

# R3.IOM.16 : introduction à l'internet des objets (IdO – IoT)

Yassine HADDAB

*Professeur à l'Université de Montpellier*

*[Yassine.haddab@umontpellier.fr](mailto:Yassine.haddab@umontpellier.fr)*



# Prérequis

Notions élémentaires sur les réseaux  
notions élémentaires sur les circuits électriques

# But

Sensibilisation à l'importance de l'IoT.

Présentation des concepts fondamentaux de l'Internet des Objets.

Compréhension de la chaîne de conception des objets connectés.

# Sommaire

- 1- Introduction
- 2- Le marché de l'IoT
- 3- Concepts fondamentaux
- 4- Interactions entre le « monde numérique » et le « monde physique »
- 5- Infrastructures pour l'IoT
- 6- Solutions technologiques pour les plateformes embarquées
- 7- Exemples d'application
- 8- Bibliographie

# 1- Introduction

# Telegarden : juin 1995 (Univ. Of California)

---

Le concept d' « objet connecté » n'est pas nouveau. Exemple : Telegarden.



Agriculture – art - internet

# Nabaztag, lancé par la société Violet en 2005

---

Ce lapin connecté en Wi-Fi peut déjà lire des mails à haute voix, émettre des signaux visuels et diffuser de la musique. L'objet est toujours commercialisé aujourd'hui, sous le nom de **Karotz**.



# Pourquoi l'IoT ?

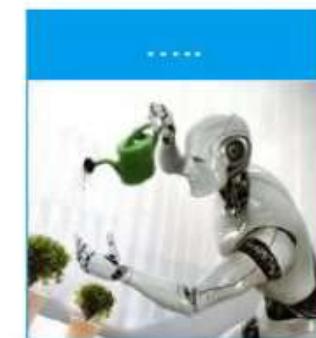
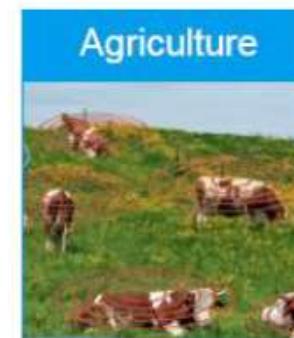
---

- Evolution naturelle des technologies : lien inévitable entre le monde numérique et le monde physique.
- Assistance à nos activités professionnelles et personnelles.
- Permet une réduction considérable des dépenses dans l'économie d'aujourd'hui (industrie, santé, sécurité, etc.).
- L'IoT est ici et il évolue rapidement ! Il n'y a pas de temps à perdre.
- 50 milliards d'objets en 2020 (estimation) !

# Domaines applicatifs de l'IoT ?

---

## Les objets connectés - IoT



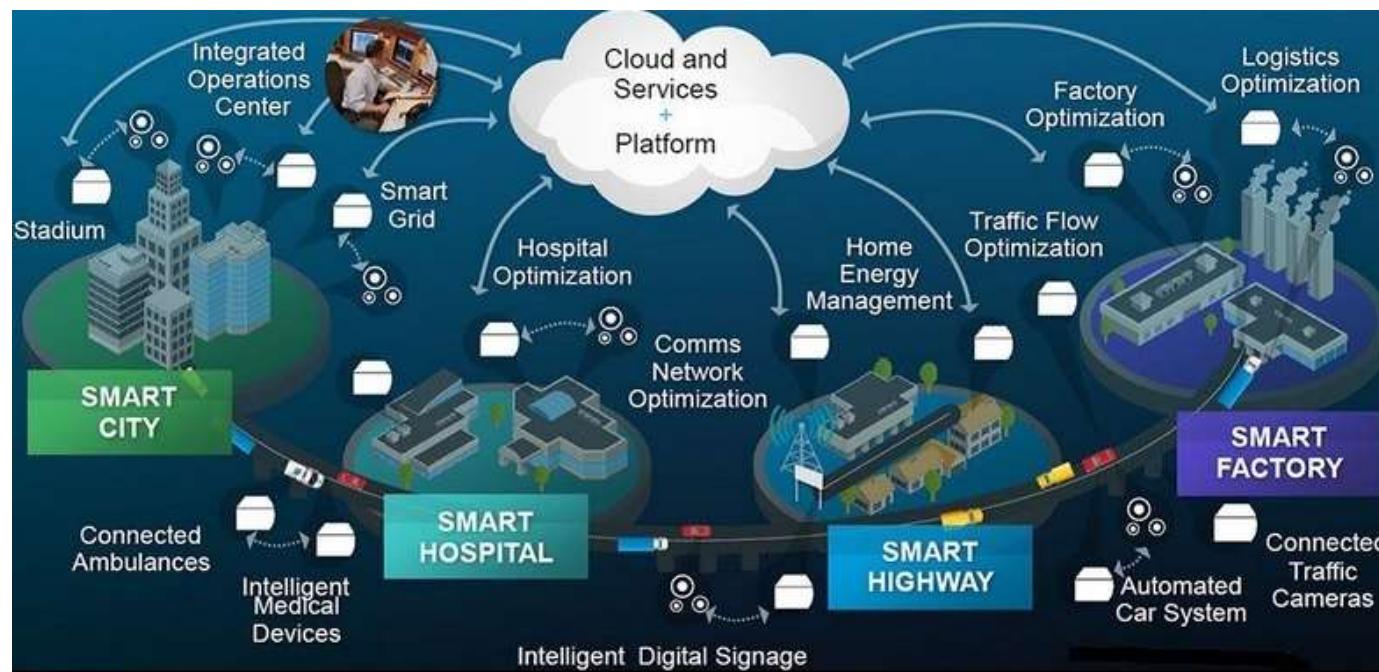
# Domaines applicatifs de l'IoT ?

---

- **Ville intelligente** : circulation routière intelligente, transports intelligents, collecte des déchets, cartographies diverses (bruit, énergie, etc.).
- **Environnements intelligents** : prédition des séismes, détection d'incendies, qualité de l'air, etc.
- **Sécurité et gestion des urgences** : radiations, attentats, explosions.
- **Logistique** : aller plus loin que les approches actuelles.
- **Contrôle industriel** : mesure, pronostic et prédition des pannes, dépannage à distance.
- **Santé** : suivi des paramètres biologiques à distance.
- **Agriculture intelligente, domotique, applications ludiques** etc.

# Domaines applicatifs de l'IoT ?

---



# Quelques définitions

---

- **Objet connecté** : objet possédant la capacité d'échanger des données avec d'autres entités physiques ou numériques.
- **Internet des objets (IdO)** : expansion du réseau internet à des objets et/ou des lieux du monde physique. En anglais, on parle d'**IoT** : **Internet of Things**.

« L'Internet des Objets est un réseau de réseaux qui permet, via des systèmes d'identification électronique normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant. »

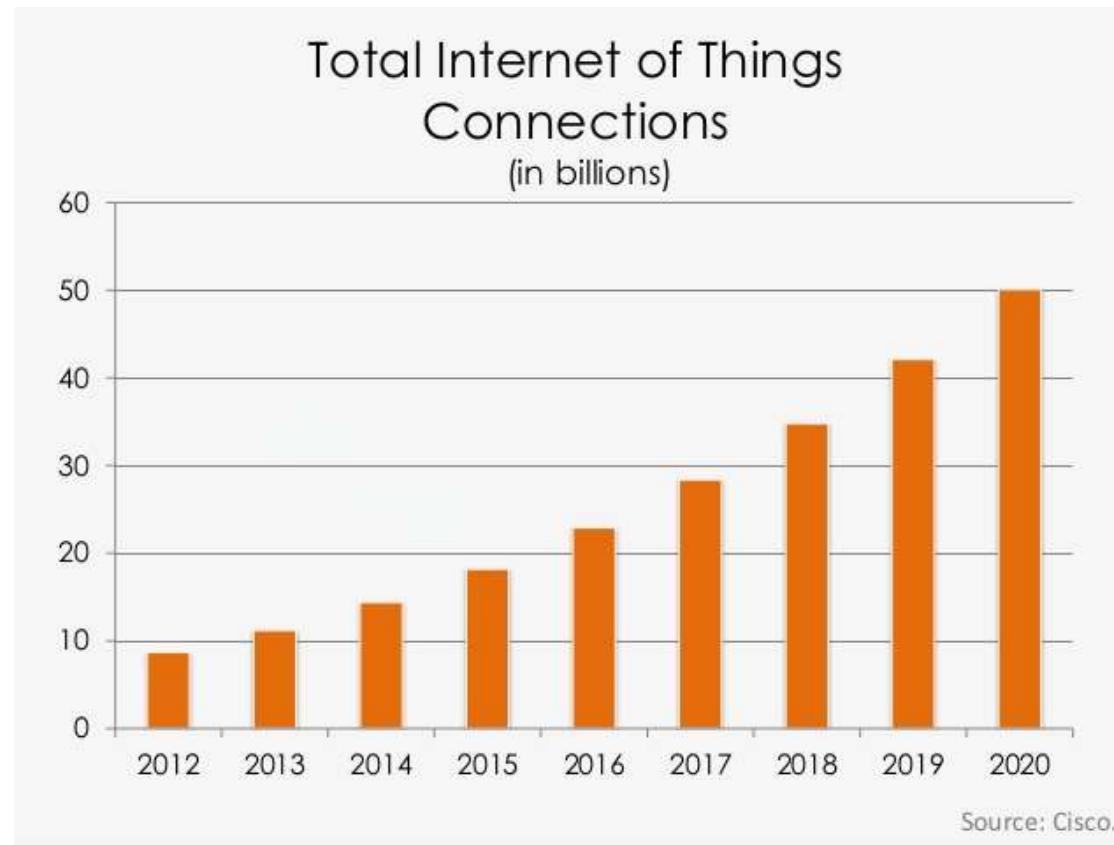
*Source : L'Internet des objets de Pierre-Jean Benghozi, Sylvain Bureau et Françoise Massit-Folléa (Edition MSH)*

- **M2M** : machine to machine, échange d'informations entre deux machines sans intervention humaine.

## 2- Le marché de l'IoT

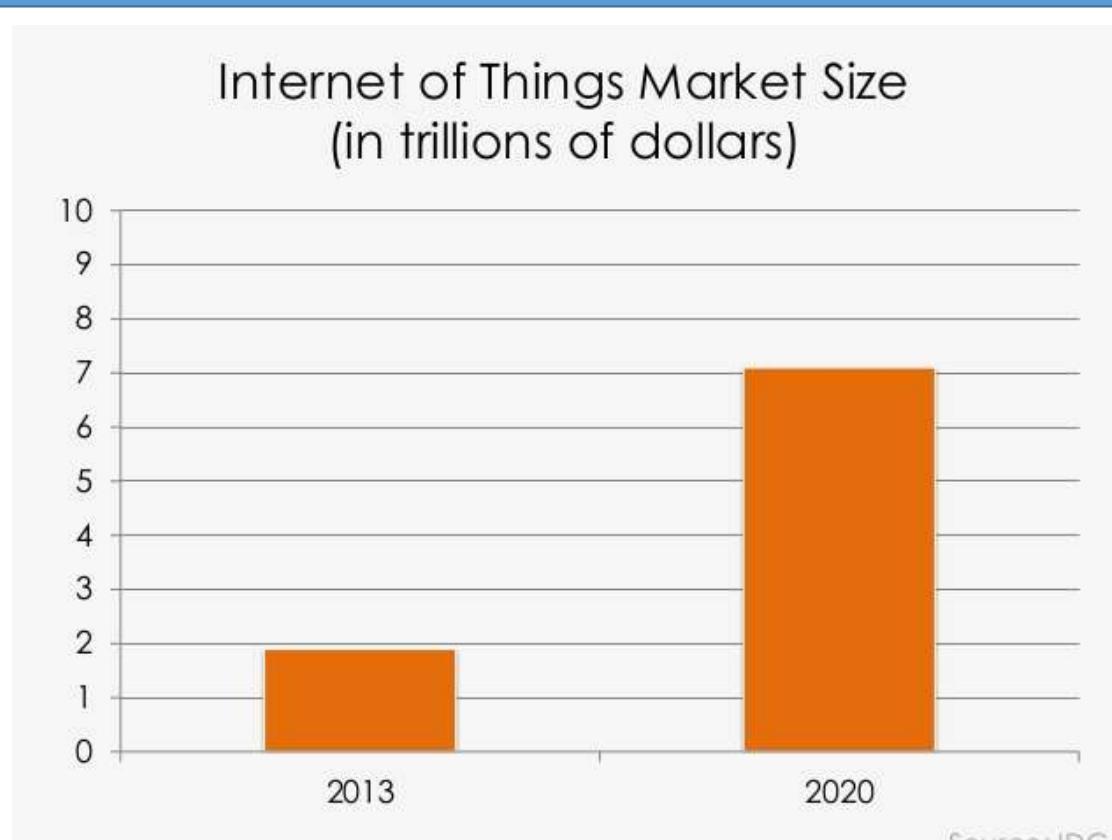
# Le marché de l'IoT

---



# Le marché de l'IoT

---



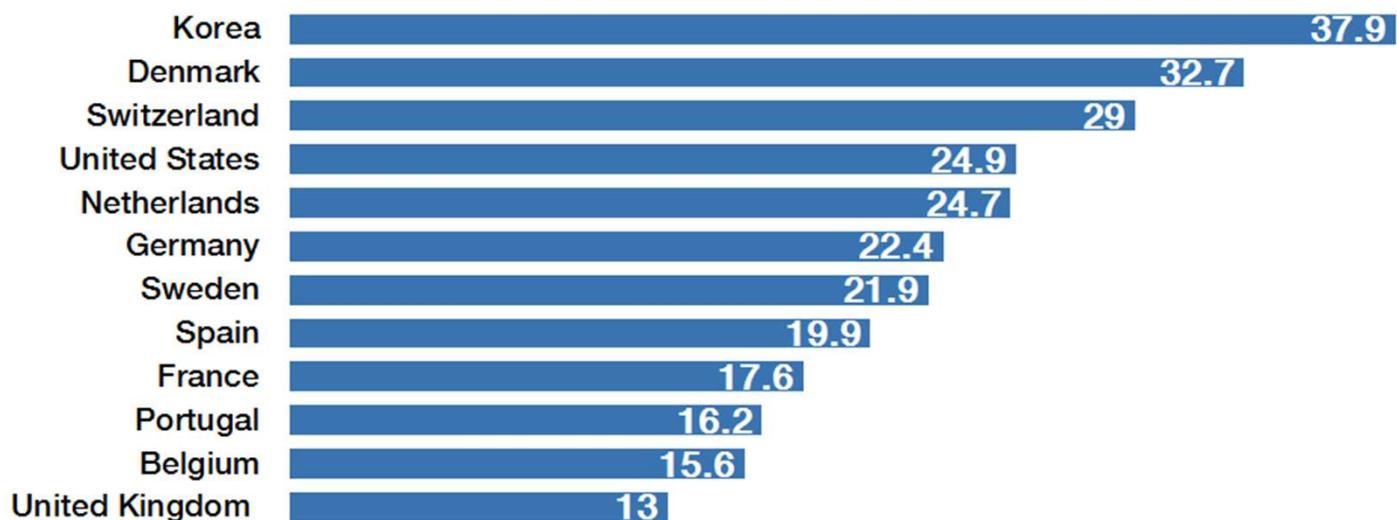
Rappel : 1 trillion = 1000 milliards

# Le marché de l'IoT

---

## Countries with the most IoT devices

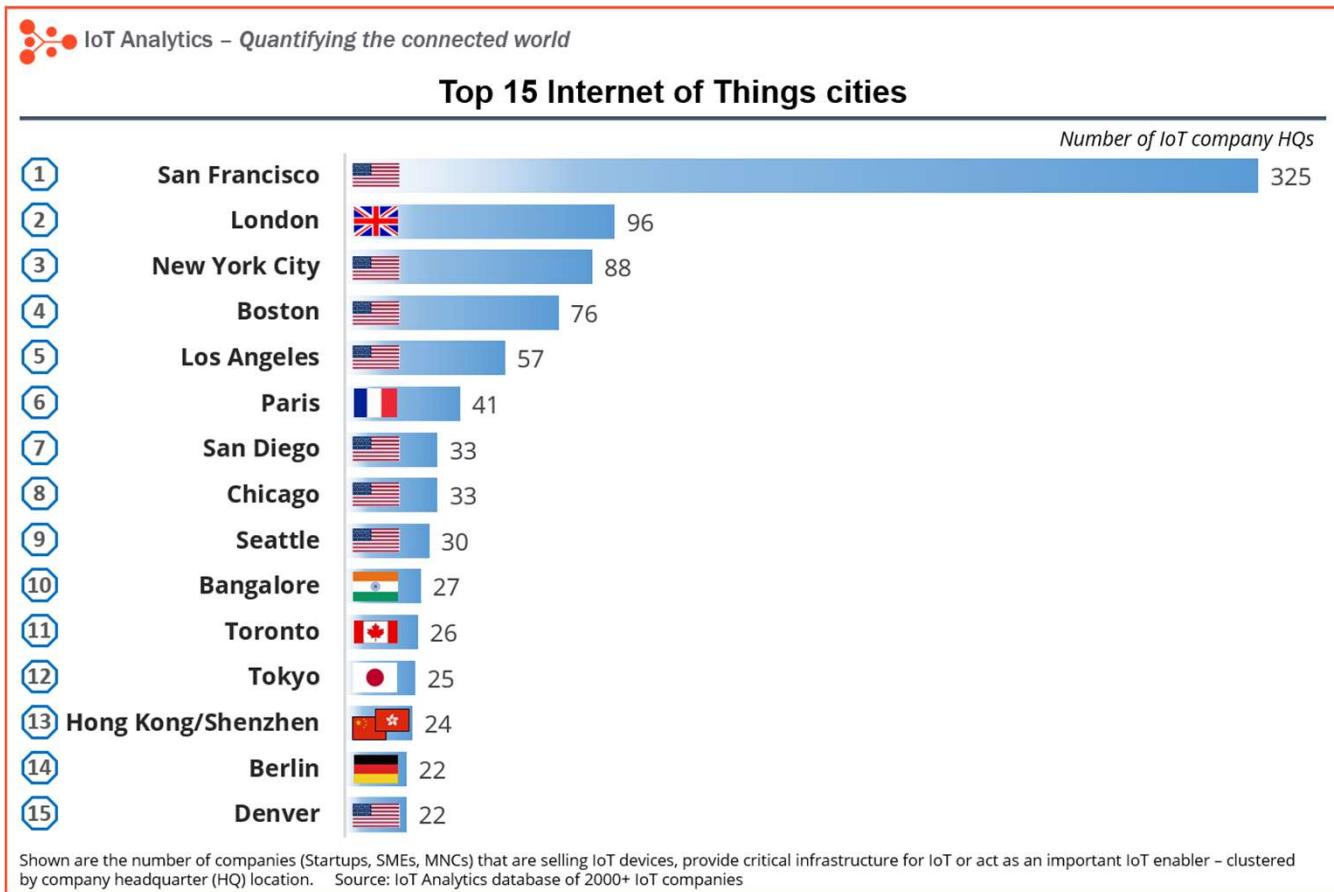
Devices online per 100 people



Data: Shodan/OECD Source: Quartz

Données publiées en mars 2016

# Le marché de l'IoT

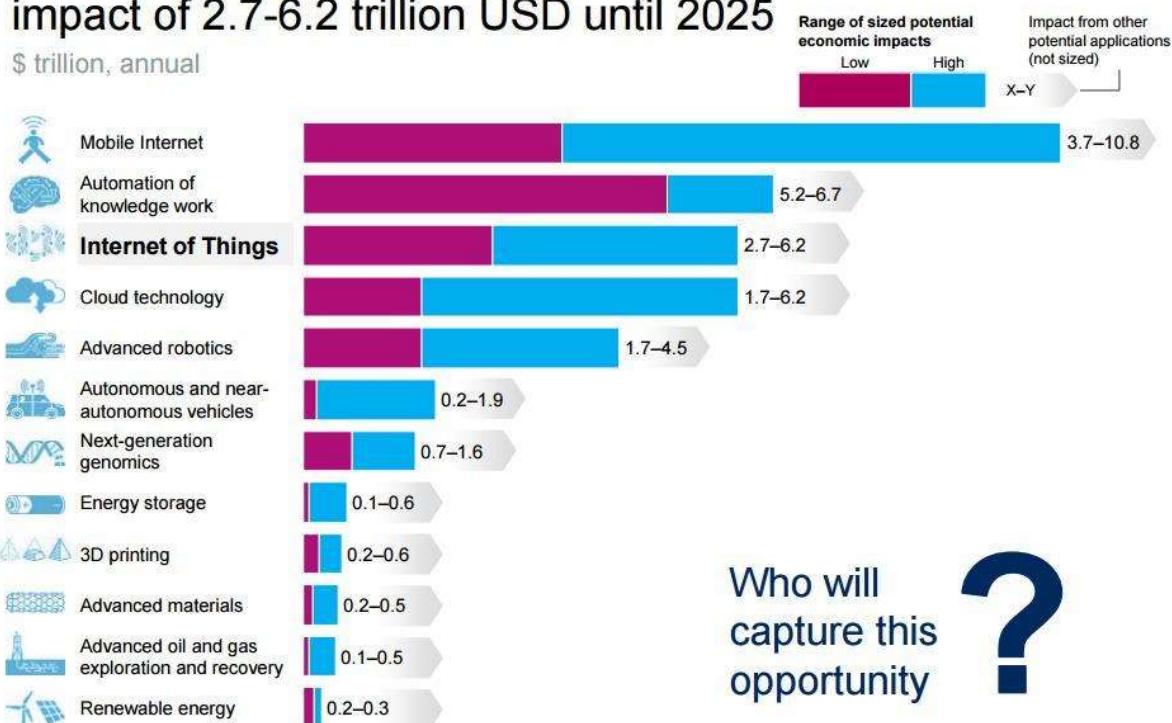


# Le marché de l'IoT

## THE IoT PLATFORM OPPORTUNITY

The Internet of Things (IoT) has a potential economic impact of 2.7-6.2 trillion USD until 2025

\$ trillion, annual



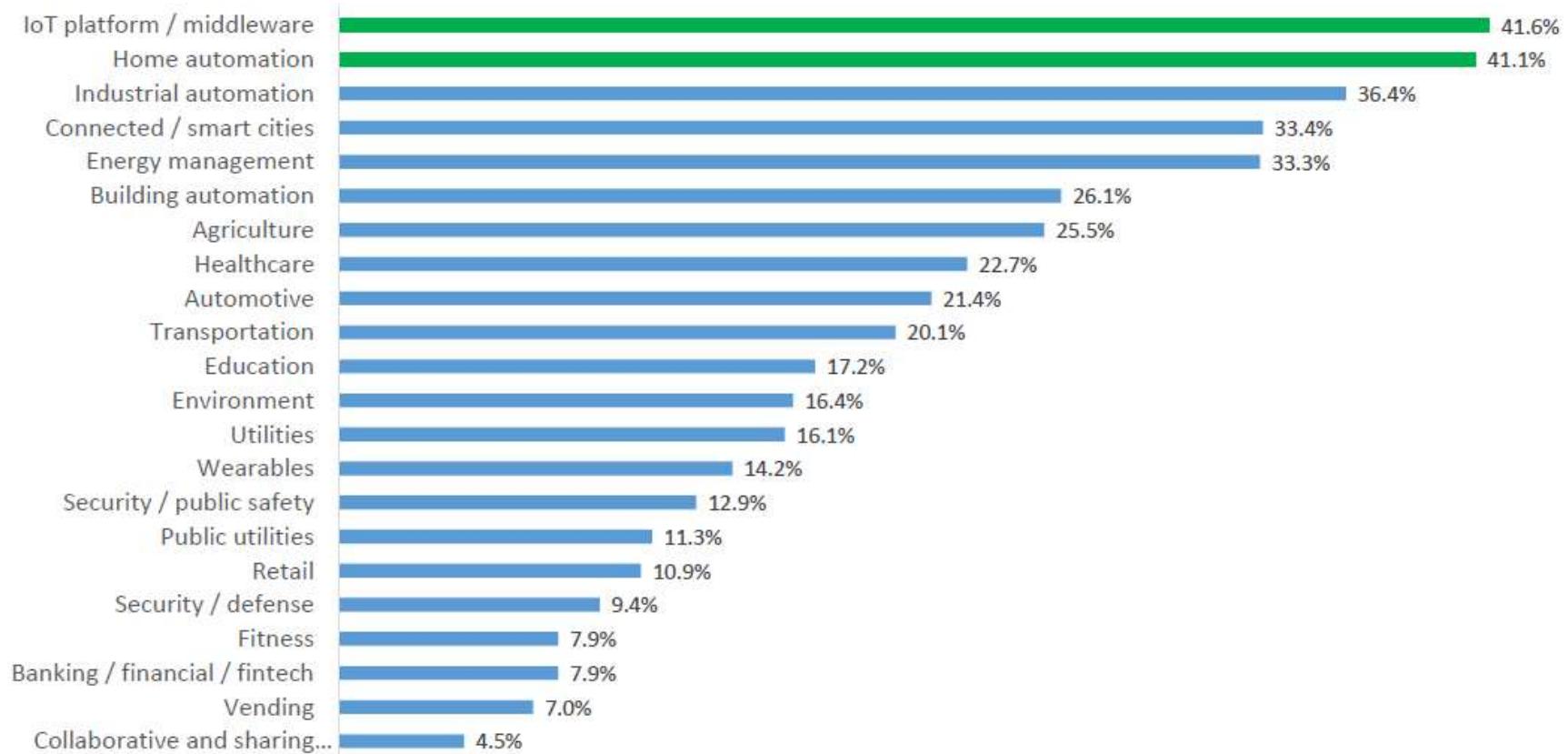
Who will  
capture this  
opportunity



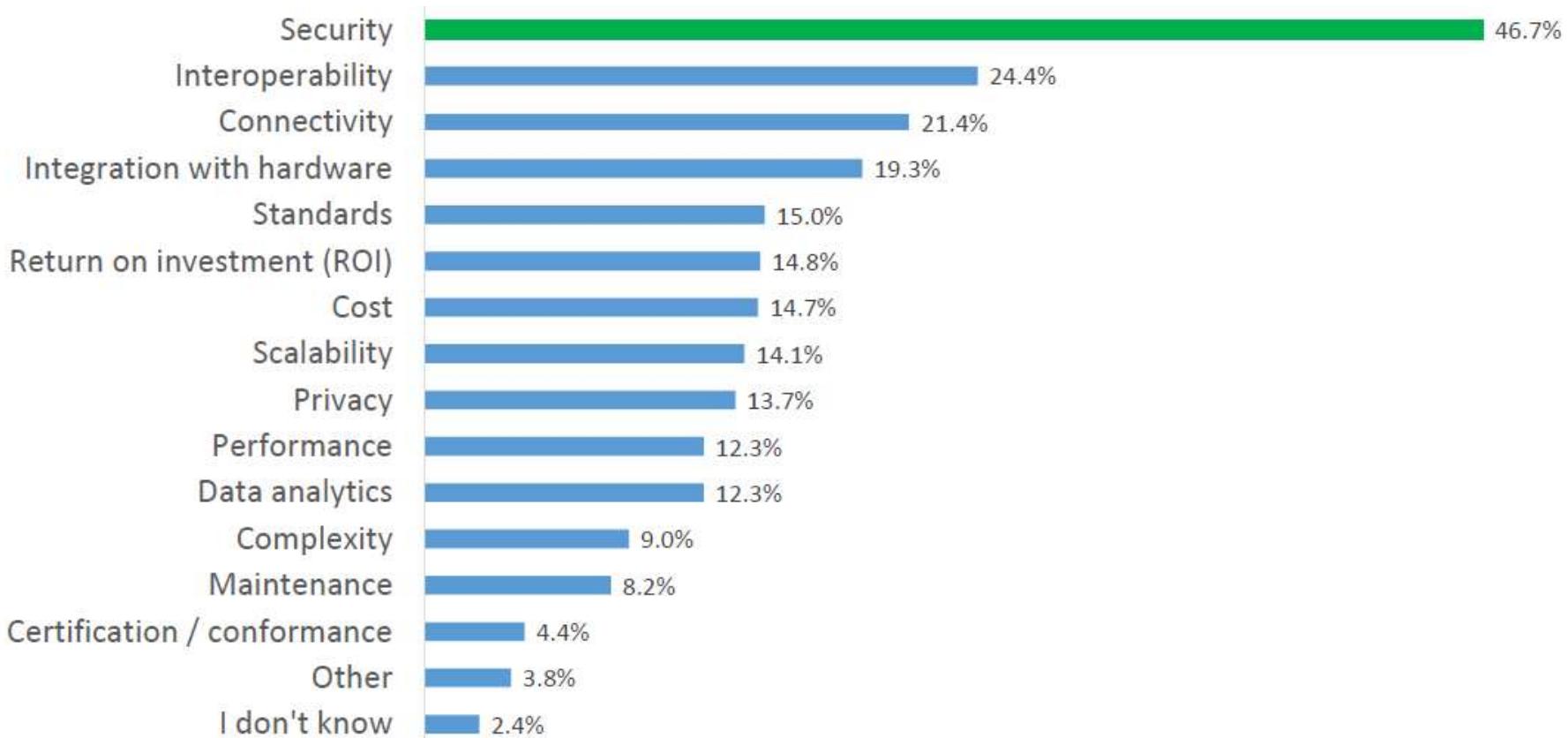
SOURCE: McKinsey Global Institute analysis

McKinsey & Company 3

# Le marché de l'IoT (domaines-visés)

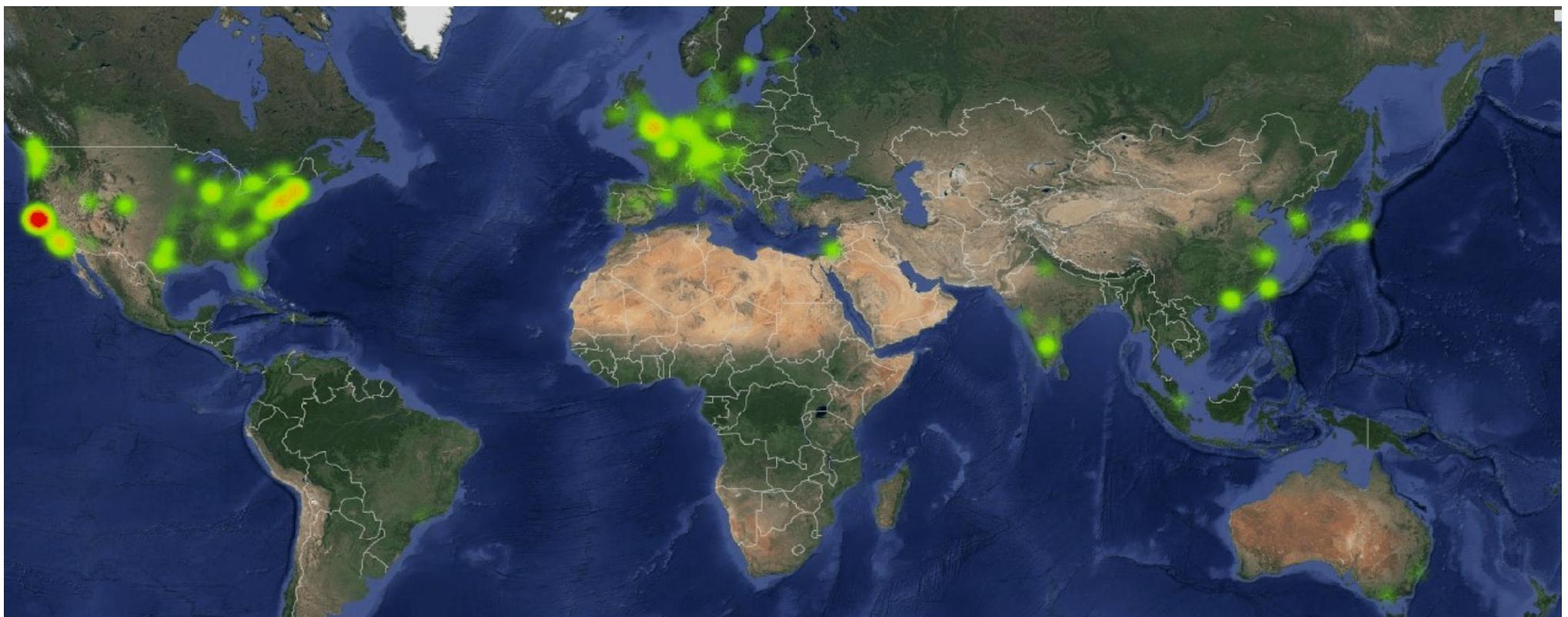


# Le marché de l'IoT (domaines de développement et de recherche)



# Le marché de l'IoT

---



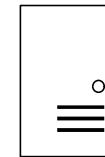
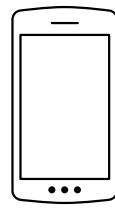
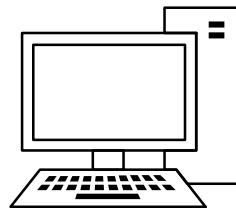
Localisation des principales 2000 entreprises de l'IoT à travers le monde.

# 3- Concepts fondamentaux

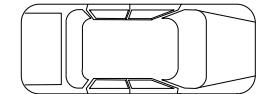
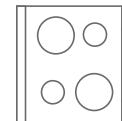
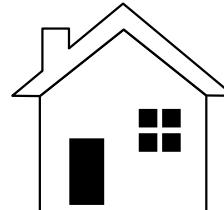
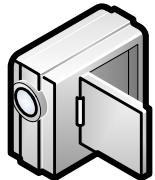
# Quelques objets connectés

---

**Objets « traditionnels » :** ordinateurs, tablettes, smartphones, etc.



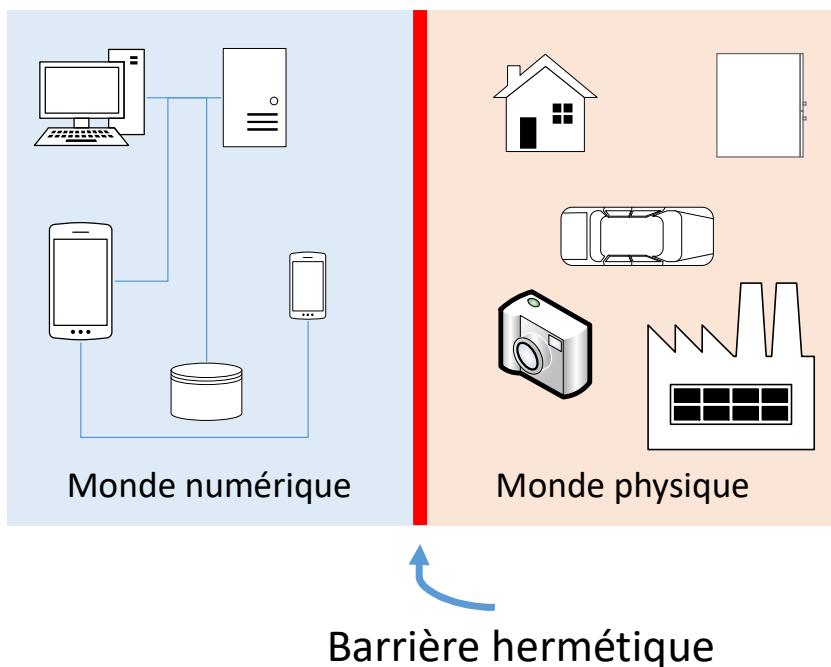
**Nouveaux objets connectés :** appareils électroménagers, instruments de mesure, robots, serrures, machines-outils, bennes à ordures, drones, jouets, montres, véhicules, etc.



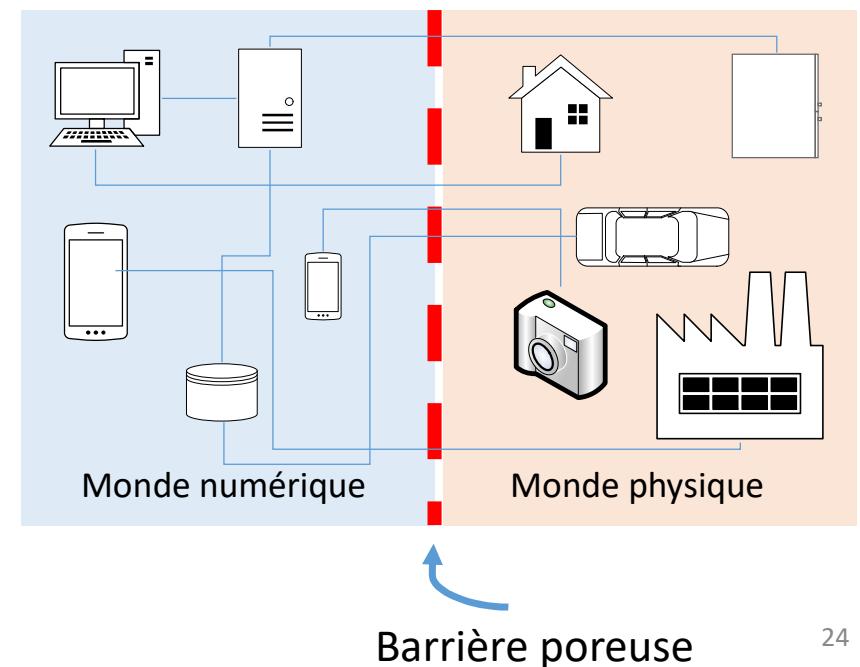
# Concepts fondamentaux (1)

## Jonction entre le monde physique et le monde numérique

### Avant l'internet des objets



### Aujourd'hui



# Concepts fondamentaux (2)

---

**Acquisition de signaux issus du monde physique** (capteurs, transcription de grandeurs physiques en grandeurs électriques, puis numérisation puis transmission vers des systèmes informatiques ou d'autres systèmes physiques).

**Action sur le monde physique** (déclenchement du fonctionnement de dispositifs, chauffage, extinction de feux, ouverture d'une porte, mise en service d'une machine, régulation d'une grandeur physique, exécution d'une tâche robotique, etc.) L'ordre peut provenir d'un système informatique ou d'autres objets physiques connectés.

# Concepts fondamentaux (3)

---

**La mise en réseau de plusieurs objets connectés apporte de nouvelles fonctionnalités de mesure de l'environnement et d'actionnement collectif.** Par exemple : collaboration entre plusieurs objets pour exécuter une tâche qu'un objet ne peut réaliser seul. Exemple : le relevé des températures dans une forêt en plusieurs points permet de prévenir les départs d'incendie et éviter les cas aberrants.

# Concepts fondamentaux (4)

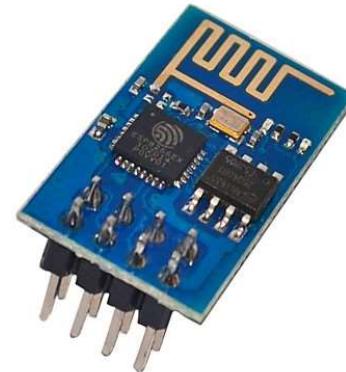
---

**Une problématique de sécurité accrue : en effet, le risque dépasse largement le vol de données.** Par exemple, arrêter le fonctionnement d'une usine ou y provoquer des dégâts matériels ou humains, ouvrir les portes d'une habitation ou d'un magasin pour y effectuer des vols, contrôler à distance un véhicule terrestre ou volant, bloquer le trafic routier de toute une ville, etc. Il convient d'être particulièrement vigilant lors de la conception d'un objet connecté.

# Concepts fondamentaux (5)

---

Aujourd’hui, la baisse des coûts des microcontrôleurs ainsi que des puces de communication sans fil (WiFi, Bluetooth, Zigbee, etc.) permet de mettre une intelligence et des moyens de communication dans beaucoup d’objets de la vie courante ou professionnels. Exemple, systèmes fondés sur la puce esp8266 de ESPRESSIF.



Coût : 3 €

## 4- Interactions entre le « monde numérique » et le « monde physique »

# Monde « numérique » et monde « physique »

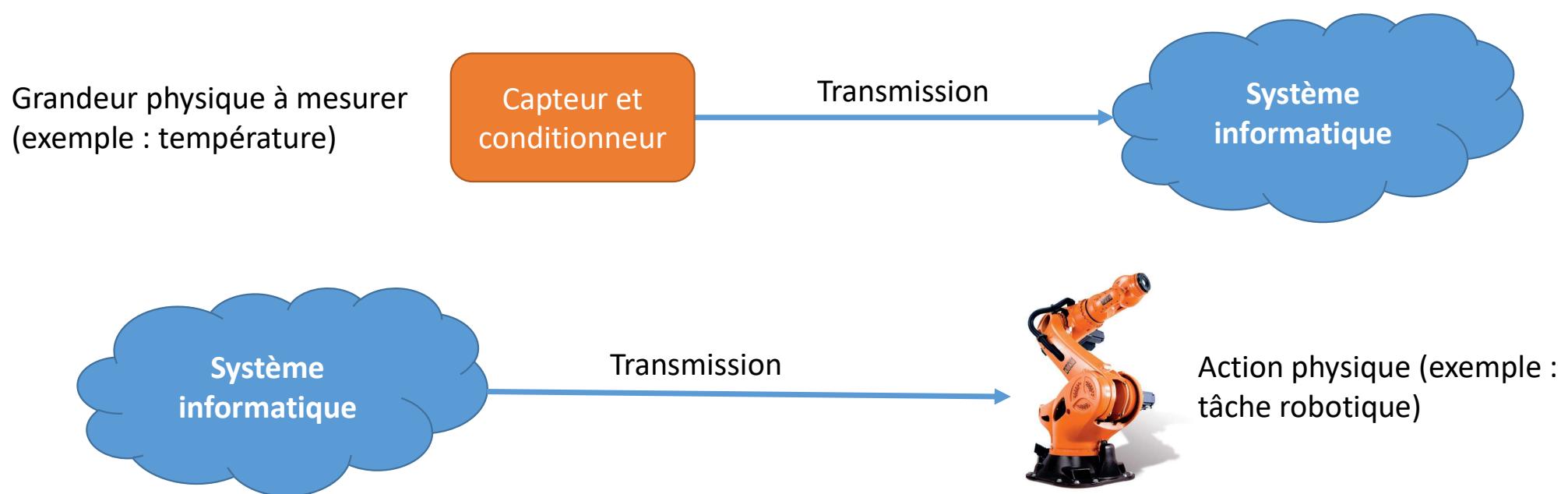
---

- On désignera par « **monde numérique** » tous les **systèmes** manipulant des données numérique ainsi que les réseaux permettant l'échange des données (calculateurs, microcontrôleurs, serveurs informatiques, bases de données, réseaux informatiques, protocoles de communication filaires ou sans fil, etc.).
- On désignera par « **monde physique** » tous les **systèmes vivants ou artificiels** que nous côtoyons et qui interagissent entre eux par divers effets physiques (être humains, animaux, végétaux, véhicules, objets du quotidien, outils, machines diverses, outils de production, etc.). Ces systèmes interagissent entre eux au moyens de grandeurs physiques (forces, déplacements, variation de température ou de pression, voix, son, lumière, etc.)

**Attention !** Certains objets du « monde numérique » sont bien « physiques » mais ils manipulent essentiellement des données (un ordinateur par exemple) et on s'intéresse généralement à leur aptitude à traiter des données.

# Enjeu majeur de l'IoT : comment faire interagir ces deux mondes ?

- Pour cela, il convient de mettre en œuvre des moyens permettant à **une grandeur physique de renseigner un système informatique** et, inversement, des moyens permettant à **un système informatique d'agir sur le monde physique** (c'est-à-dire : changer son état).



# Exemple : allumage et extinction d'une LED

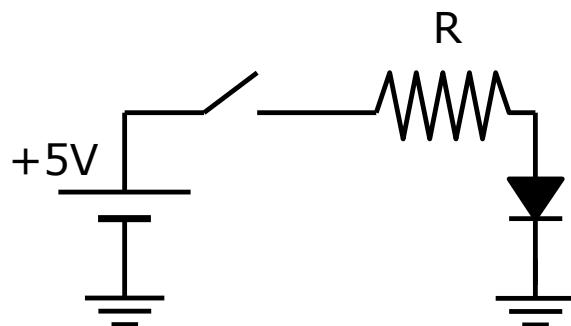
---

- Une LED (Light Emitting Diode) ou DEL (Diode ElectroLuminescente) est un composant électronique très utilisé dans les appareils électroniques comme indicateur ou afficheur.

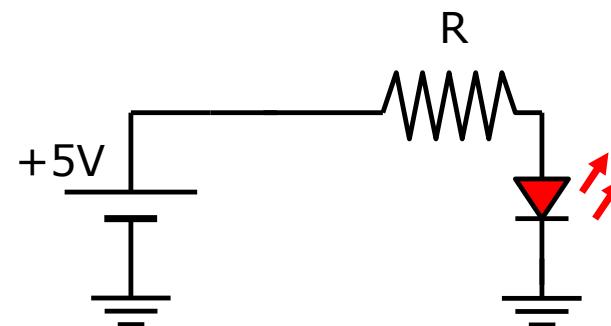


# Exemple : allumage et extinction d'une LED

- L'allumage d'une LED s'effectue en appliquant à ses bornes une tension électrique à travers une résistance de limitation de courant.



LED éteinte  
(état 0)

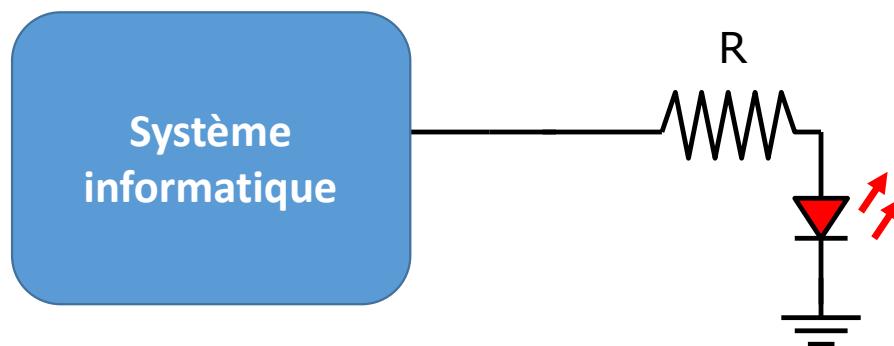


LED allumée  
(état 1)

# Exemple : allumage et extinction d'une LED

---

- Pour réaliser ce fonctionnement à l'aide d'un système informatique, il convient d'utiliser un dispositif d'entrée/sortie (E/S).

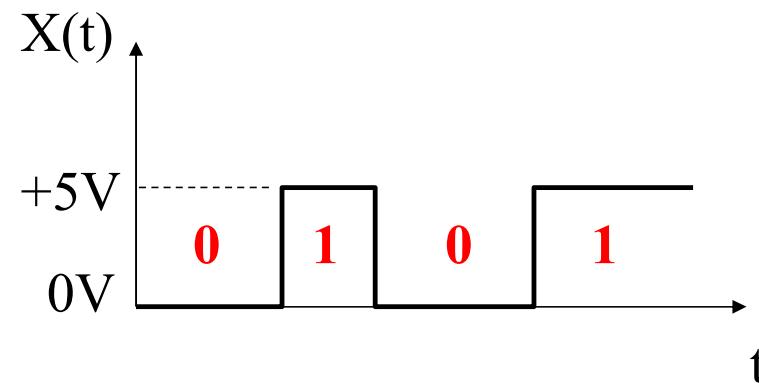


Le système informatique pilote l'allumage et l'extinction de la LED  
par application de deux niveaux de tension électrique

# Exemple : allumage et extinction d'une LED

---

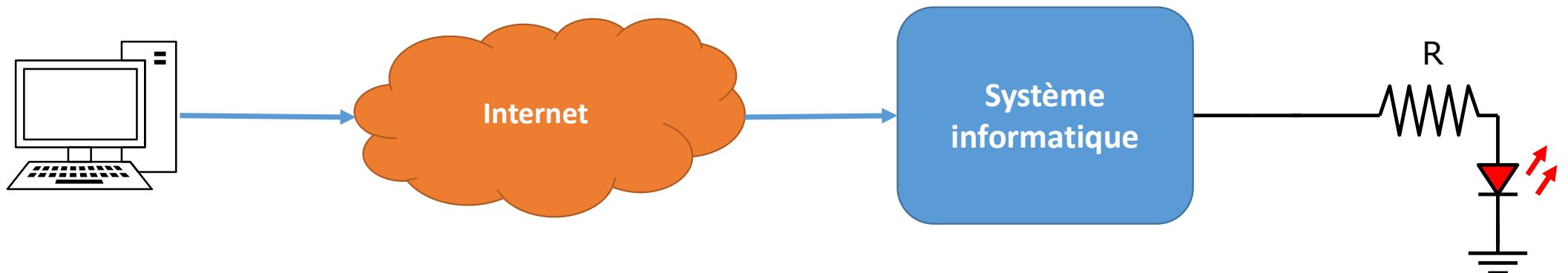
- Représentation physique des états logiques
- Les états logiques sont matérialisés par des niveaux de tensions 0V et +5V (ou 0V et 3,3V).



# Exemple : allumage et extinction d'une LED

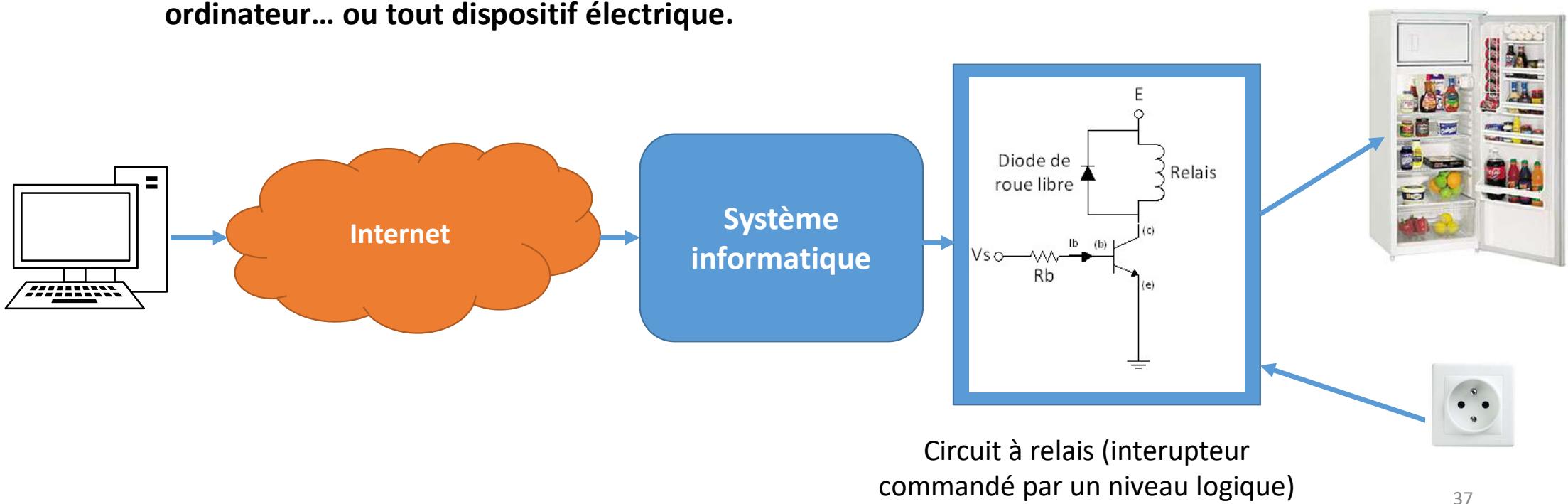
---

- Idée : comment commander l'état de la LED à distance (par exemple via le réseau internet) ?



# Extension : déclenchement d'un dispositif quelconque à distance

- La commande d'une LED à distance ne présente qu'un intérêt limité. Cependant, il est possible d'étendre ce concept pour déclencher divers dispositifs à distance : **éclairage, ventilation, climatisation, moteur, déverrouillage ou ouverture de sorties de secours, allumage d'un ordinateur... ou tout dispositif électrique.**



# Interactions : capteurs et actionneurs

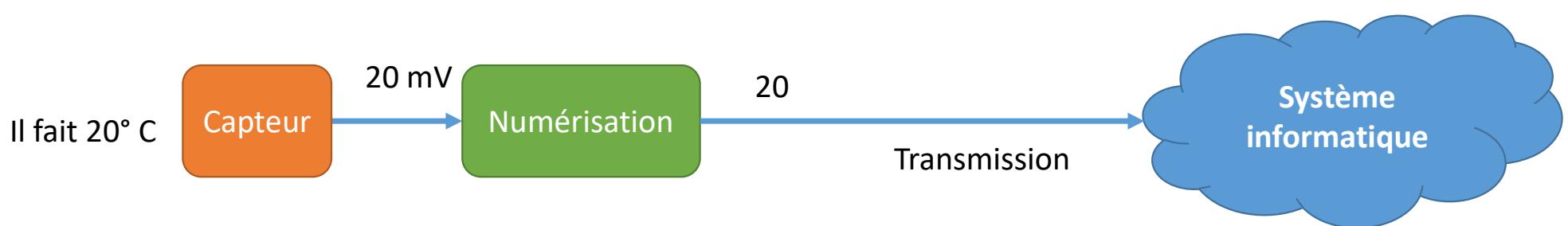
---

- De manière générale, l'IoT met en œuvre deux types d'éléments pour interagir avec le monde physique : des **capteurs** et des **actionneurs**.
- **Les capteurs** permettent de recueillir des informations depuis le monde physique et de les transmettre vers le système informatique.
- **Les actionneurs** permettent au système informatique d'agir sur le monde physique en modifiant son état.

# Les capteurs

---

- Ils permettent de **traduire une grandeur physique en un signal électrique**. Ce dernier est ensuite **numérisé** pour être **transmis** au système informatique.
- Par exemple : un capteur de température permet de traduire l'amplitude de la température en une tension électrique. Cette dernière est numérisée puis transmise.



# Les capteurs

---

Grandeur communément mesurées :

- Systèmes à deux états (0,1), (fermé,ouvert) (éteint,allumé), etc.
- Comptage d'impulsions (tachymètre), cardio-fréquencemètre
- Température
- Pression
- Luminosité
- Position
- Vitesse
- ...

# Les actionneurs

---

- Ils permettent d'**agir dans le monde physique**, c'est-à-dire, **changer son état**.
- Par exemple : un actionneur peut allumer un appareil à distance (voir l'exemple donné dans les transparents précédents).

# Les actionneurs

---

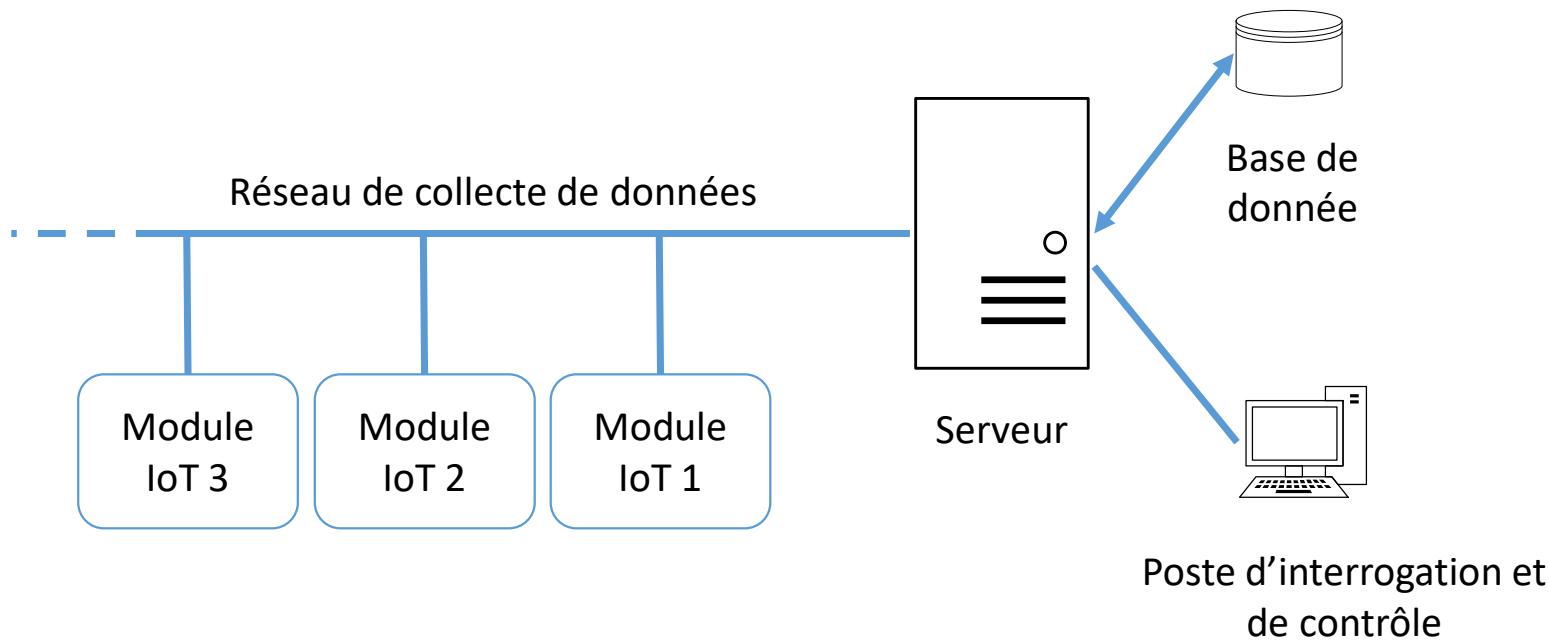
Actionneurs couramment utilisés :

- Allumage d'un éclairage
- Déclenchement d'un avertisseur sonore
- Allumage d'une machine
- Génération de mouvements (ex. servomoteur)
- Commande de robots
- Commande de moteurs (à courant continu, pas-à-pas, etc.)
- Contrôle de débits (air, pression, liquides, etc.)
- ...

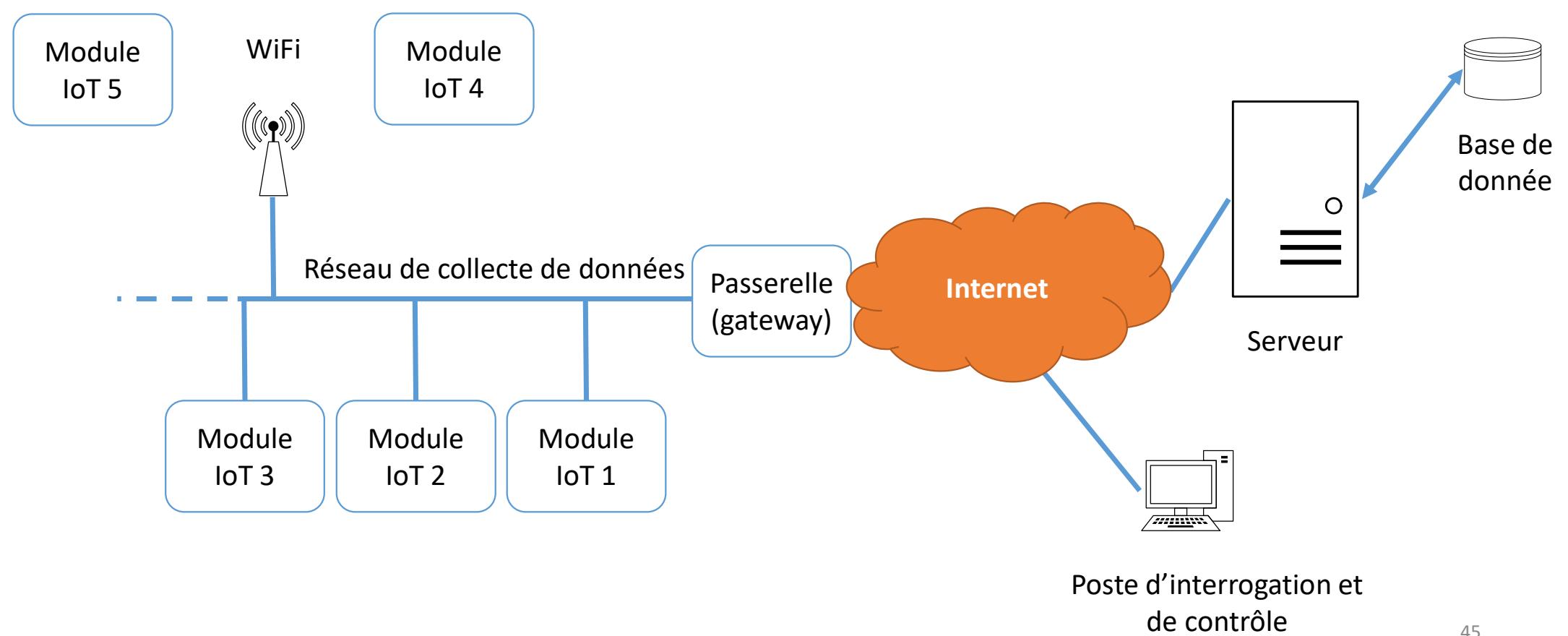
# 5- Infrastructures pour l'IoT

# Infrastructure élémentaire

---

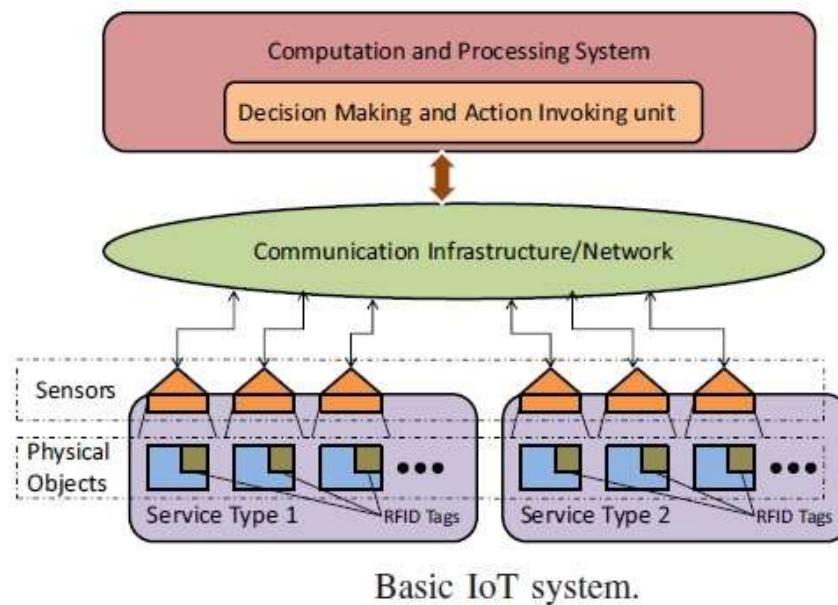


# Exploitation du réseau internet



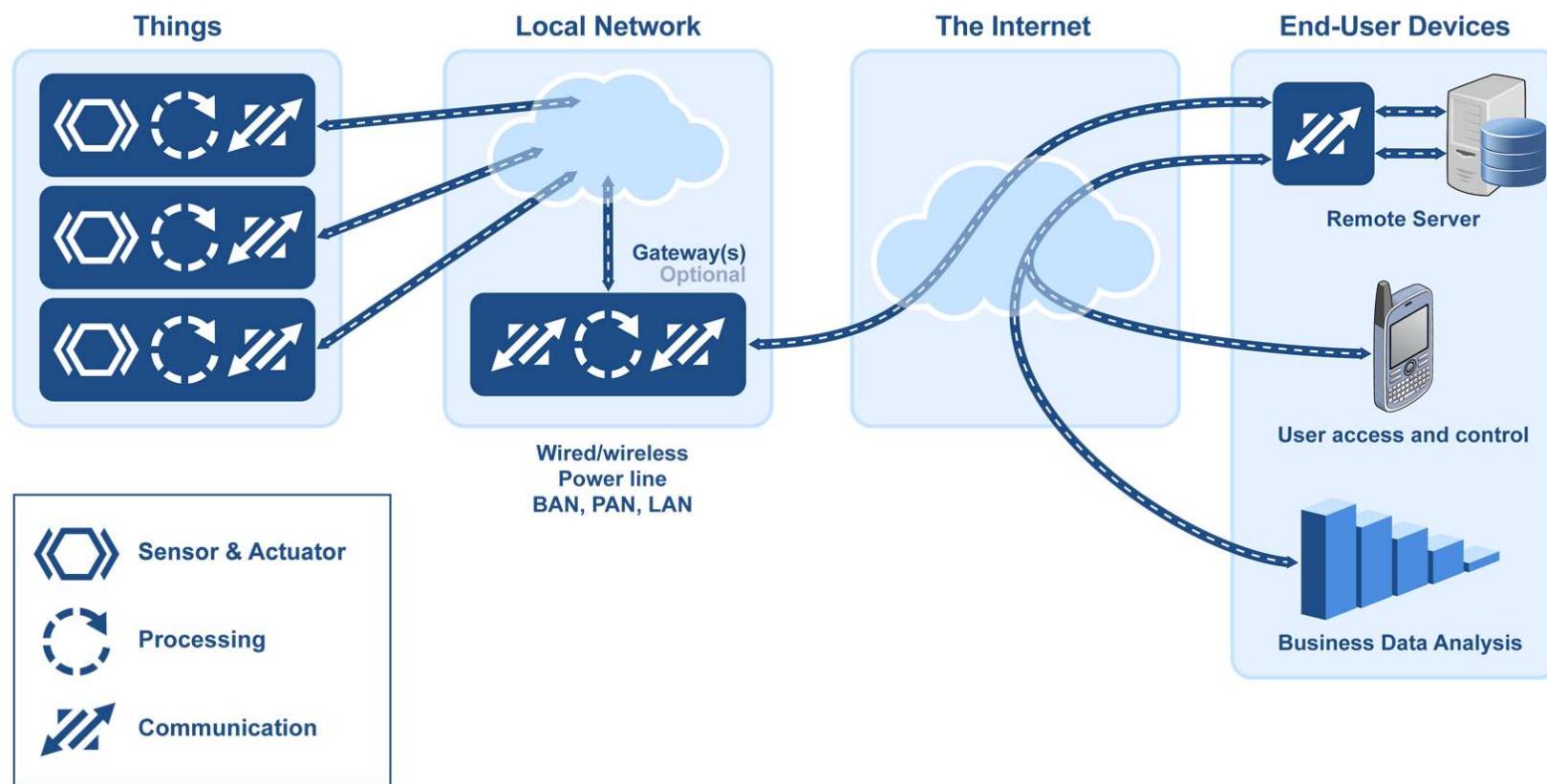
# Exploitation du réseau internet

---



# Exploitation du réseau internet

IoT is Made of Embedded Devices

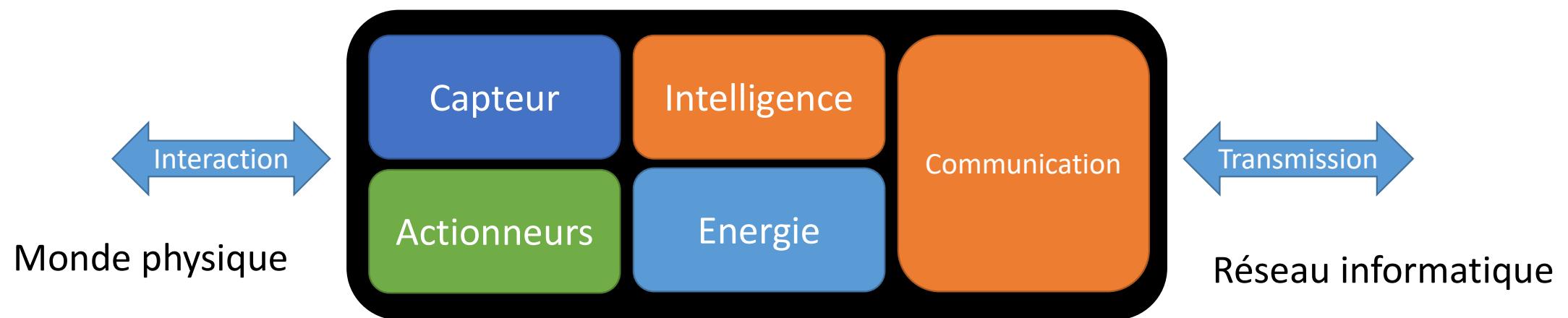


# 6- Solutions technologiques pour les plateformes embarquées

# Solutions technologiques

---

Caractéristiques générales d'une plateformes pour l'IoT



# Solutions technologiques

---

## Caractéristiques générales d'une plateformes pour l'IoT

Capteur

Traduction d'une  
grandeur physique en un  
signal électrique

Energie

Alimentation de la  
plateforme en énergie  
électrique. Doit être  
adaptée à l'application

Actionneurs

Modification de l'état de  
l'environnement

Intelligence

Traitement local des  
données (simple ou  
complexe)

Communication

Codage et transmission  
des données, protocoles  
standards ou dédiés,  
communication filaire ou  
sans fil.

# Solutions technologiques

---

Avant de se lancer dans la réalisation d'un objet connecté, il convient de bien choisir la plateforme technologique à utiliser.

## Caractéristiques générales d'une plateformes pour l'IoT

- Quelle puissance de calcul ? (quels sont les traitements et calculs à effectuer en local ?)
- Quelle quantité de mémoire ? (que doit-on stocker en local ?)
- Quelles interactions avec le monde physique ? (capteurs - actionneurs)
- Quelle autonomie énergétique ? (durée d'utilisation, accessibilité, usage de batteries, panneaux solaires, etc.)
- Quelles caractéristiques logicielles ? (programmation simple ou complexe, accès distant, OS, etc.)
- Quels protocoles de communication ? (protocoles standards ou dédiés, liaison filaire ou sans fil, cryptage, etc.)
- Quel coût ? (nombre d'objets à réaliser, budget dédié à la partie IoT, degré de fiabilité requis, etc.)
- ...

# Solutions technologiques

---

La révolution des systèmes embarqués à faible coût :

Deux approches majeures :

- Systèmes construits autour d'un OS embarqué (RASPBERRY PI, BEAGLEBONE, et plateformes similaires)
  - **Avantages** : ouverts, puissants, langages de programmation multiples,
  - **Inconvénients** : parfois complexes à mettre en œuvre, prise en main longue, réactivité moyenne, coût relativement élevé, interfaçage plus difficile.
- Systèmes dédiés compacts à logiciel propriétaire (ARDUINO, GENUINO, INTEL GALILEO, ESP8266, ESP32, etc.)
  - **Avantages** : Très réactifs, très faible coût, fonctionnement plus robuste (pas de couches logicielles), interfaçage aisné, prise en main très rapide.
  - **Inconvénients** : moins puissants, langages de programmation plus limités, moins flexibles sur le plan logiciel.

# Solutions technologiques

---

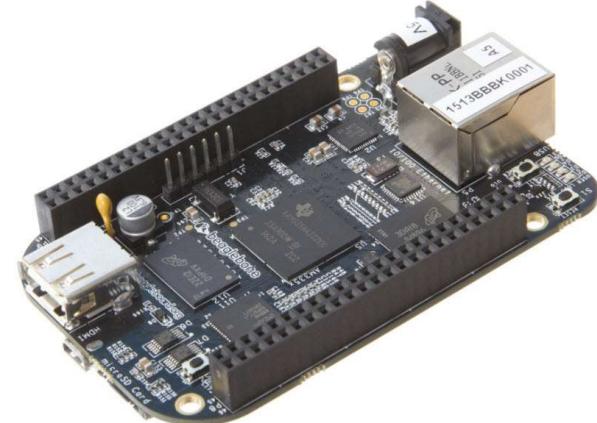
Quelques dignes représentants...



Arduino + shield



Raspberry Pi



BeagleBone

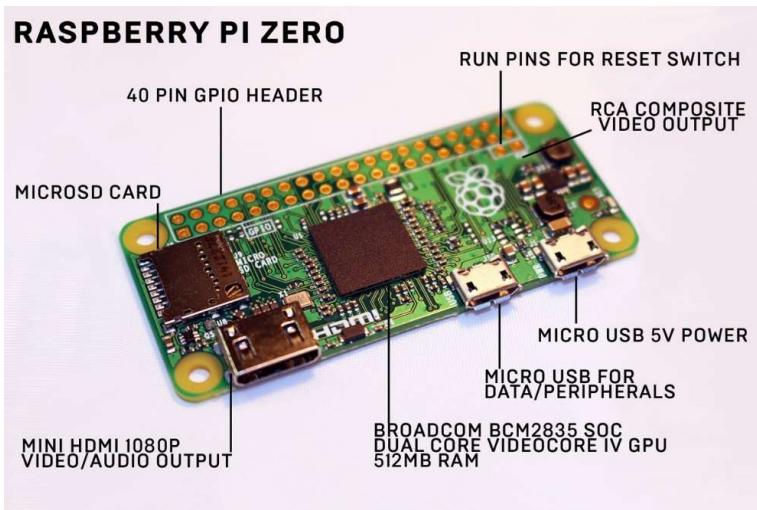
# Solutions technologiques

---

| NOM DE LA CARTE                           | ARDUINO UNO                                    | BEAGLEBONE   | RASPBERRY PI (MODEL B)   |
|---|--|--|--|
| Origine                                   | Interaction Design Institute d'Ivrea (Italie)  | Projet de Hardware Open Source piloté par Texas Instruments                                | Université de Cambridge  |
| Organisation en charge des spécifications | Arduino.cc                                     | BeagleBoard.org  | Raspberry Pi Foundation (fondation de droit anglais)                                   |
| Naissance                                 | 2005 (fabrication en Italie par Smart Projets) | 2008 (BeagleBoards) - 2011 (BeagleBone) (accord de fabrication/distribution avec Digi-Key) | 2008 (accord de fabrication avec RS Components et Farnell/Element 14 en 2011)          |
| Prix                                      | 30\$   | 90\$(45 \$ pour le BeagleBone Black)   | Moins de 40\$  |
| Taille                                    | 45,43x32,34mm                                  | 86,36x53,34mm (bords arrondis)   | 85,60x53,98mm  |
| Processeur                                | ATmega328 8 bits d'Atmel à 16MHz               | Sitara 335x de TI basé sur un Cortex-A8 à 720MHz (1 GHZ pour la BeagleBone Black)          | BCM2835 de Broadcom basé sur un ARM11 à 700MHz GPU intégrée (Video Core 4 de Broadcom) |
| Mémoires                                  | 2 Ko Ram, 1 Ko Eeprom                          | 256Mo DDR2 (512Mo DDR3 pour la BeagleBone Black)   | 512Mo Sdram  |
| Mémoire Flash                             | 32 Ko  | Sur MicroSD (4 Go)   | Sur carte SD   |
| Tension d'entrée                          | 7V - 12V                                       | 5V – 3,3V  | 5V   |
| Consommation                              | 42 mA (0,5W)                                   | 210 à 450mA (2,5W max.)  | 700 mA (3,5W)  |
| Ethernet                                  | Non  | 10/100 Ethernet  | 10/100 Ethernet  |
| USB                                       | Non  | 1 USB 2.0  | 2 USB 2.0  |
| Sorties vidéo                             | Non  | Non (micro HDMI pour la BeagleBone Black)  | Composite et HDMI  |
| Développement                             | Langage de programmation Arduino               | Environnement BoneScript. Langages Phyton, Scratch, Squeak                                 | Langages Scratch, Squeak   |

# Solutions technologiques

Les nouveaux... encore plus petits, encore moins chers



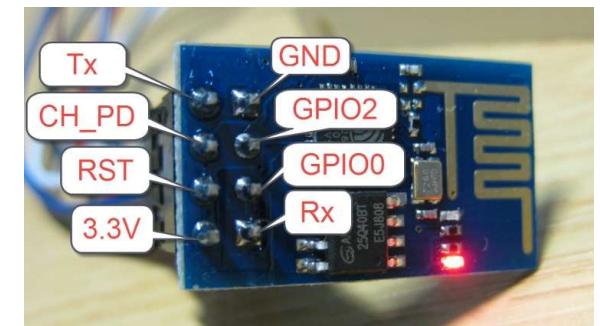
Raspberry Pi zero (5\$)

ESP8266 (Espressif) (2€)

# Solutions technologiques

## Une application complète : caractéristiques

- Processeur : Tensilica L106, 32 bits
- Horloge processeur : 80 / 160 MHz
- Mémoire RAM : 32 + 80 Ko
- Mémoire flash : max. 16 Mo, typique 512 Ko ou 1 Mo
- Alimentation : 3,3 V, ne tolère PAS le 5 V
- Consommation : 10 µA à 170 mA
- Lignes de port GPIO : 17, avec les fonctions I<sup>2</sup>C, SPI, UART, MLI multiplexées
- Port analogique : 1, résolution de 10 bit, plage 0 à 1 V
- Compatible 802.11 : b/g/n/....
- Liaisons TCP simultanées : max 5
- Modes disponibles : station, point d'accès, station + point d'accès
- Switch transmetteur embarqué, balun, LNA, Ampli de puissance, circuit d'adaptation.
- Circuits de gestion de la puissance.



# Solutions technologiques

## Une application complète : programmation

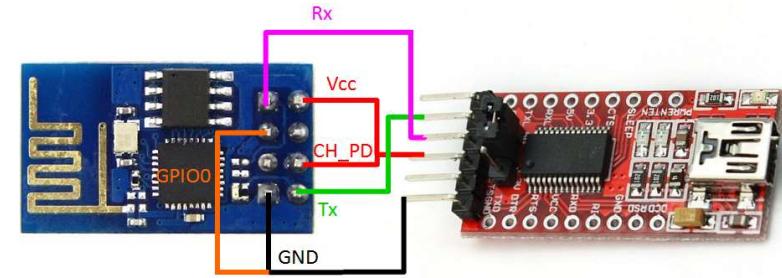


```
#include <ESP8266WiFi.h>

// Spécifier le SSID du point d'accès Wifi
const char* ssid = "your-ssid";
// Spécifier le mot de passe associé
const char* password = "your-password";
// Crée le serveur Web en spécifiant le port TCP/IP
// 80 est le port par défaut pour HTTP
WiFiServer server(80);

// Démarrage
void setup() {
    // Communication série 115200
    Serial.begin(115200);
    // Petite pause
    delay(10);
    // GPIO2 en sortie
    pinMode(2, OUTPUT);
    // GPIO2 à la masse
    digitalWrite(2, 0);
    // Deux sauts de ligne pour faire le ménage car
    // le module au démarrage envoie des caractères sur le port série
    Serial.println();
    Serial.println();
    Serial.print("Connexion à : ");
    Serial.println(ssid);
    // Connexion au point d'accès
    WiFi.begin(ssid, password);
    // On boucle en attendant une connexion
    // Si l'état est WL_CONNECTED la connexion est acceptée
    // et on a obtenu une adresse IP
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connecté");
    // Démarrage du serveur Web
    server.begin();
    Serial.println("Serveur démarré");
    // On affiche notre adresse IP
    Serial.println(WiFi.localIP());
}

// Boucle principale
void loop() {
    // Est-ce qu'un client Web est connecté ?
    WiFiClient client = server.available();
    if (client) {
        // Non, on abandonne ici et on repart dans un tour de loop
        return;
    }
}
```



# Solutions technologiques

---

Exemple d'application : brosse à dents connectée

A screenshot of a mobile application interface titled "Real-time Monitor". At the top, there is a green header bar with the text "STATUS Brushing" and a toggle switch icon. Below the header, there is a large, light gray rectangular area. At the bottom, there is a table titled "Devices" with one row of data:

| ID | Device      | Description   | Type      | Last Check-in   | Connected |
|----|-------------|---------------|-----------|---|-----------|
| 1  | Test device | A test device | Test type | Mon Feb 13 2018 06:54:48 GMT+1300 (AUS Eastern Daylight Time) |           |

Suivi de l'usage (enfants par exemple), données commerciales  
(durée du brossage, fréquence...)

# Solutions technologiques

---

Exemple d'application : relevé automatique des compteurs



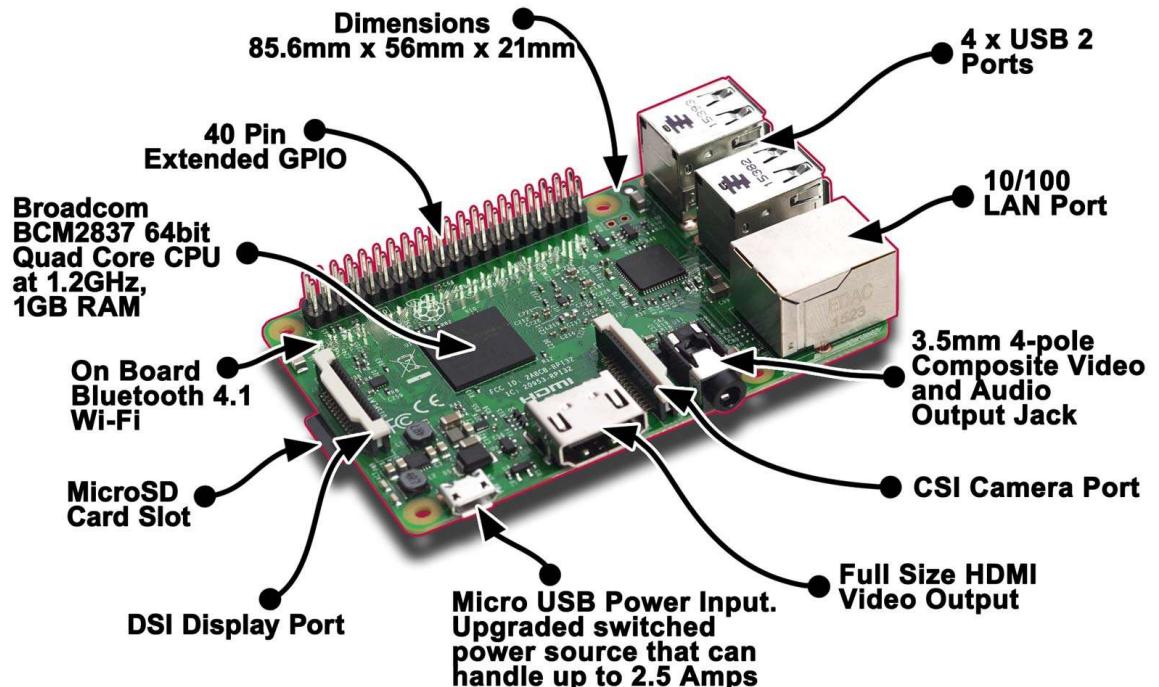
Economies.

Collecte de données.

Surveillance de populations à risques.

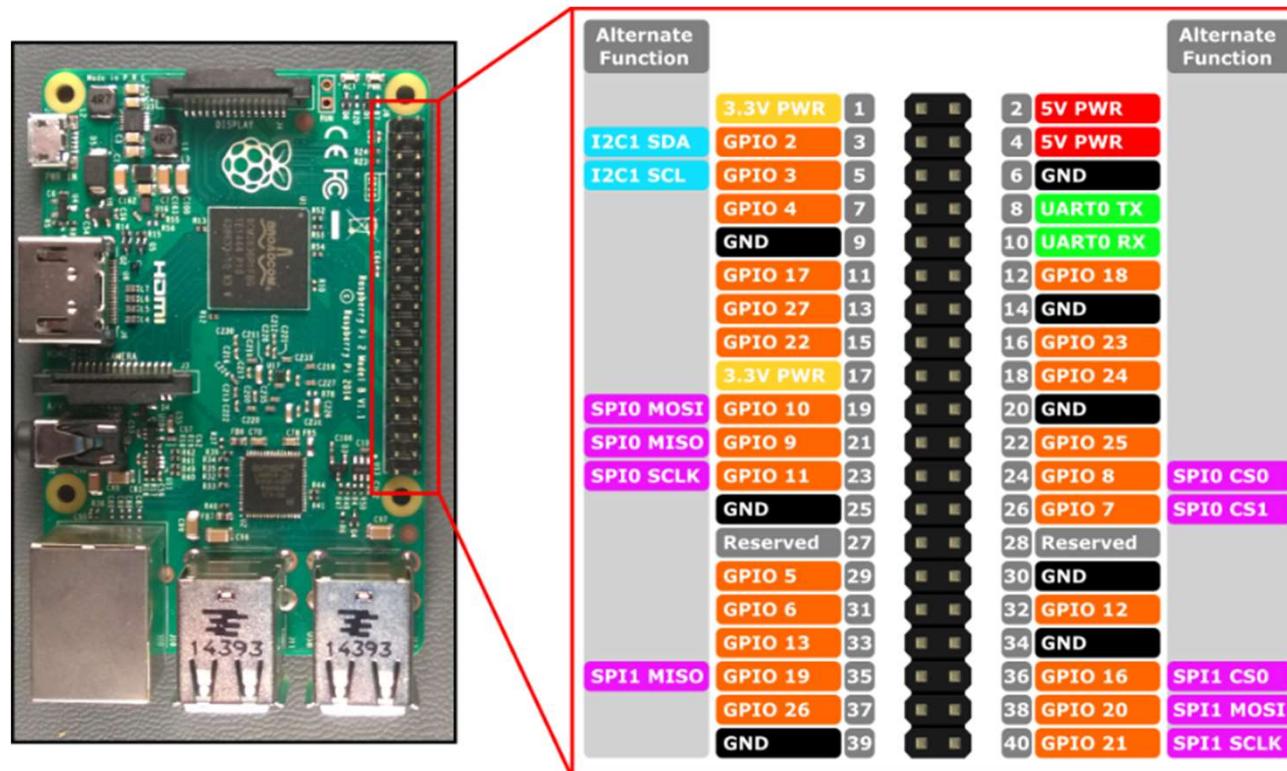
# Raspberry pi 3

|                                |  |
|--------------------------------|--|
| Chipset                        | Broadcom BCM2837 SoC                   |
| Processeur                     | ARM Cortex-A53 1.2 GHz                 |
| Mémoire Vive                   | 1 Go LPDDR2                            |
| Format de carte SD             | Carte MicroSD<br>4x Ports USB 2.0      |
| Connectique                    | 40 GPIO<br>HDMI, audio/vidéo composite |
| Puissance électrique / tension | 600mA jusqu'à 2.5A @ 5V                |
| Connectivité                   | 1x 10/100 Ethernet, Bluetooth, WiFi    |



# Raspberry pi 3

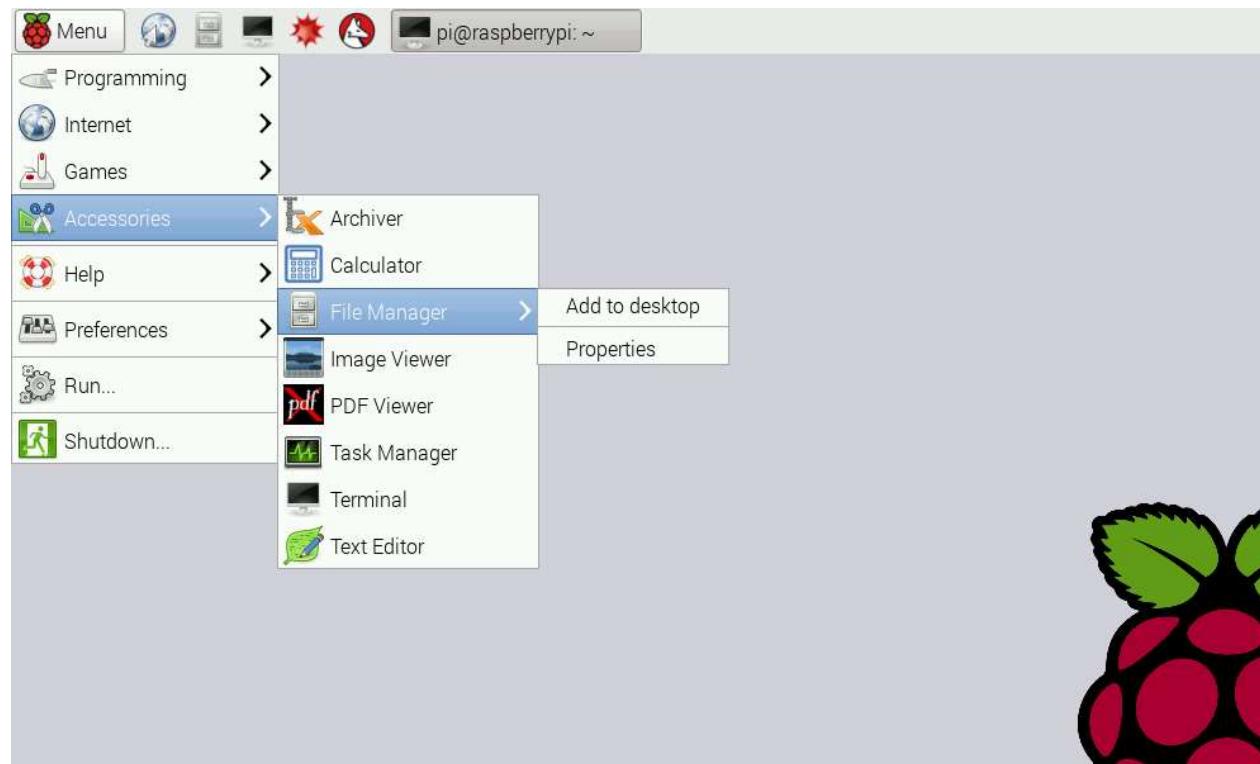
## Connecteur GPIO (General Purpose Input Output)



# Raspberry pi 3

---

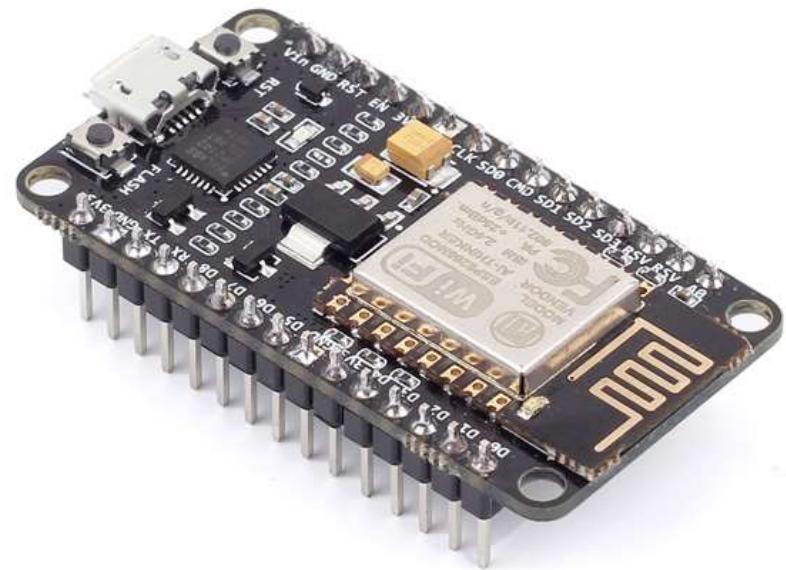
Système d'exploitation : Raspbian



# NodeMCU (basé sur un ESP8266)

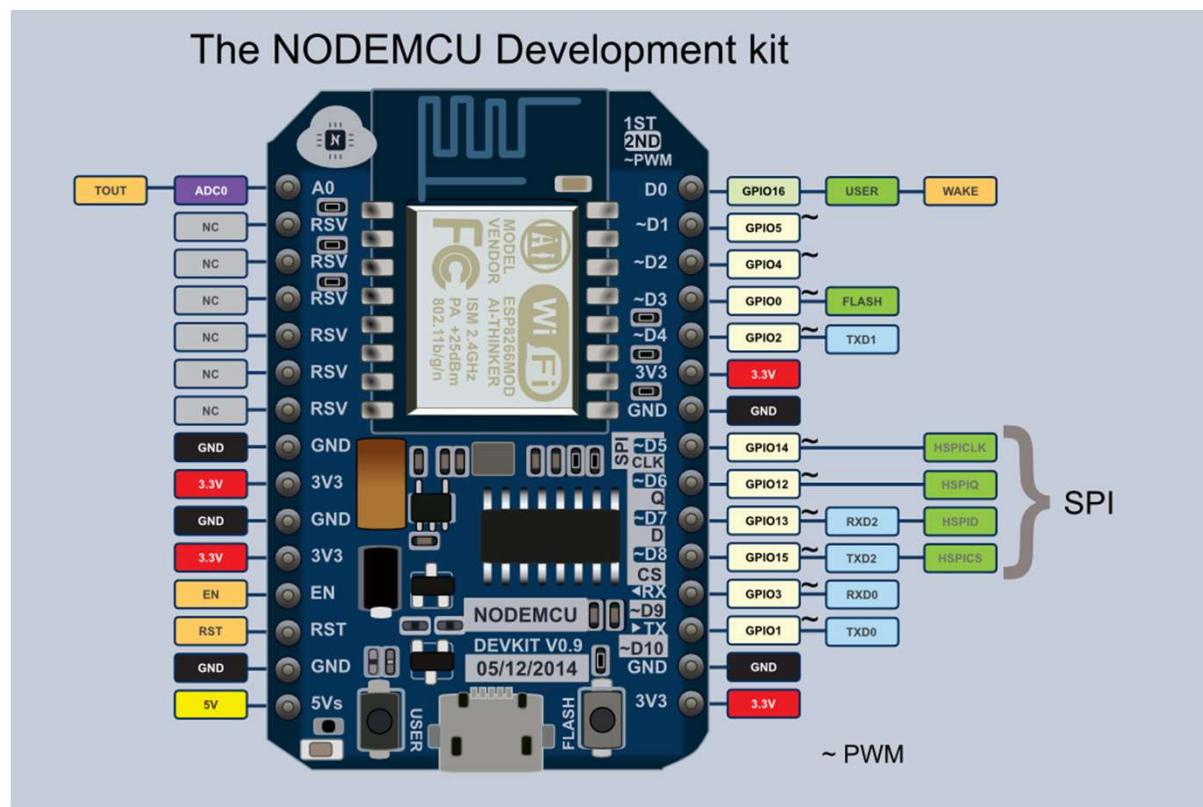
---

- 32-bit RISC CPU: Tensilica Xtensa LX106, 80 MHz
- 64 KiB of instruction RAM, 96 KiB of data RAM
- External QSPI flash - 512 KiB to 4 MiB (up to 16MiB is supported)
- IEEE 802.11 b/g/n Wi-Fi
- Integrated TR switch, balun, LNA, power amplifier and matching network
- WEP or WPA/WPA2 authentication, or open networks
- 16 GPIO pins
- SPI, I<sup>2</sup>C,
- I<sup>2</sup>S interfaces with DMA (sharing pins with GPIO)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 1 10-bit ADC



# NodeMCU (basé sur un ESP8266)

Entrées/sorties d'un module NodeMCU



# NodeMCU (basé sur un ESP8266)

---

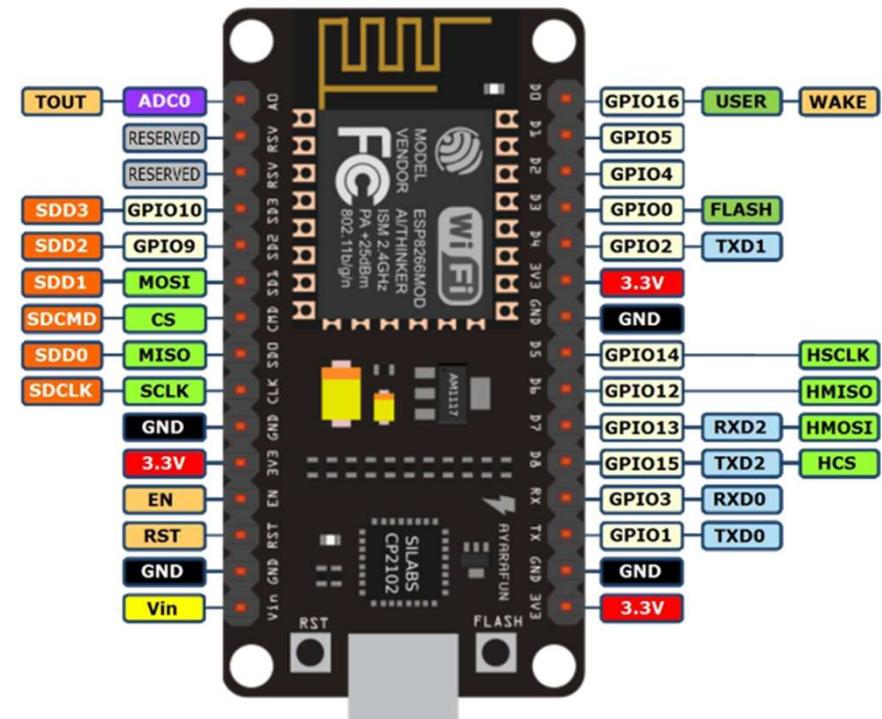
- L'ESP8266 peut se programmer de plusieurs façons :
- Avec des scripts Lua, interprétés ou compilés, avec le firmware NodeMCU
- En C, avec l'IDE Arduino
- En JavaScript, avec le firmware Espruino
- En MicroPython, avec le firmware MicroPython 2
- En C, avec le SDK d'Espressif
- En C, avec le SDK esp-open-sdk3 basé sur la chaîne de compilation GCC

## 7- Exemples d'application

# Application 1 : commande d'une LED

---

- On utilisera un module NodeMCU (utilisant un ESP8266).
- But : pouvoir allumer et éteindre une LED (indicateur lumineux).
- On utilisera l'environnement de développement ARDUINO.
- On reliera la LED et la résistance de limitation de courant entre la masse (GND) et la broche GPIO5 (D1) par exemple.



# Application 1 : commande d'une LED

Fichier Édition Croquis Outils Aide

led\_clignotte

```
void setup(void)
{
    // Set GPIO 5 as output
    pinMode(5, OUTPUT);
    // Set GPIO 5 on a HIGH state

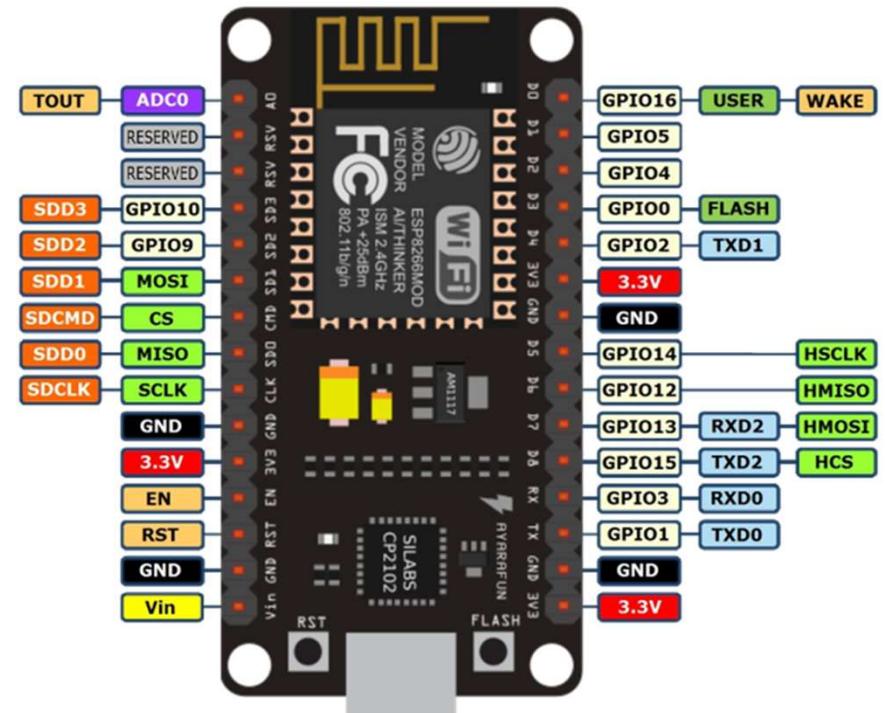
}

void loop() {
    digitalWrite(5, HIGH);
    delay(1000);
    digitalWrite(5, LOW);
    delay(1000);
}
```

Compilation terminée.

Le croquis utilise 222 205 octets (21%) de l'espace de stockage.  
Les variables globales utilisent 31 572 octets (38%) de l'espace de stockage.

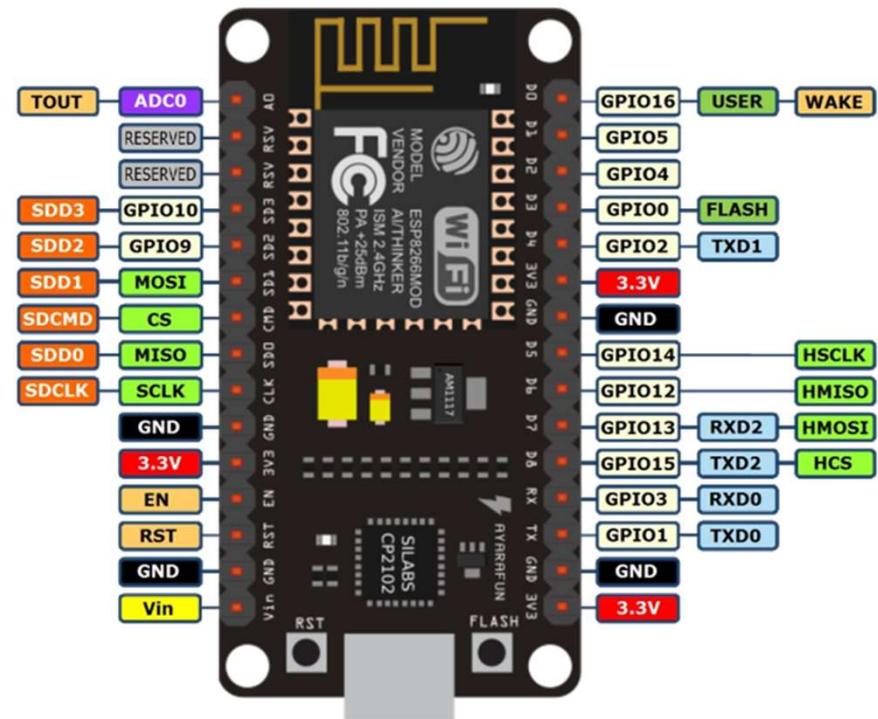
15 NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (3M SPIFFS) sur COM8



# Application 2 : lecture d'une entrée numérique

- On utilisera un module NodeMCU (utilisant un ESP8266).
  - But : pouvoir lire l'état d'une entrée numérique.
  - On utilisera l'environnement de développement ARDUINO.
  - On reliera l'entrée numérique sur la broche GPIO14 (D5).

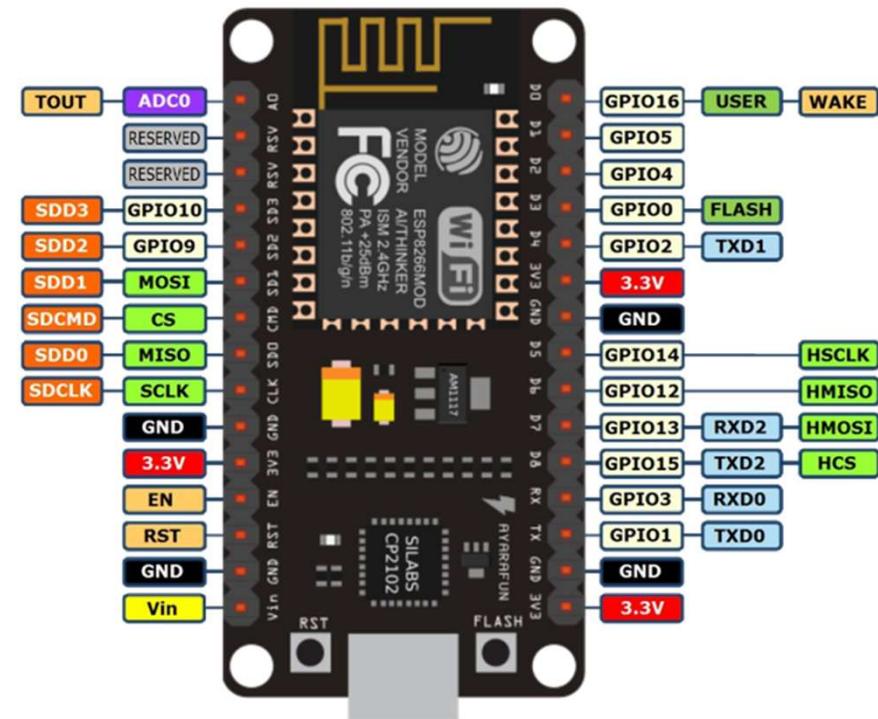
```
void setup(void)
{
// Start Serial
Serial.begin(115200);
// Set D5 as input
pinMode(14, INPUT);
}
void loop() {
// Read D5 and print it on Serial port
Serial.print("State of D5: ");
Serial.println(digitalRead(14));
// Wait 1 second
delay(1000);
}
```



# Application 3 : Scan des réseaux wifi disponibles et affichage à l'aide de la liaison série

---

- On utilisera un module NodeMCU (utilisant un ESP8266).
- But : Déetecter les réseaux wifi disponibles et afficher le SSID, la puissance du signal et indiquer si le réseau est ouvert ou crypté.



# Application 3 : Scan des réseaux wifi disponibles et affichage à l'aide de la liaison série

```
#include "ESP8266WiFi.h"

void setup() {
    Serial.begin(115200);

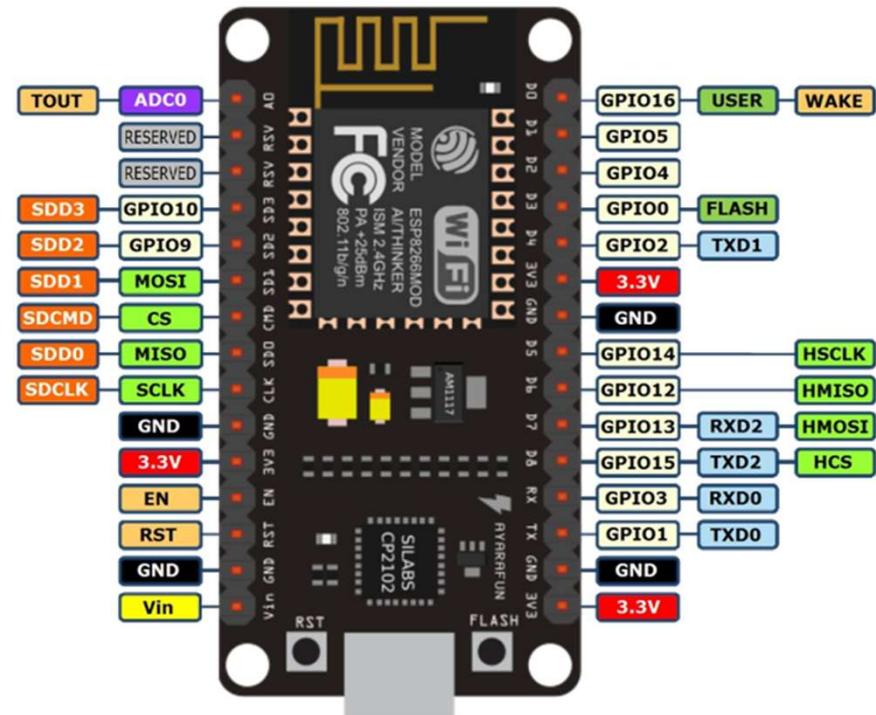
    // Set WiFi to station mode and disconnect from an AP if it was previously connected
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("Setup done");
}

void loop() {
    Serial.println("scan start");

    // WiFi.scanNetworks will return the number of networks found
    int n = WiFi.scanNetworks();
    Serial.println("scan done");
    if (n == 0)
        Serial.println("no networks found");
    else
    {
        Serial.print(n);
        Serial.println(" networks found");
        for (int i = 0; i < n; ++i)
        {
            // Print SSID and RSSI for each network found
            Serial.print(i + 1);
            Serial.print(": ");
            Serial.print(WiFi.SSID(i));
            Serial.print(" (");
            Serial.print(WiFi.RSSI(i));
            Serial.print(")");
            Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE)? " ":"*");
            delay(10);
        }
    }
    Serial.println("");

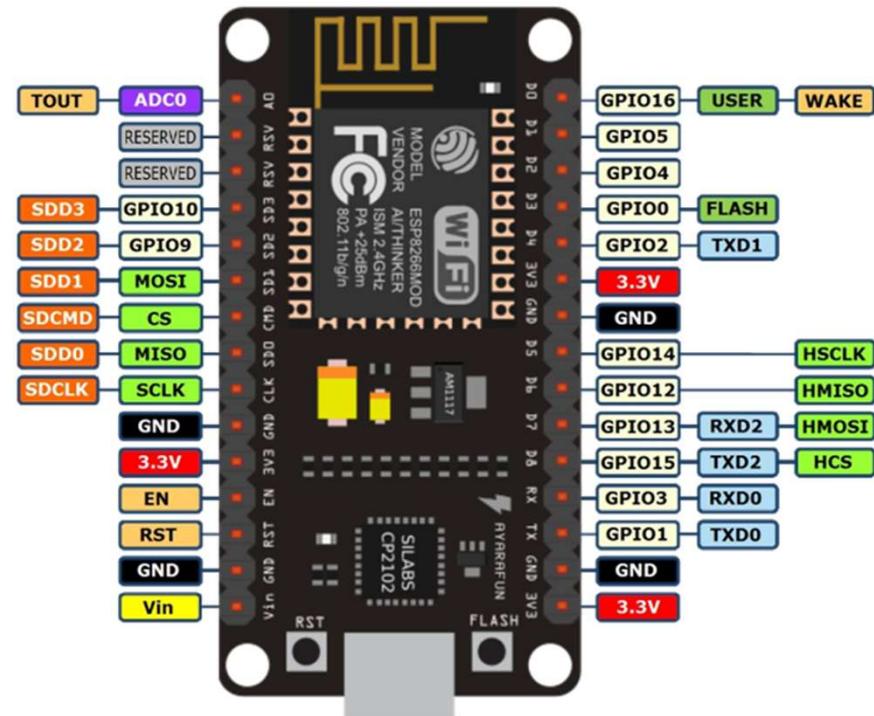
    // Wait a bit before scanning again
    delay(5000);
}
```



# Application 3 : Scan des réseaux wifi disponibles et affichage à l'aide de la liaison série

---

```
scan start
scan done
16 networks found
1: SFR-0000 (-69)*
2: Freebox-5XXECD (-77)*
3: FreeWifi (-80)
4: FreeWifi_secure (-79)*
5: SFR-f04332 (-74)*
6: Livebox-57A46 (-93)*
7: orange_banane (-93)
8: NETwork2 (-47)*
9: FreeWifi (-47)
10: FreeWifi_secure (-46)*
11: FreeWifi (-86)
12: FreeWifi_secure (-87)*
13: freebox_AZZOF (-85)*
14: freefree (-89)*
15: FreeWifi (-88)
16: SFR WiFi FON (-91)
```



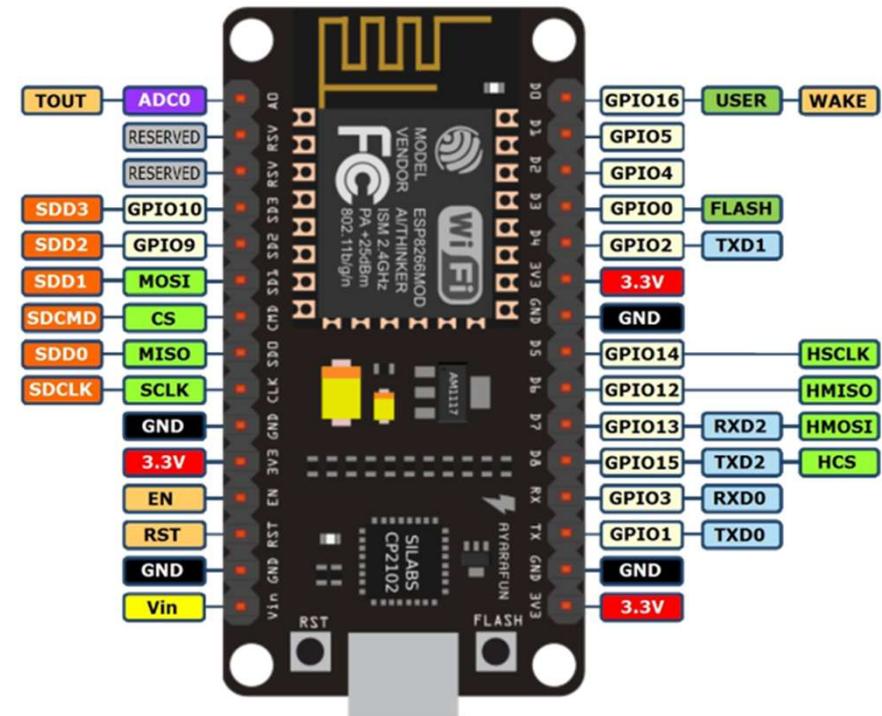
## Application 4 : connexion à un réseau WiFi (connaissant le SSID et le mot de passe) et affichage de l'adresse IP obtenue par DHCP

```
#include <ESP8266WiFi.h>

// WiFi parameters
const char* ssid = "mon_SSID"; // à remplacer par le SSID
const char* password = "mon_mot_de_passe"; // mettre ici le mot de passe

void setup(void)
{
// Start Serial
Serial.begin(115200);
// Set WiFi to station mode and disconnect from an AP if it was previously
connected
WiFi.mode(WIFI_STA);
WiFi.disconnect();
delay(100);
// Connect to WiFi
WiFi.begin(ssid, password);
Serial.println("");
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
// Print the IP address
Serial.println(WiFi.localIP());
}

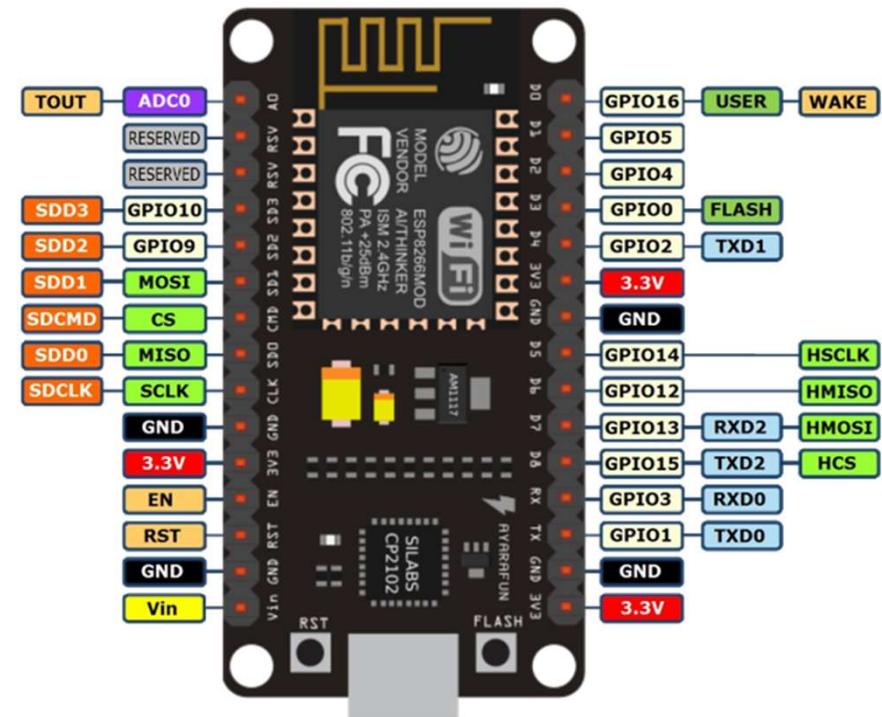
void loop() {
```



Application 4 : connexion à un réseau WiFi (connaissant le SSID et le mot de passe) et affichage de l'adresse IP obtenue par DHCP

---

.....  
WiFi connected  
192.168.0.6

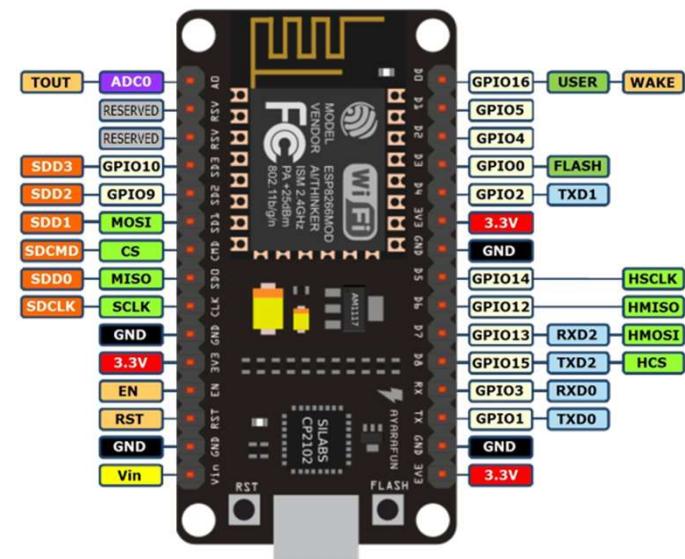


# Application 5 : AP WiFi, serveur web, commande à distance d'une LED et html/css

---

Caractéristiques de l'application :

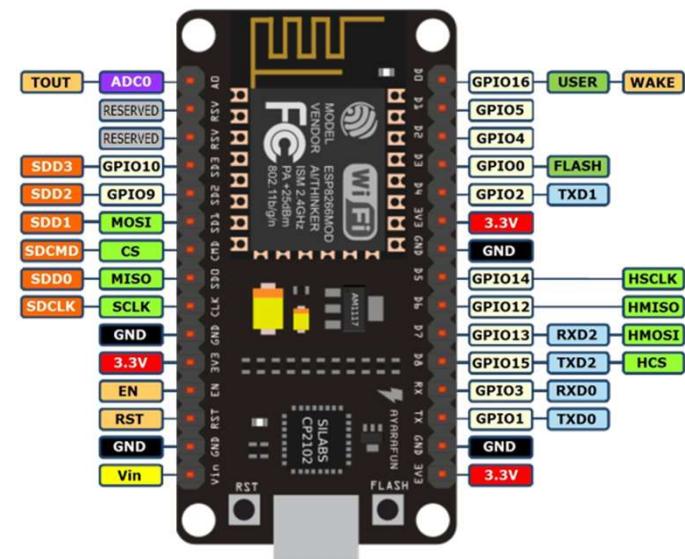
- point d'accès WiFi (AP) ;
- serveur WEB ;
- commande à distance d'une LED ou de tout autre dispositif ;
- composition de pages en html et css.



# Application 5 : AP WiFi, serveur web, commande à distance d'une LED et html/css

---

Voir le fichier complet !  
(trop long pour être affiché sur une page de présentation)



# 8- Bibliographie

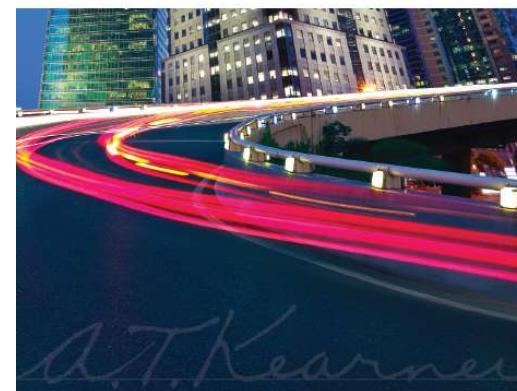
# Ressources documentaires

## Etudes des institutions nationales et européennes (stratégies)



### The Internet of Things: A New Path to European Prosperity

With the right focus, Europe can capitalize on the rise of the Internet of Things and the €1 trillion opportunity that it brings.



# Ressources documentaires

---

Ouvrages scientifiques et techniques (architectures, structuration, organisation)



# Ressources documentaires

---

## Aspects technologiques

- Magazines techniques (Hackable, MISC, OpenSilicium, Embarqué, The MagPi, etc.)
- Ouvrages des éditions techniques (nombreux ouvrages sur les plateformes technologiques embarquées)

Merci pour votre attention

---

# Compléments

# Principe général

---

**Objet communiquant** : système capable de :

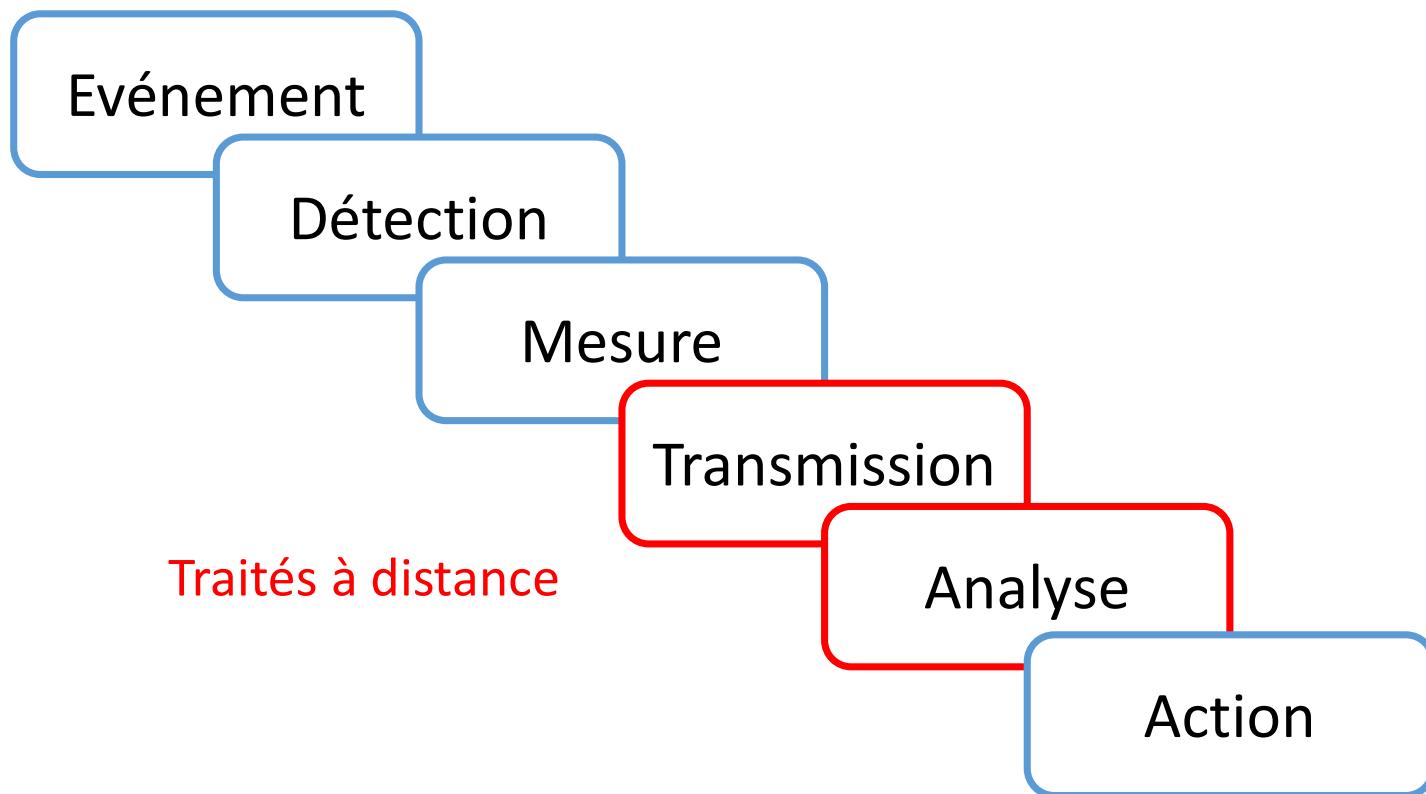
- réaliser des mesures de grandeurs physiques ;
- traiter (ou pré-traiter) ces données et de les mettre en forme ;
- transmettre ces informations ;
- recevoir des données ;
- agir (influer sur l'environnement).

... Et cela en respectant des contraintes fortes (faible coût, faible encombrement, faible consommation énergétique, robustesse, etc.)

# Principe général

---

Fonctions de base de l'IoT

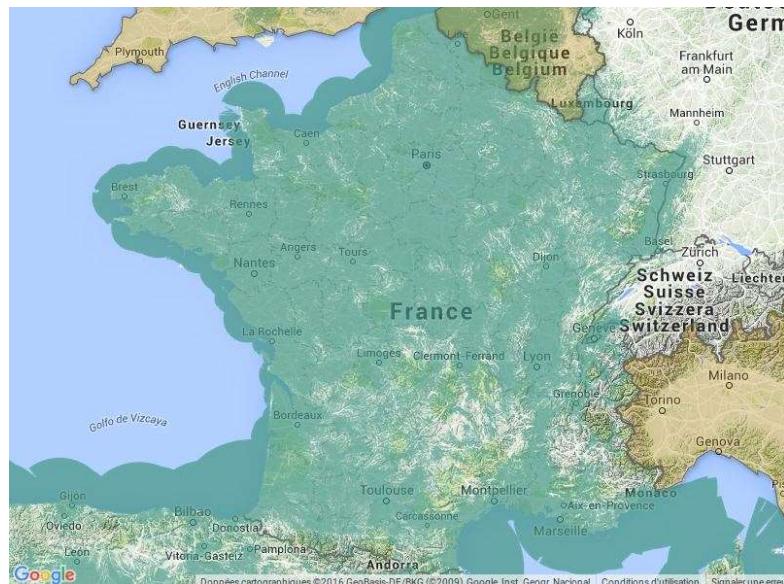


Transmissions à longue distance  
et à très faible débit

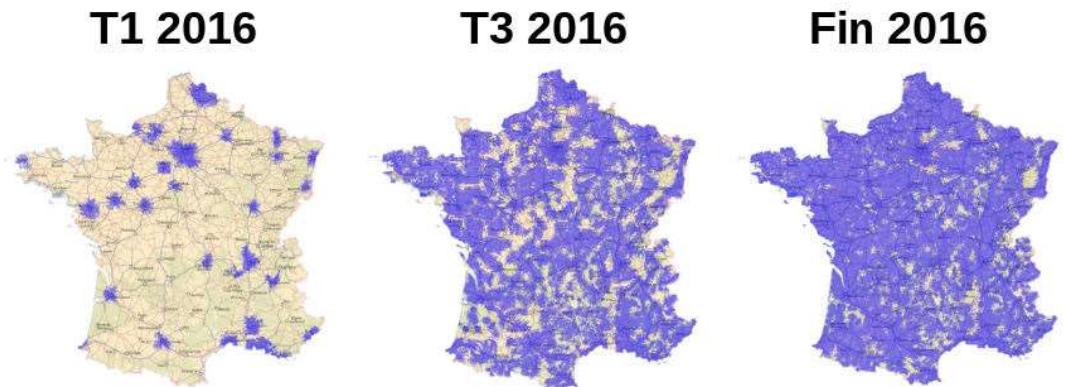
# Transmission à longue distance et à très faible débit

---

Couverture du réseau SigFox en 2016.



Couverture du réseau LoRa de Bouygues télécom (objenious) en 2016.



Sigfox promet une meilleure pénétration et une meilleure portée, LoRa annonce une meilleure communication bidirectionnelle et une localisation par triangulation plus fine.

# Transmission à longue distance et à très faible débit

---

## Applications potentielles

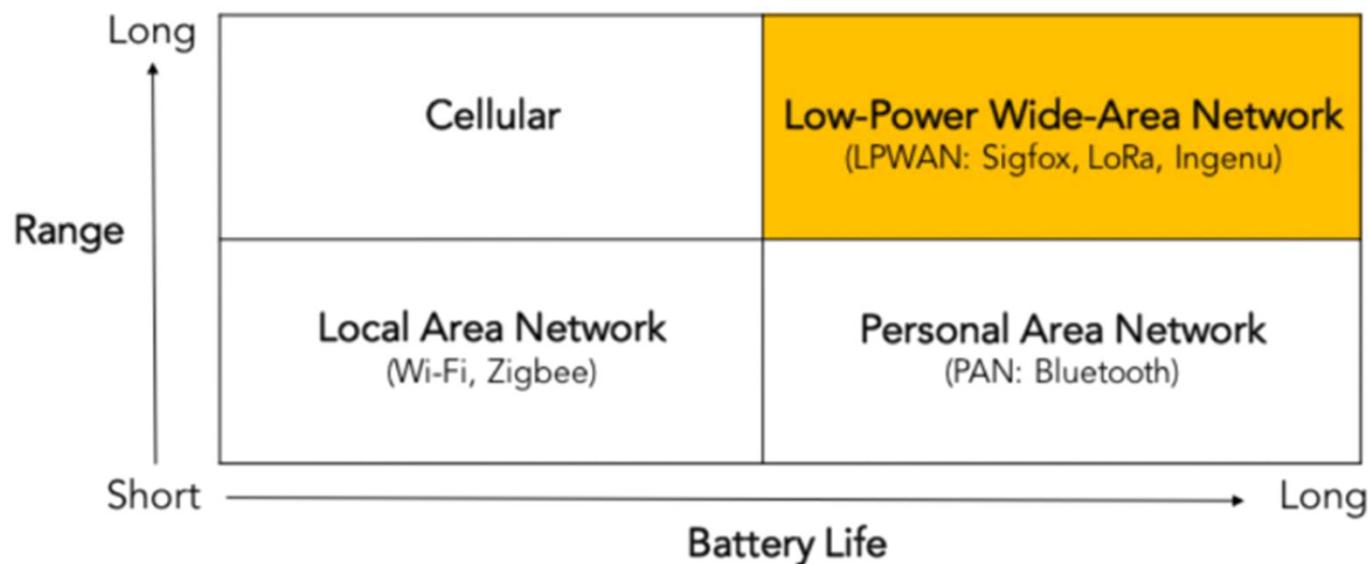
- Suivi de données biologiques chez les patients (transmission de paramètres à une structure médicale pour un suivi massif à distance).
- Ville intelligente (signalisation routière, gestion des parkings, surveillance, etc.).
- Etude de phénomènes naturels.

## Intérêt majeur

- Une pile bouton peut offrir plus d'un an d'autonomie et une bonne portée (jusqu'à 20 km en champ libre, environ 1 km en zone urbaine). 4 000 antennes sont suffisantes pour assurer une couverture comparable en surface à celle obtenue avec plus de 15 000 antennes du réseau 3G !

# Transmission à longue distance et à très faible débit

---



RFID (Radio Frequency Identification)  
& NFC (Near Field Communication)

# RFID & NFC

---



# Wearable technology

# Wearable

---



# Wearable

---

“System” Sensors

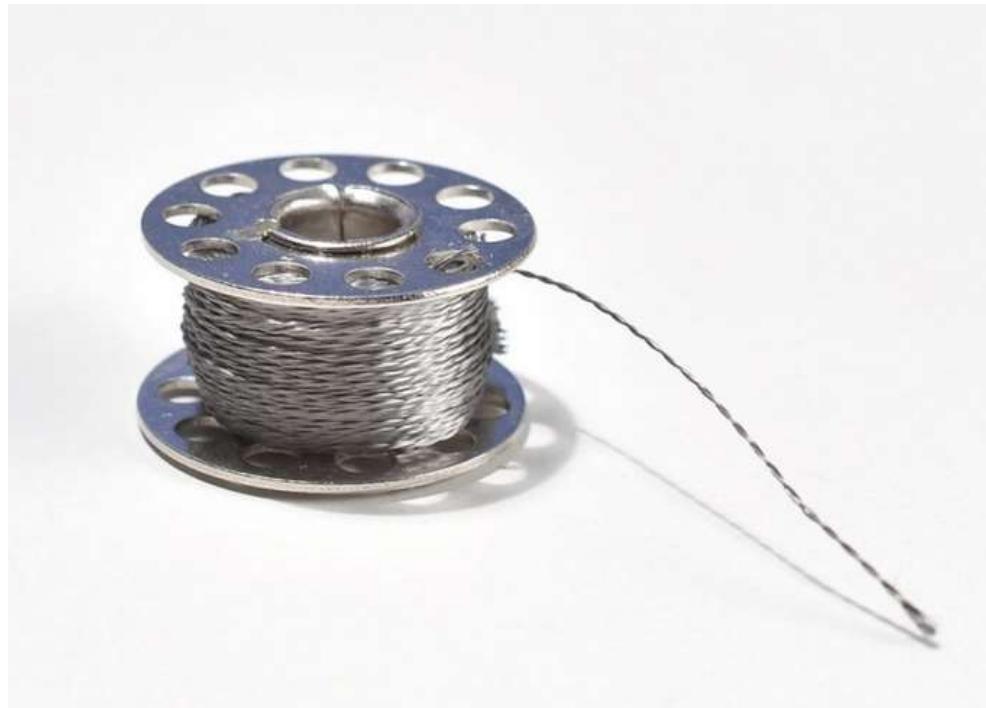


Human Sensors



# Wearable

---



Fil conducteur inoxydable pour couture électronique e-textile



Velostat : feuille conductrice sensible à la pression

# Wearable

---

- Exemple : société SPECKTR (Montpellier), [www.specktr.com](http://www.specktr.com)
- Gant connecté pour applications musicales ou autres.

Centrale inertielle,  
traitement des données,  
liaison BlueTooth.



Fils conducteurs électriques

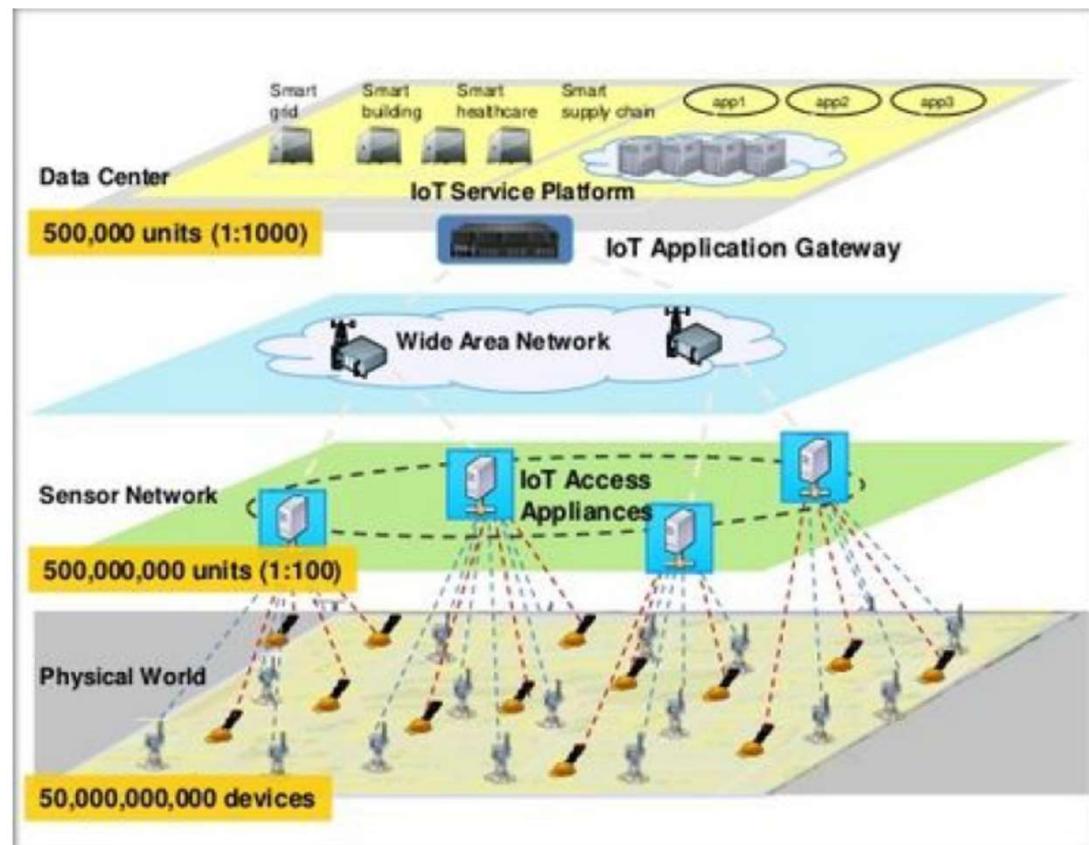


# Architectures de l'IoT

# Architecture de l'IoT

---

- IPv6
- Protocoles adaptés (MQTT : (Message Queuing Telemetry Transport) fondé sur TCP/IP
- Sécurité
- Liens avec les bases de données (MySQL, etc.)
- Programmation
- Etc.



# Architecture de l'IoT

---

- Problématique ouverte
- Pas (encore) de standardisation
- Compétition forte entre les acteurs majeurs de l'embarqué

# R&T et IoT

- Exemple d'application complète exploitant plusieurs disciplines

Autentification

|    | ref      | variable | valeur  |    |             |         |
|----|----------|----------|---------|----|-------------|---------|
| 1  | Modifier | Copier   | Effacer | 1  | temperature | 25      |
| 2  | Modifier | Copier   | Effacer | 2  | temperature | 12.23   |
| 3  | Modifier | Copier   | Effacer | 3  | V1          | 123.456 |
| 4  | Modifier | Copier   | Effacer | 4  | V1          | 123.456 |
| 5  | Modifier | Copier   | Effacer | 5  | V1          | 123.456 |
| 6  | Modifier | Copier   | Effacer | 6  | V1          | 123.456 |
| 7  | Modifier | Copier   | Effacer | 7  | V1          | 123.456 |
| 8  | Modifier | Copier   | Effacer | 8  | V2          | 1.2     |
| 9  | Modifier | Copier   | Effacer | 9  | V2          | 1.3     |
| 10 | Modifier | Copier   | Effacer | 10 | V2          | 1.4     |
| 11 | Modifier | Copier   | Effacer | 11 | ESP8266     | 123.456 |
| 12 | Modifier | Copier   | Effacer | 12 | ESP8266     | 123.456 |
| 13 | Modifier | Copier   | Effacer | 13 | ESP8266     | 123.456 |
| 14 | Modifier | Copier   | Effacer | 14 | ESP8266     | 123.456 |
| 15 | Modifier | Copier   | Effacer | 15 | ESP8266     | 123.456 |
| 16 | Modifier | Copier   | Effacer | 16 | ESP8266     | 123.456 |
| 17 | Modifier | Copier   | Effacer | 17 | ESP8266     | 123.456 |
| 18 | Modifier | Copier   | Effacer | 18 | ESP8266     | 123.456 |
| 19 | Modifier | Copier   | Effacer | 19 | ESP8266     | 123.456 |

Serveur WEB

Interpréteur  
PHP

Internet



Interfaçage



WIFI

Configuration réseau

Sécurité

MySQL

Requêtes SQL

Programmation en C

99

```
#include <ESP8266WiFi.h>
// Spécifier le SSID du point d'accès Wifi
const char* ssid = "your-ssid";
// Spécifiez le mot de passe associé à votre point d'accès
const char* password = "your-password";
// Crée le serveur Web en spécifiant le port TCP/IP
WiFiServer server(80);
// Démarrage
void setup() {
    // Initialisation série 115200
    Serial.begin(115200);
    // WiFi: à la main
    delay(10);
    // WiFi en mode
    WiFi.mode(WIFI_STA);
    // WiFi: à la main
    delay(10);
    // Deux sauts de ligne pour faire le séparateur car
    // le modèle de démarrage envoie des caractères sur le port série
    Serial.println();
    Serial.println("Connexion à : ");
    Serial.print(ssid);
    Serial.println("");
    // Connexion au point d'accès
    WiFi.begin(ssid, password);
    // On boucle en attendant une connexion
    // Si l'état est WL_CONNECTED la connexion est acceptée
    while (WiFi.status() != WL_CONNECTED) {
        // On affiche notre adresse IP
        Serial.print("IP: ");
        Serial.println(WiFi.localIP());
    }
    // Boucle principale
    void loop() {
        // Si un client Web est connecté ?
        WiFiClient client = server.available();
        if (client) {
            // On abandonne ici et on repart dans un tour de loop
            return;
        }
    }
}
```

# Merci pour votre attention !



Merci de votre attention