire qu'on va chainer les paquets
- RFU : Futur Use
- Length : de Payload + MIC

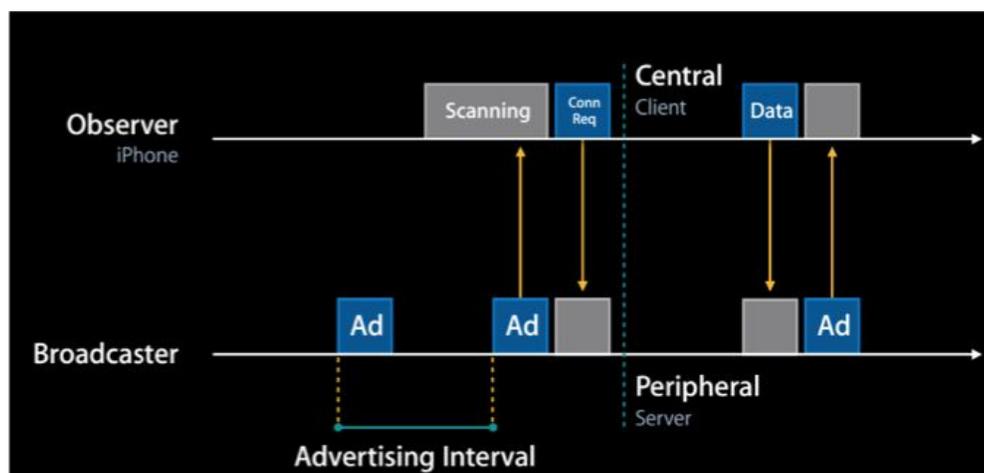
LLData									
AA (4 octets)	CRCInit (3 octets)	WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	ChM (5 octets)	Hop (5 bits)	SCA (3 bits)

IOC - MI

33

Connexion

- Quand une connexion est établie entre un peripheral device et un central device, c'est le central qui gère et qui démarre les échanges.
- Le central (client) va choisir une adresse d'accès qui sera utilisée pour identifier les transactions avec son périphérique (server).
- Pour réduire la consommation énergétique, les deux devices s'entendent sur des échanges périodiques (connection event) entre lesquels ils sont en veille.

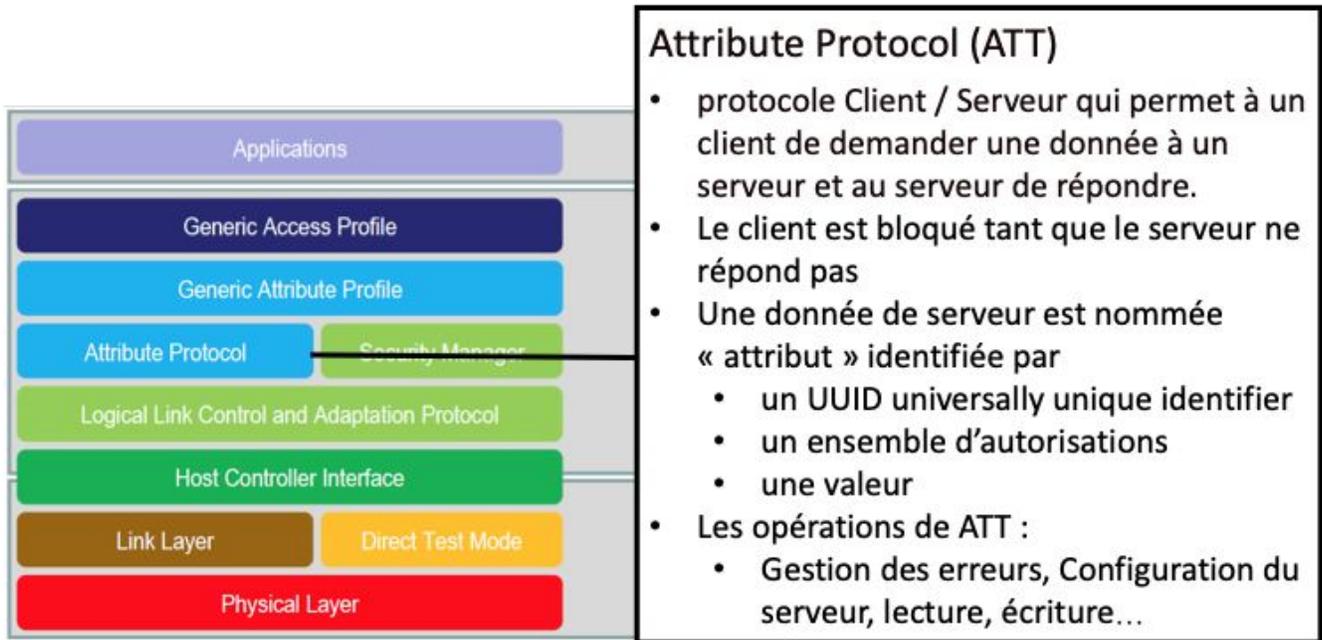


IOC - MU4IN109

34

Couche ATT

GATT signifie "attributs génériques", il définit une structure de données hiérarchique exposée par les périphériques BLE. GATT définit la manière dont deux appareils BLE envoient et reçoivent des messages standard.

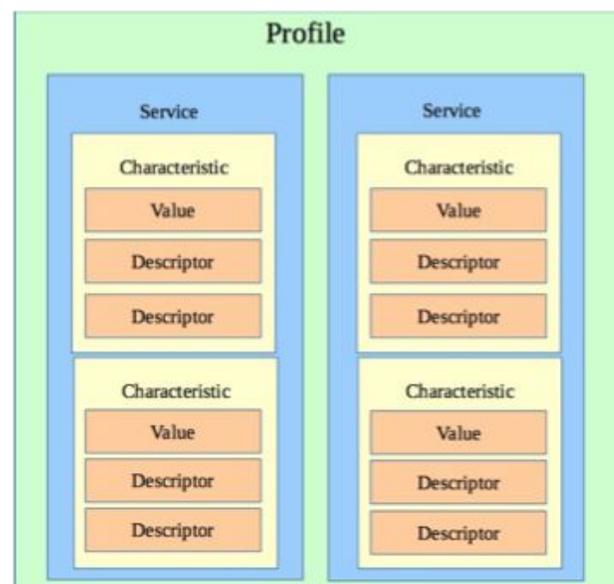


IOC - MU4IN109

35

GATT Generic Attribute Profile

- Les données sont organisées de manière standard
- Un serveur peut proposer un ensemble de profiles comprenant plusieurs services définis par des characteristics répondant à des propriétés qui sont des types d'accès à des descriptors et des values.
- Les values peuvent être des entiers, des tableaux, des strings, etc.
- Les descriptors permettent de configurer des fonctionnalités (notifications, etc.)
- Les types de profiles sont standardisés
 - Blood Pressure (capteur santé)
 - fitness
 - alert notification
 - etc.



IOC - MU4IN109

36

GATT Generic Attribute — Définitions

Profile

Structure dans laquelle les données de profil sont échangées.
Elle définit les éléments de base, tels que les services et les caractéristiques.

Service

Ensemble de données et de comportements associés permettant d'accomplir une fonction ou une caractéristique particulière.
Elle peut contenir des caractéristiques obligatoires ou facultatives.

Characteristic

Valeur utilisée dans un service avec des propriétés et des informations de configuration sur le mode d'accès, d'affichage ou de représentation de la valeur.

Par exemple, le *profile* d'un capteur cardiaque qui aurait 2 services.

1. Service "fréquence cardiaque" avec 3 caractéristiques:
 - a. *characteristic* obligatoire : mesure de la fréquence cardiaque avec la valeur.
 - b. *characteristic* facultative : emplacement du capteur corporel .
 - c. *characteristic* conditionnelle : point de contrôle de la fréquence cardiaque .
2. Service de batterie :
 - a. *Characteristic* obligatoire : niveau de batterie .

Universally Unique ID : UUID

Numéro unique identifiant les *services*, *characteristics* et *descriptors*.
Les UUID sont inclus dans les messages envoyés par un dispositif BLE qui offre des services pour permettre à un autre dispositif de reconnaître et d'accéder à ces services spécifiques.

Il existe deux types d'UUID:

- short UUID 16 bits, par ex. le service de fréquence cardiaque est (0x180D et sa *characteristics* mesure de la fréquence est 0x2A37).
Ces UUID sont standardisés par Bluetooth SIG.
- base UUID 128 bits (libres) permet de créer des services et caractéristiques personnalisés, on peut se créer un UUID de base par ex. 4A98xxxx-1CC4-E7C1-C757-F1267DD021E8 et utiliser les 16 bits xxxx pour identifier ses propres services et *characteristics*.

UUID standard

<https://www.bluetooth.com/specifications/gatt/generic-attributes-overview>

- <https://www.bluetooth.com/specifications/gatt/services>
- <https://www.bluetooth.com/specifications/gatt/characteristics>
- <https://www.bluetooth.com/specifications/gatt/descriptors>

Services

Name	Uniform Type Identifier	Assigned Number
Generic Access	org.bluetooth.service.generic_access	0x1800
Alert Notification Service	org.bluetooth.service.alert_notification	0x1811
Automation IO	org.bluetooth.service.automation_io	0x1815
Battery Service	org.bluetooth.service.battery_service	0x180F
Blood Pressure	org.bluetooth.service.blood_pressure	0x1810
Body Composition	org.bluetooth.service.body_composition	0x181B
Bond Management Service	org.bluetooth.service.bond_management	0x181E
Continuous Glucose Monitoring	org.bluetooth.service.continuous_glucose_monitoring	0x181F
Current Time Service	org.bluetooth.service.current_time	0x1805
Cycling Power	org.bluetooth.service.cycling_power	0x1818
Cycling Speed and Cadence	org.bluetooth.service.cycling_speed_and_cadence	0x1816
Device Information	org.bluetooth.service.device_information	0x180A
Environmental Sensing	org.bluetooth.service.environmental_sensing	0x181A
Fitness Machine	org.bluetooth.service.fitness_machine	0x1826
Generic Attribute	org.bluetooth.service.generic_attribute	0x1801
Glucose	org.bluetooth.service.glucose	0x1808
Health Thermometer	org.bluetooth.service.health_thermometer	0x1809
Heart Rate	org.bluetooth.service.heart_rate	0x180D

Characteristics

Name	Uniform Type Identifier	Assigned Number
Characteristic Aggregate Format	org.bluetooth.descriptor.gatt.characteristic_aggregate_format	0x2905
Characteristic Extended Properties	org.bluetooth.descriptor.gatt.characteristic_extended_properties	0x2900
Characteristic Presentation Format	org.bluetooth.descriptor.gatt.characteristic_presentation_format	0x2904
Characteristic User Description	org.bluetooth.descriptor.gatt.characteristic_user_description	0x2901
Client Characteristic Configuration	org.bluetooth.descriptor.gatt.client_characteristic_configuration	0x2902
Environmental Sensing Configuration	org.bluetooth.descriptor.es_configuration	0x2908
Environmental Sensing Measurement	org.bluetooth.descriptor.es_measurement	0x290C
Environmental Sensing Trigger Setting	org.bluetooth.descriptor.es_trigger_setting	0x290D
External Report Reference	org.bluetooth.descriptor.external_report_reference	0x2907
Number of Digitals	org.bluetooth.descriptor.number_of_digitals	0x2909
Report Reference	org.bluetooth.descriptor.report_reference	0x2908
Server Characteristic Configuration	org.bluetooth.descriptor.gatt.server_characteristic_configuration	0x2903
Time Trigger Setting	org.bluetooth.descriptor.time_trigger_setting	0x290E
Valid Range	org.bluetooth.descriptor.valid_range	0x2906
Value Trigger Setting	org.bluetooth.descriptor.value_trigger_setting	0x290A

IOC - MU4IN109

39

Outils pour accéder au composant BLE

Sur Linux (sur un ordinateur disposant d'un périphérique BLE)

- [hciconfig](#) : pour configurer le périphérique BLE.
- [hcidtool](#) : pour configurer une connexion BLE
- [gatttool](#) : pour accéder aux attributs
- [nRF-Connect](#) : outil graphique pour manipuler les composants BLE
- [pyBluez](#) : bibliothèque Python pour accéder aux composants BLE

Sur un smartphone

- [nRF-Connect](#) : app (iOS, Android) permettant d'accéder au comp. BLE
- [LightBlue](#) : une autre app (iOS, Android) pour l'accès au BLE

Sur l'ESP32

- Les bibliothèques téléchargées pour programmer l'ESP32 sur l'IDE Arduino supportent le BLE, il y a plusieurs exemples:

IOC - MU4IN109

40

BLE Scan

```
/* Based on Neil Kolban example for IDF:
https://github.com/nkolban/esp32-snippets/blob/master/cpp_utils/tests/BLE%20Tests/SampleScan.cpp
Ported to Arduino ESP32 by Evandro Copercini */

#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEScan.h>
#include <BLEAdvertisedDevice.h>

int scanTime = 5; //In seconds
BLEScan* pBLEScan;

class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {
    void onResult(BLEAdvertisedDevice advertisedDevice) {
        Serial.printf("Advertised Device: %s \n", advertisedDevice.toString().c_str());
    }
};

void setup() {
    Serial.begin(115200);
    Serial.println("Scanning...");
    BLEDevice::init("");
    pBLEScan = BLEDevice::getScan(); //create new scan
    pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());
    pBLEScan->setActiveScan(true); //active scan uses more power, but get results faster
    pBLEScan->setInterval(100);
    pBLEScan->setWindow(99); // less or equal setInterval value
}

void loop() {
    // put your main code here, to run repeatedly:
    BLEScanResults foundDevices = pBLEScan->start(scanTime, false);
    Serial.print("Devices found: ");
    Serial.println(foundDevices.getCount());
    Serial.println("Scan done!");
    pBLEScan->clearResults(); // delete results fromBLEScan buffer to release memory
    delay(2000);
}
```

IOC - MU4IN109

41

TME

Utilisation des capacités radio de l'ESP32 :

- MQTT / WiFi pour communiquer avec les RPi3
- Installation d'une application smartphone pour lire les composants BLE environnants
- Programmation de l'ESP32 (mode server) pour exposer la mesure la valeur de la photorésistance
- Programmation de l'ESP32 (mode client) pour lire les valeurs des ESP32 en mode server.

IOC - MU4IN109

42