



# Internet of Things



# *Chapitre I : Concepts et Terminologies IOT*



## Concepts de base de l'IOT

- ❖ **Objets physiques:** des capteurs, technologie de connectivité et intelligence.
- ❖ **Réseaux de communications :** utilisés pour transporter des données d'objets.
- ❖ **Cloud computing:** fournit les outils de stockage, de corrélation et d'analyse des données (processus décisionnels capables de remonter les objets physiques).



# Definition des objets physiques et virtuels

**Objet physique :** Equipment électronique – Equipment  
électronique- des robots- un réfrigérateur-une machine à laver.

**Objets virtuel:** un contenu multimedia ou un logiciel.



## Concepts de base de l'IOT

**Système embarqué :** C'est un système informatique et électronique autonome effectuant une tâche précise. Il communique donc avec d'autres systèmes.

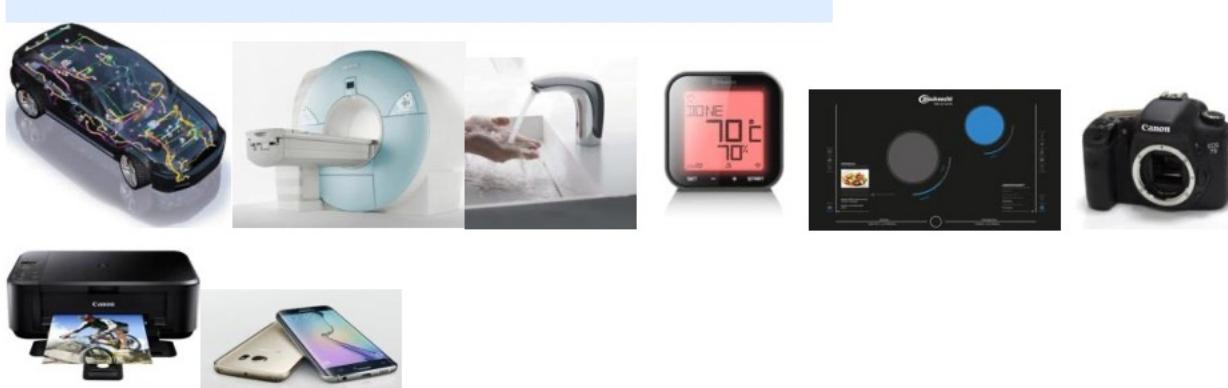
**Exemple d'embedded systems :**

- Panneau d'affichage urbain
- Pompe à essence
- Télévision
- Caméra digitale d'une voiture

## Concepts de base de l'IOT

**Systèmes embarqués n'est pas forcément**

- ✓ Mobile
- ✓ Petit
- ✓ cache
- ✓ Puissant
- ✓ Complexe
- ✓ Alimenté par batterie
- ✓ Economie en énergie
- ✓ cher



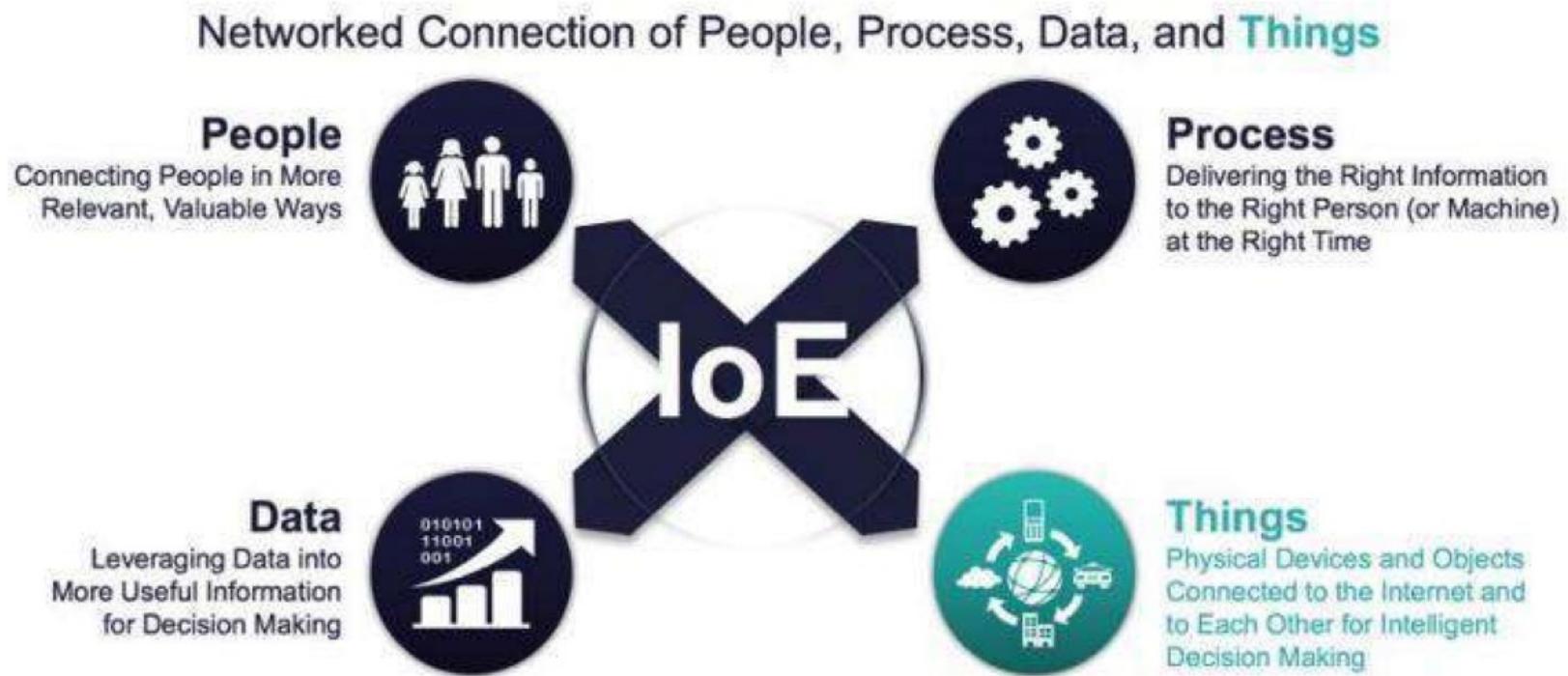
## Concepts de base de l'IOT

### Un Systemes embarqué:

- Tout système conçu pour résoudre un problème dédié
- Système électronique et informatique autonome possédant des entrées-sorties spécifiques



# Internet of Every Things





# M2M

**M2M** = machine to machine = communication entre machines

- ✓ contact (RFID, NFC)
- ✓ QR Code

## **Exemple :**

- système de freinage anti-bloquant
- régulateur de vitesse des automobiles



# Internet des objets

L'IoT, ou **Internet des objets**:

Le processus de connexion d'**objets** physiques à **Internet, des objets** du quotidien

- ❖ Ampoules
- ❖ Dispositifs médicaux
- ❖ Appareils portables

# Internet des objets



# Objets connectés

**Telegarden** : juin 1995 (Univ. Of California)



Agriculture – art - internet

# Objets connectés

Nabaztag (Karotz), lancé par la société Violet en 2005

Ce lapin connecté en Wi-Fi peut :

- ✓ Lire des mails à haute voix
- ✓ Emettre des signaux visuels
- ✓ Diffuser de la musique.





# L'objectif de l'IoT

- ❖ Assistance à nos activités professionnelles et personnelles
- ❖ Permet une réduction considérable des dépenses dans l'économie d'aujourd'hui (industrie, santé, sécurité, etc.).



## Definitions IOT

**Definition 1:** objets soient identifiables d'une façon unique, représentation virtuelle et une connexion internet(**Wikipedia**)

**Definition 2:** se sont des objets physiques accessibles par Internet qui combinent plusieurs technologies, les systèmes embarqués (**CISCO**)



# Concepts de l'IOT

**Les protocoles de communication :** assurer la connectivité entre objets et applications.

Il définissent le format des données, taille paquets, adressage, routage, etc.

**Les réseaux de capteurs sans fil RCSF:** (Wireless Sensor Network, WSN) se compose d'un nombre de Noeuds-Capteurs qui ont des fonctionnalités de capturer et traiter/transmettre les données.



# Concepts de l'IOT

**Cloud Computing** : fournit un espace de stockage de données IOT et offre des services de visualisation, analyse et archivage des données.

**Big Data** : offre des outils d'analyse avancées pour les données massives collectées par les objets IOT selon leurs caractéristiques.



## Definitions de IOT

**Definition3:** Les objets du monde physique qui sont transposes dans le monde virtuel Information qui circule dans le reseau pour avoir un type de service de valeur ajoutee.

Ces objets sont intelligents, sont capables d'echanger une information, communiquer via le reseau , une interoperabilite  
**(Union internationale des Telecommunication)**



# Types de communications

- **Things to Things**( montre intelligente avec mon smartphone)
- **Human to Human** en utilisant un PC ou un autre équipement.
- **Human to Thing**( humain avec son smartphone)
- **Anytime communication** ca peut se faire n'importe quel moment
- **Any Place communication** Ca peut se faire dans n'importe quel emplacement.

# Technologies proches de IOT

## P2P: Peer to Peer

Deux Transfert machines ou deux PCs

## **M2P: Machine to personne**

**Exemple :** les personnes avec les smartphones

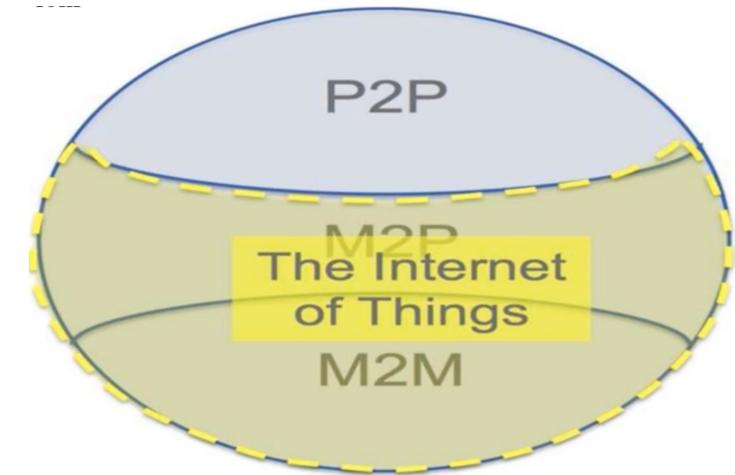
## **Les assistants personnels**

## M2M:

Machine to Machine (pas un reseau) internet.

### **Exemple :**

-Cle de voiture qui va l'ouvrir automatiquement.





# Caracteristiques d'un objet connecte

- ❖ **Interconnectivite:** L'objet doit etre connecte
- ❖ **Heterogenite:** des objets differents connectes via bluthooth, via Wifi.
- ❖ **Changements dynamiques:** l'état des objets va changer d'une façon continue, le nombre d'objets aussi est dynamique (position, la vitesse)



## Caractéristiques d'un objet connecté

- 1-connecté en réseau Wifi
- 2-Communiquer des messages et réagir à des événements.
- 3-Capacité de comprendre l'événement et de réagir selon cet événement.

## Exemple: Réfrigérateur Objet Connecté



**Maintient les aliments froides**

## Exemple: Réfrigérateur Intelligent

Quand la porte du réfrigérateur est ouverte pour une période assez grande.

- Alarme déclenche pour fermer la porte.
- Il déclare que l'eau du filtre doit être changée.
- un capteur ayant une caméra qui identifie si le beurre est périmé.
- propose des recettes en se basant sur les courses existantes dans le réfrigérateur le beurre, le lait .....





## Exercice

Décrire l'évolution d'une voiture traditionnelle vers une voiture IoT et quelles sont les fonctionnalités offertes par cette dernière?

Décrire l'évolution d'une caméra normale vers une caméra IoT et quelles sont les fonctionnalités offertes par cette dernière?



# Internet of Things: Objet connecté

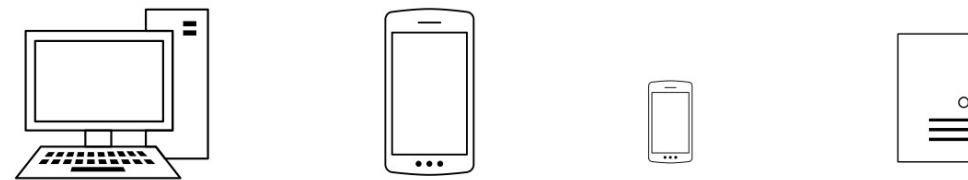
L'IoT (Internet of Things ou Internet des Objets en français) représente la prochaine évolution de l'Internet.

un objet qui a une fonction mécanique et/ou électrique propre  
conçu directement connectable

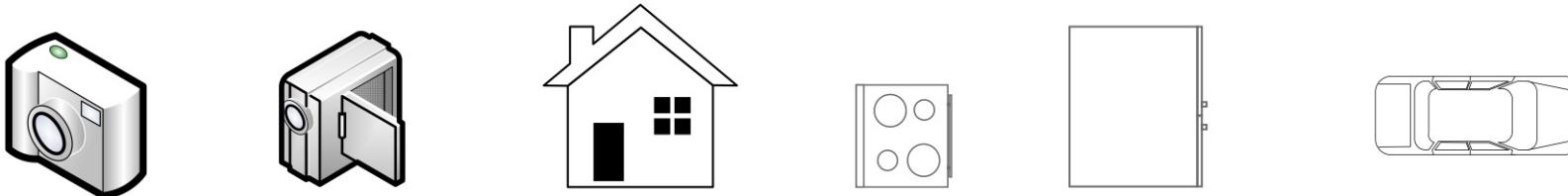
→ collecter des données des capteurs, traiter ces données et les communiquer à l'aide de d'une fonction de connectivité et de recevoir des instructions pour exécuter une action.

# Concepts fondamentaux

Quelques objets connectés traditionnels (Ordinateurs-smartphone)



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





# Evolution Internet of Things

**Exemple objet connecté:** les montres qui calculent les distances parcourues et partager ces informations dans des réseaux sociaux.

**Objet connecté:** rendre des services dans le domaine du transport par exemple Jawaz pour compter les nombres des passagers qui ont traversé les stations du payages grâce à la technologie RFID.



## Types d'objets connectés

**Objets dédiés au grand public (wearables)** : montre, bracelet, vêtement, etc.

**Objets connectés dans le B2B** sont source de nouveaux business à usage industriel



# Evolution Internet of Things

**1999:** Kevin Ashton a inventé le terme Internet of Things qui désigne le lien entre RFID et l'internet.

**2003:** Création du premier objet connecté, la lampe DIAL commercialisée en 2003 par la firme Violet.

**2007:** l'apparition des smartphones.

**2008:** Création des adresses IPSO, adresses IP qui vont permettre aux objets connectés d'interagir entre eux.



# Evolution Internet of Things

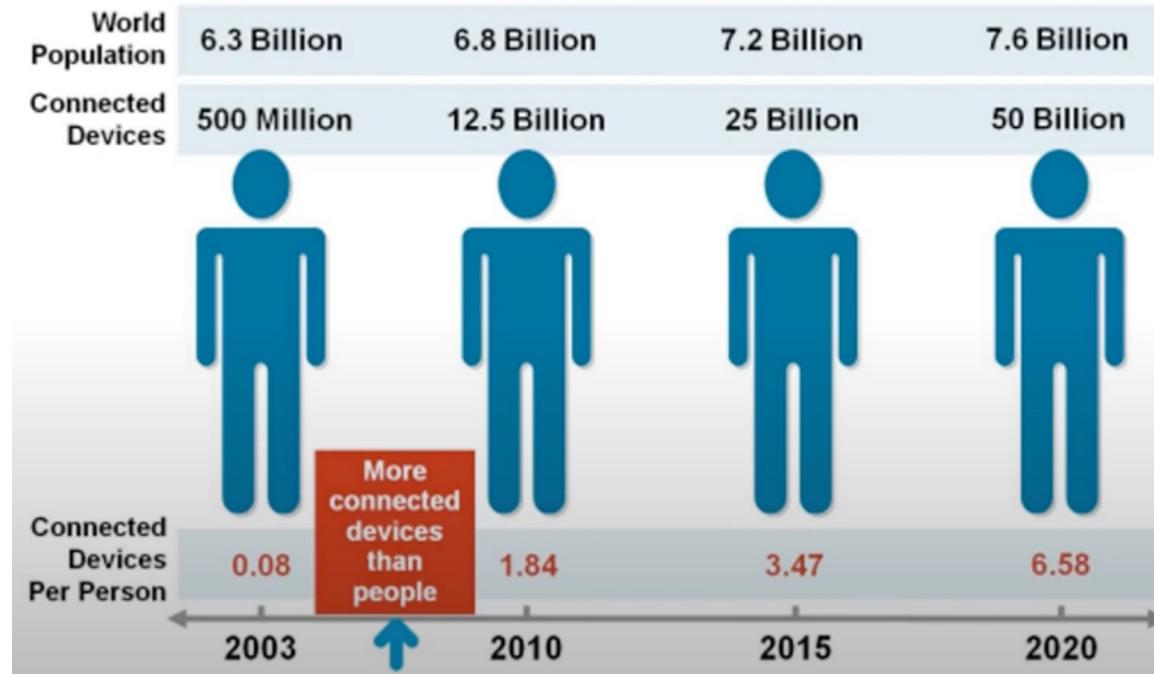
Pourquoi les adresses IPV4 ne sont adaptes pour etiqueter et pour connaitre les objets connectes?

## **Exemple: Objet connecté Violet**

- ✓ Lampe DIAL est connectee en reseau Wifi, elle possede 9 Leds
- ✓ Lampe intelligente qui peut s'allumer avec plusieurs couleurs en fonction de plusieurs elements (meteo- Bourse-Pollution-alertes Google)
- ✓ Communiquer des messages et reagir a des evenements.
- ✓ Capacité de comprendre l' événement et de réagir selon cet événement.

# Evolution des objets connectés

En 2020, il existe 6.5 Billions d'objets connectés dans le monde





## Caracteristiques d'un objet connecte

Depuis 2017, on est passe des personnes connectees a des objets connectes

Personne est connectée par : Smartphone- montre.....

Un simple verre de temps peut etre transforme en un objet connecte(temperature) → le verre est colorie ou envoie un message pour dire que l'infusion est prete.

Une chaise, un stylo, un refrigerateur.... → Objet IOT

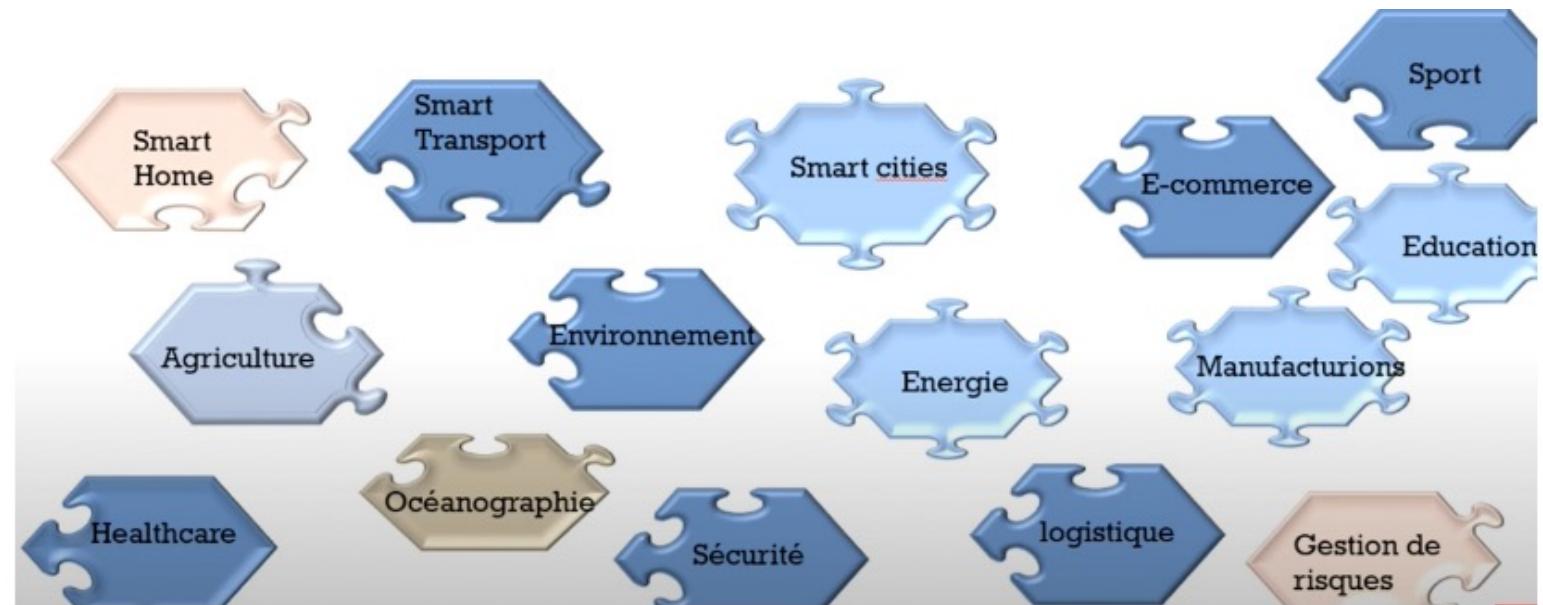


# *Chapitre II :Domaines d'applications, Avantages et Défis de l'IOT*

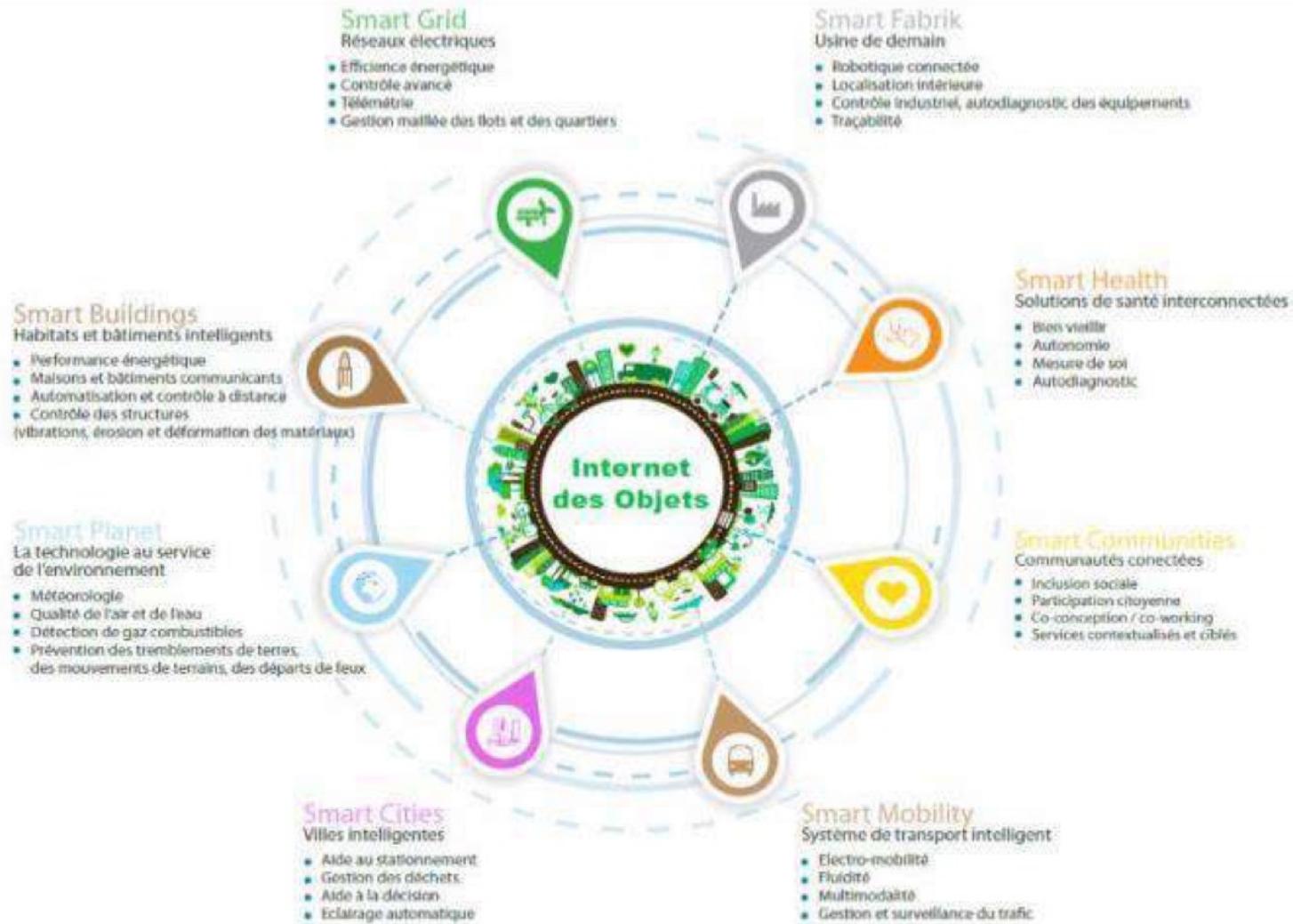
# Domaines d'application de IOT

Plusieurs domaines d'applications sont envisageables dans le domaine de IOT.

**Exemple :Climatisation**  
change  
dynamiquement en  
fonction de la  
temperature



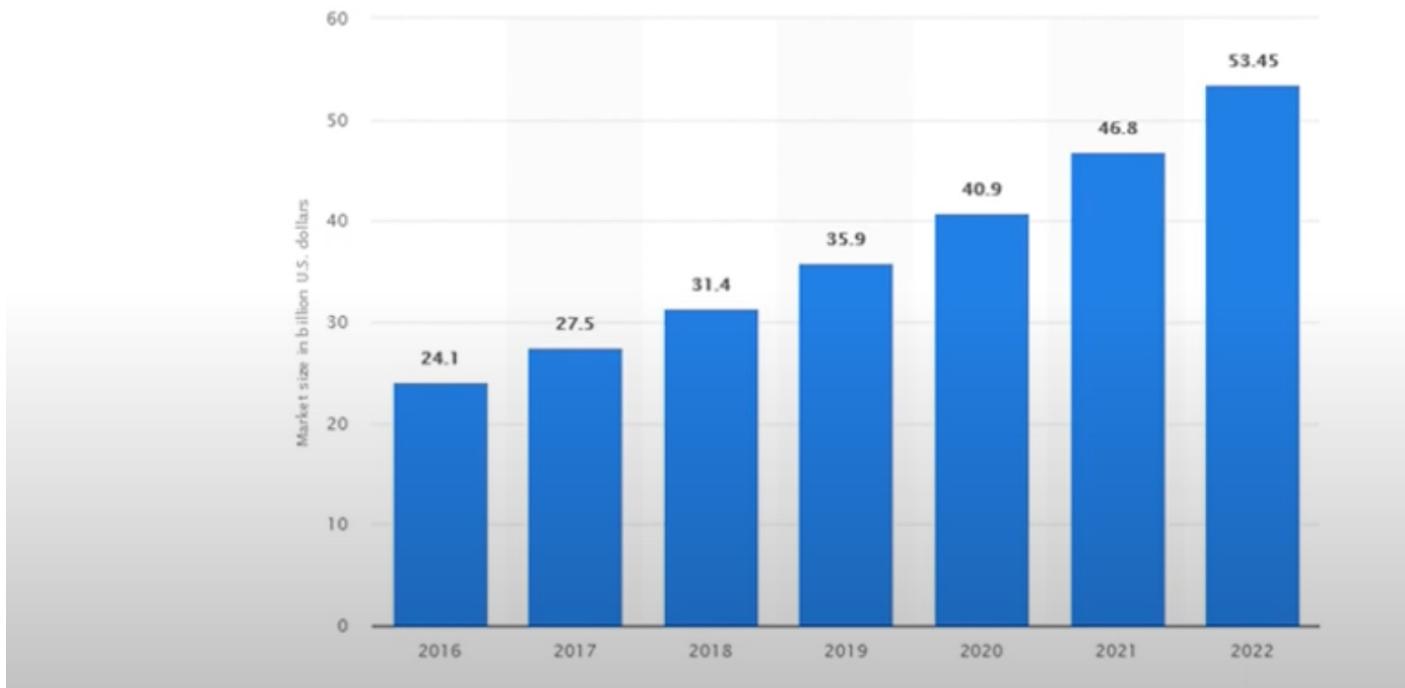
# Marche IOT



# SMART HOME

## Internet of Everything: cameras de surveillance

- Un marché en pleine expansion:



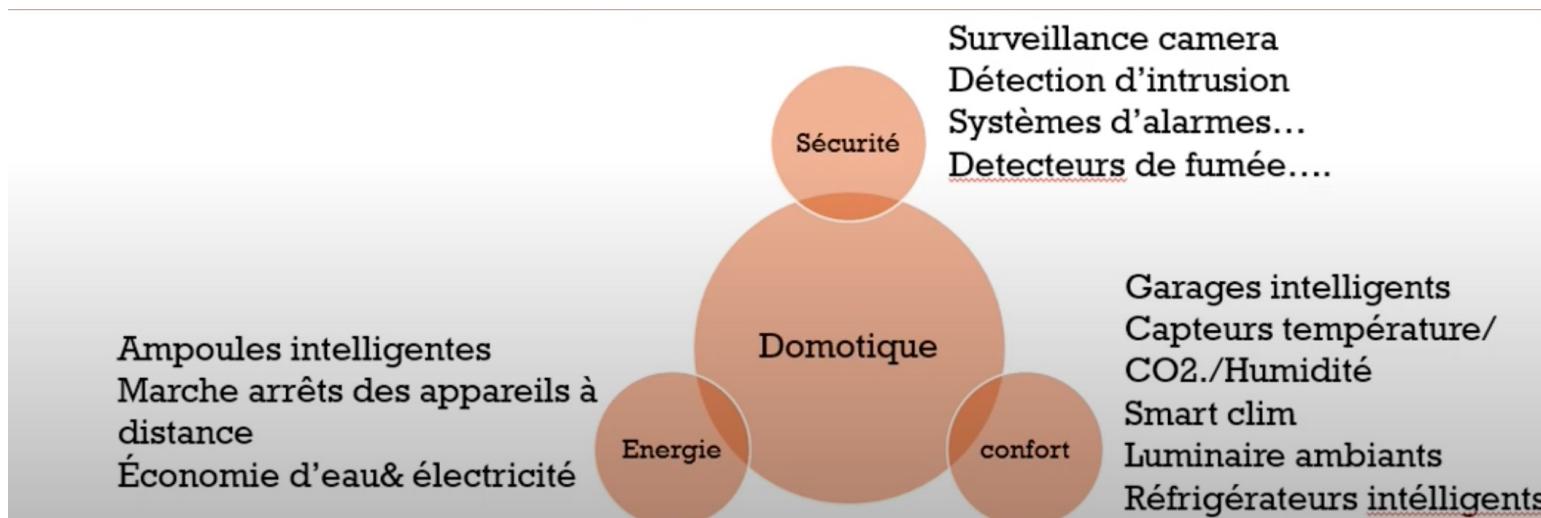


# SMART HOME

Décrire quelques objets dans une maison intelligente et leur fonctionnalités.

# SMART HOME

**Domotique:** technologie informatique+electronique  
+eletrotechnique → maison intelligente



# SMART HOME





# Smart Transportation

## sécurité

surveillance

sécurité garres/  
aéroports

Gestion  
parkings

## confort

payages

informations

circuits

Multi-modalité

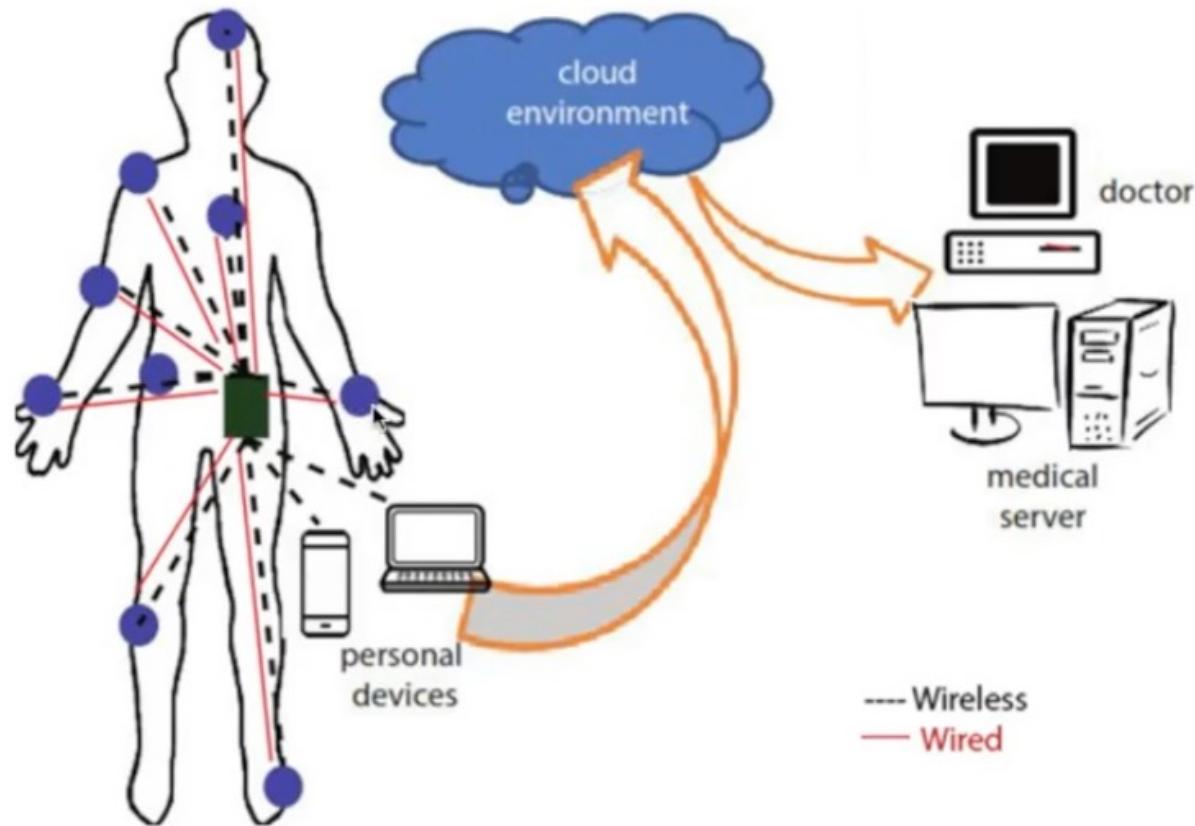
## trafic

Signalisation

optimisation

localisation

# Smart Healthcare



# Internet des objets dans le domaine de l'agriculture (Ferme Intelligente)





# Domaine d'application de l'IoT

**Ville intelligente** : circulation routière intelligente, transports intelligents, collecte des déchets, cartographies diverses (bruit, énergie, etc.)

**Environnements intelligents** : prédiction des séismes, détection d'incendies, qualité de l'air, etc.

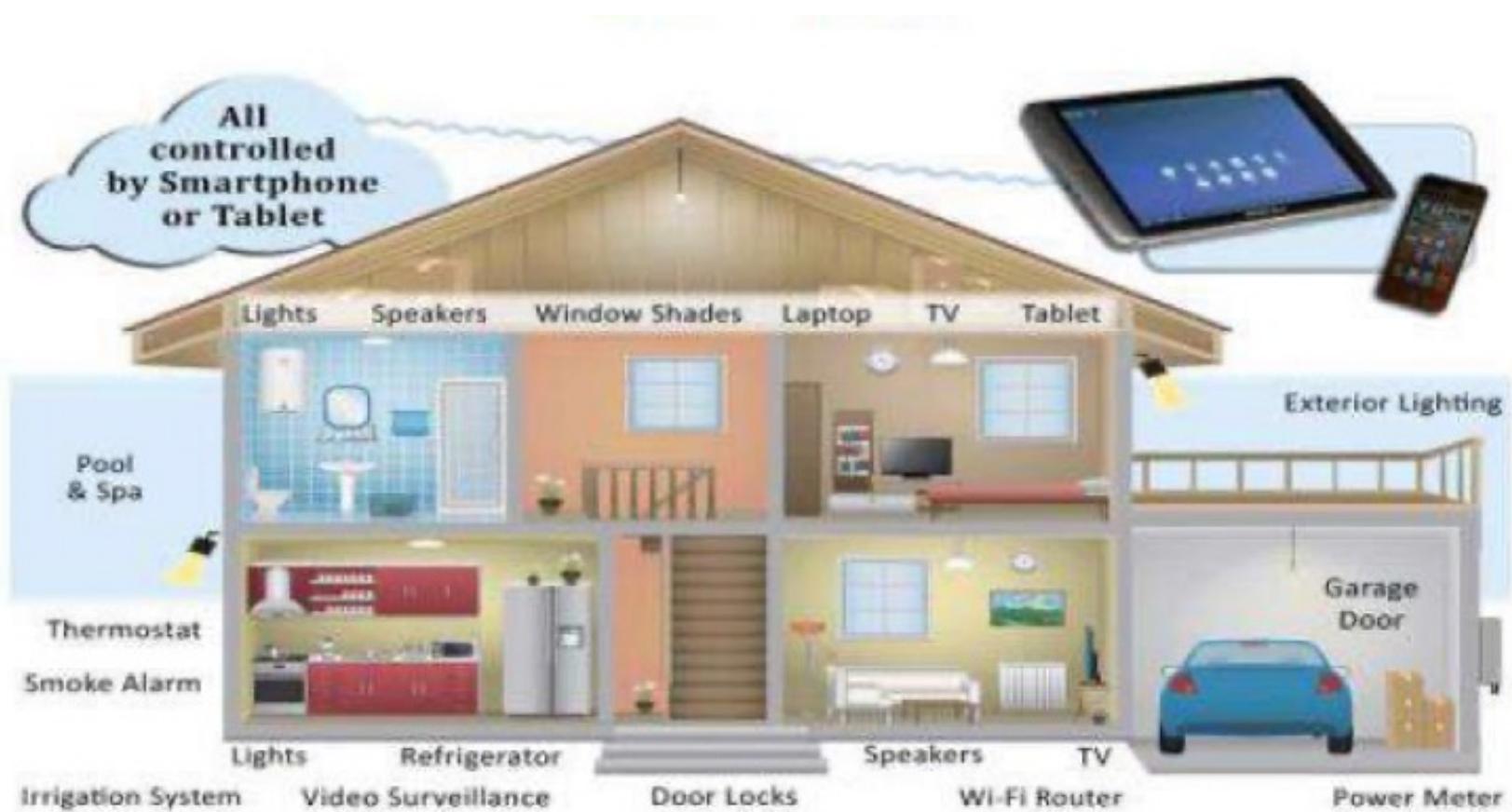
**Sécurité et gestion des urgences** : radiations, attentats, explosions.



# Domaine d'application de l'IoT

- **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

# Maison intelligente



# Smart home scenario

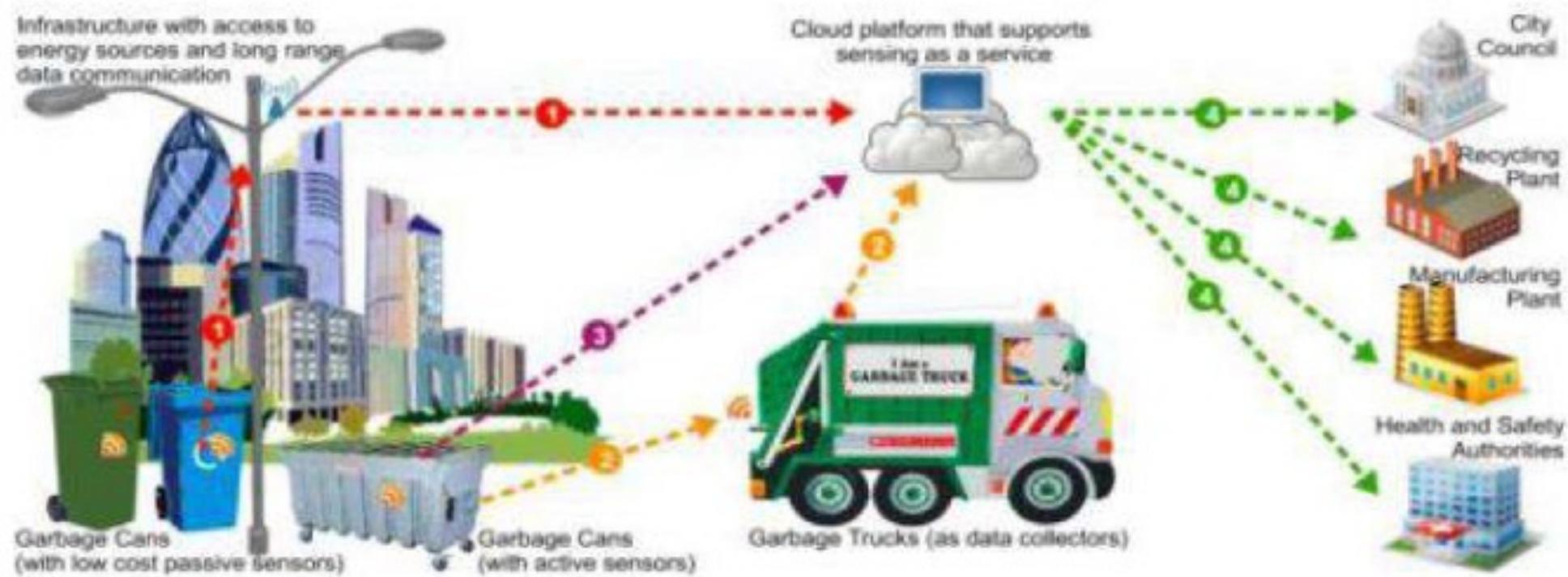




# Gestion des déchets

Décrire quelques scénarios représentant des objets connectés dans le cas de gestion des déchets

# Gestion des déchets dans les villes intelligentes





# Agriculture intelligente

Décrire quelques scénarios représentant des objets connectés dans le cas de l'agriculture intelligente.

# Agriculture intelligente



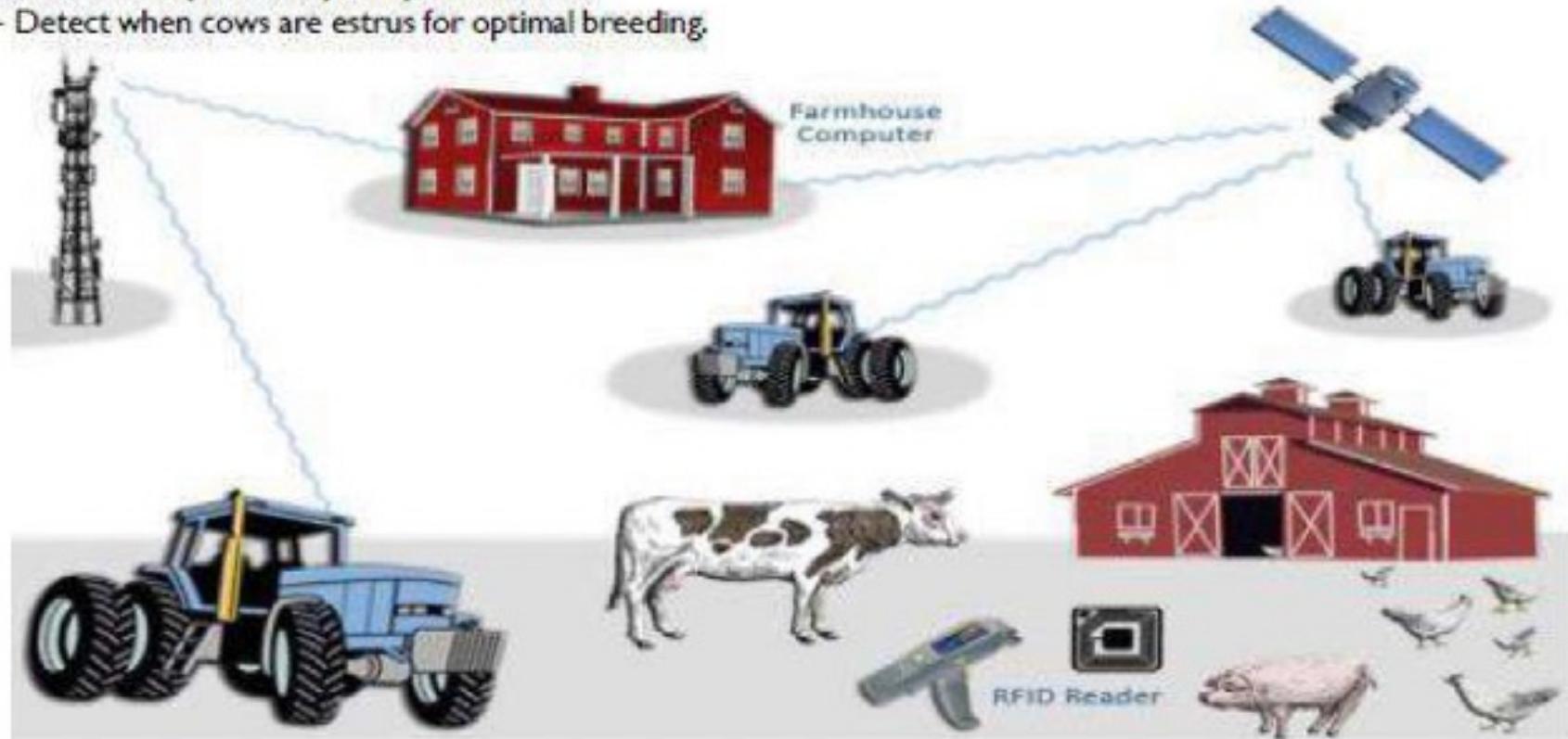


# Agriculture intelligente

Décrire quelques scénarios représentant des objets connectés dans le cas d'une ferme intelligente.

# Ferme intelligente

- Temperature sensors, moisture sensors, etc.
- Sensors to trap and analyze captured insects.
- Detect when cows are estrus for optimal breeding.





# Exercice

Décrire le scenario d'achat de courses dans un super  
marché en utilisant les caractéristiques d'un objet  
connecté.



# Smart City Applications: Smart Grid



Power Plant



Distribution Grid



Smart Meters



Smart Appliances



## Les avantages de L'Internet of Things

- ✓ Utilisation efficace des ressources
- ✓ Minimisation des efforts humains
- ✓ Gagner du temps
- ✓ Augmenter la sécurité

# Fonctionnalites de l'IOT

- ✓ Connecter plusieurs objets aux plateformes IOT
- ✓ Analyser les données
- ✓ Intégrer plusieurs modèles
- ✓ Analyser les données en temps réel
- ✓ Etablir une connexion entre les objets et le cloud





# Challenges de l'IOT

Probleme d'adressage qui a ete resolu grace a IPv6

Debit

Couverture internet

**Vie privee** : capteurs par tout

**Normalisation:** beaucoup de constructeurs CISCO travaille pour des normes pour l'IOT.

**Securite:** les donnees personnelles sont transmis dans des reseaux etendus.



# *Chapitre II :Les systèmes embarqués et leurs composants*



# Definition d'un systeme embarque

Un système embarqué est un système complexe qui intègre du **logiciel** et du **matériel** conçus ensemble afin de fournir des fonctionnalités données.

→**un ou plusieurs microprocesseurs** destinés à exécuter un ensemble de programmes

Un système embarqué est autonome et ne possède pas des entrées/sorties standards tels qu'un clavier ou un écran d'ordinateur.



# Caractéristiques d'un système embarqué

Les systèmes embarqués fonctionnent généralement en Temps Réel (TR).

Temps réel: Les temps de réponses de ces systèmes sont aussi importants que l'exactitude des résultats



# Caractéristiques d'un système embarqué

## **Un système embarqué:**

- A des ressources limitées
- Système numérique
- Le moins cher possible
- Une puissance de calcul limitée
- Pas de consommation d'énergie inutile
- Exécution de logiciel dédié aux fonctionnalités spéciales
- Une capacité de communication limitée
- Ne possède pas toujours de système de fichiers



# Caractéristiques d'un système embarqué

## Faible coût:

- Solution optimale entre le prix et la performance
- À la portée de toute personne
- Les ressources utilisées sont minimales
- Un système embarqué n'a que peu de mémoire

## Faible consommation:

- Utilisation d'une batterie d'emmagasinage d'énergie
- Gérer la consommation pour rester autonome le plus possible



# Caractéristiques d'un système embarqué

## **Faible encombrement et faible poids**

Minimiser la taille et le poids pour un système embarqué.

## **Fonctionnement en temps réel:**

- ✓ Les applications embarquées doivent répondre rapidement aux événements internes ou externes.
- ✓ Nécessaire dans les applications de système de contrôles



# Caracteristiques d'un systeme embarque

## Environnement:

- ✓ Un système embarqué est soumis à des nombreux contraintes d'environnement.
- ✓ Il doit s'adapter et fonctionner avec eux.

## Exemple:

- La température
- L'humidité
- Les vibrations
- Les chocs



# Contraintes d'un systeme embarqué

un système embarqué doit généralement respecter des contraintes temporelles fortes.

Un système d'exploitation ou un noyau Temps Réel RTOS( Systeme d'exploitation temps reel)



# Domaines applications d'un systeme embarque

- ❖ **Jeux et calcul général** : application similaire à une application de bureau mais empaquetée dans un système embarqué : jeux video, set top box..
- ❖ **Traitements du signal** : radar
- ❖ **Communication et réseaux** : transmission d'information et commutation, téléphonie

# Architecture d'un systeme embarque

Un systeme embarque a 3 couches



Les deux premières couches sont des couches software et la troisième est une couche hardware.

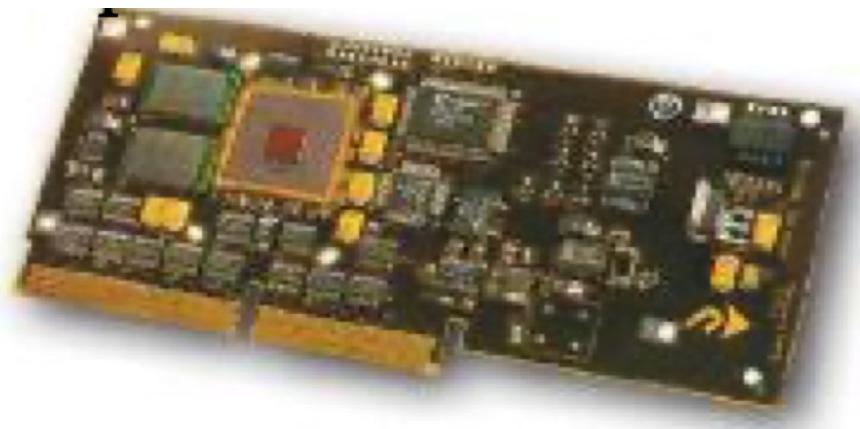
# Système sur puce (System on Chip)

Circuit complexe qui intègre tous les éléments fonctionnels d'un produit sur une même puce.

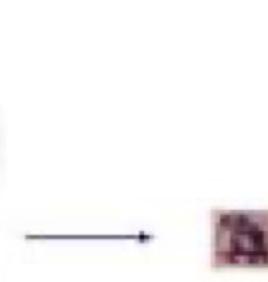
- Modules logiciels
- Mémoires
- Périphériques  
Coprocesseurs matériels
- Modules analogiques ou optoélectroniques

# Système sur puce (System on Chip)

**Objectif:** diminuer au minimum le nombre de composants sur une carte pour mettre tout dans une puce.



Système sur carte



Système sur puce



# Les systèmes d'exploitation mobiles

Un ensemble de programmes responsable de la gestion des opérations de:

- ✓ Contrôle
- ✓ Coordination
- ✓ Utilisation du matériel partage des ressources d'un dispositif entre divers programmes tournant sur ce dispositif.



# Fonctionnalités des systèmes d'exploitation mobiles

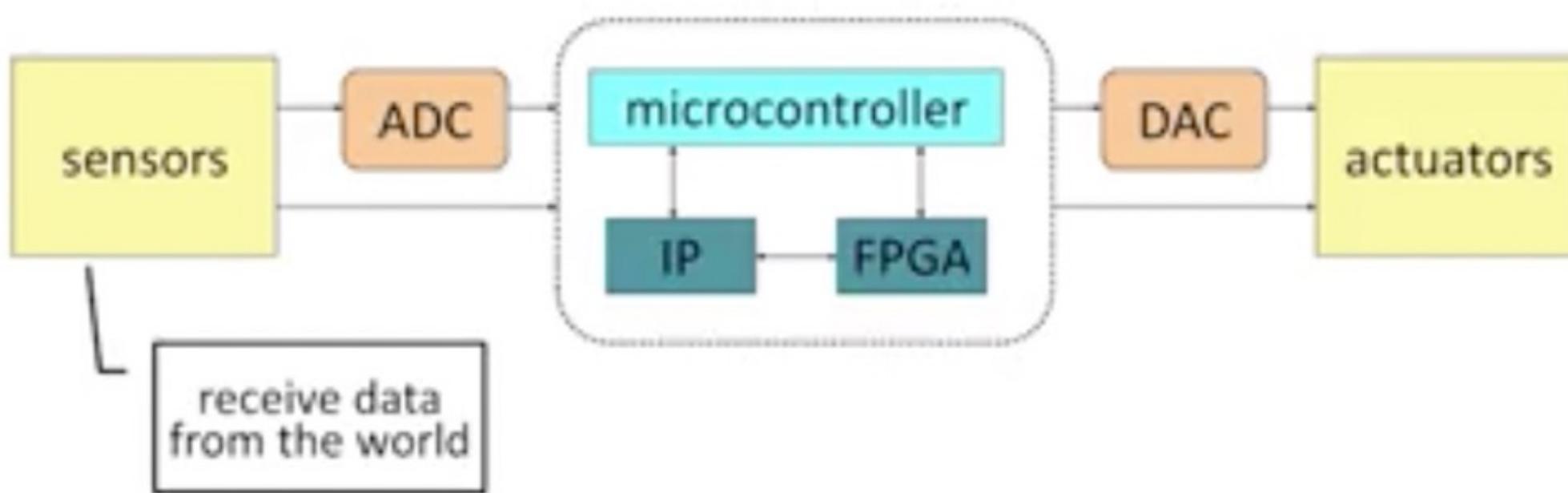
- ✓ La gestion de la mémoire
- ✓ La gestion des microprocesseurs et l'ordonnancement
- ✓ La gestion de système de fichiers
- ✓ La gestion des I/O
- ✓ La gestion de sécurité
- ✓ La gestion de fonctionnalités multimédia



# Exemple de systèmes d'exploitation mobiles

- ✓ Android
- ✓ BlackBerry OS
- ✓ iOS
- ✓ OpenMoko
- ✓ PalmOS
- ✓ HP webOS
- ✓ Symbian OS
- ✓ Windows CE
- ✓ Windows Mobile

# Structure d'un systeme embarque





# Structure d'un systeme embarque

- **Les capteurs** qui reçoivent les informations du monde exterieur. ( button, pushbutton )
- **Les actionneurs** : Effets produits a l'exterieur **LED(light)**

## Exemple:

Les hauts-parleurs sont des actionneurs pour produire du son.



# Structure d'un système embarqué

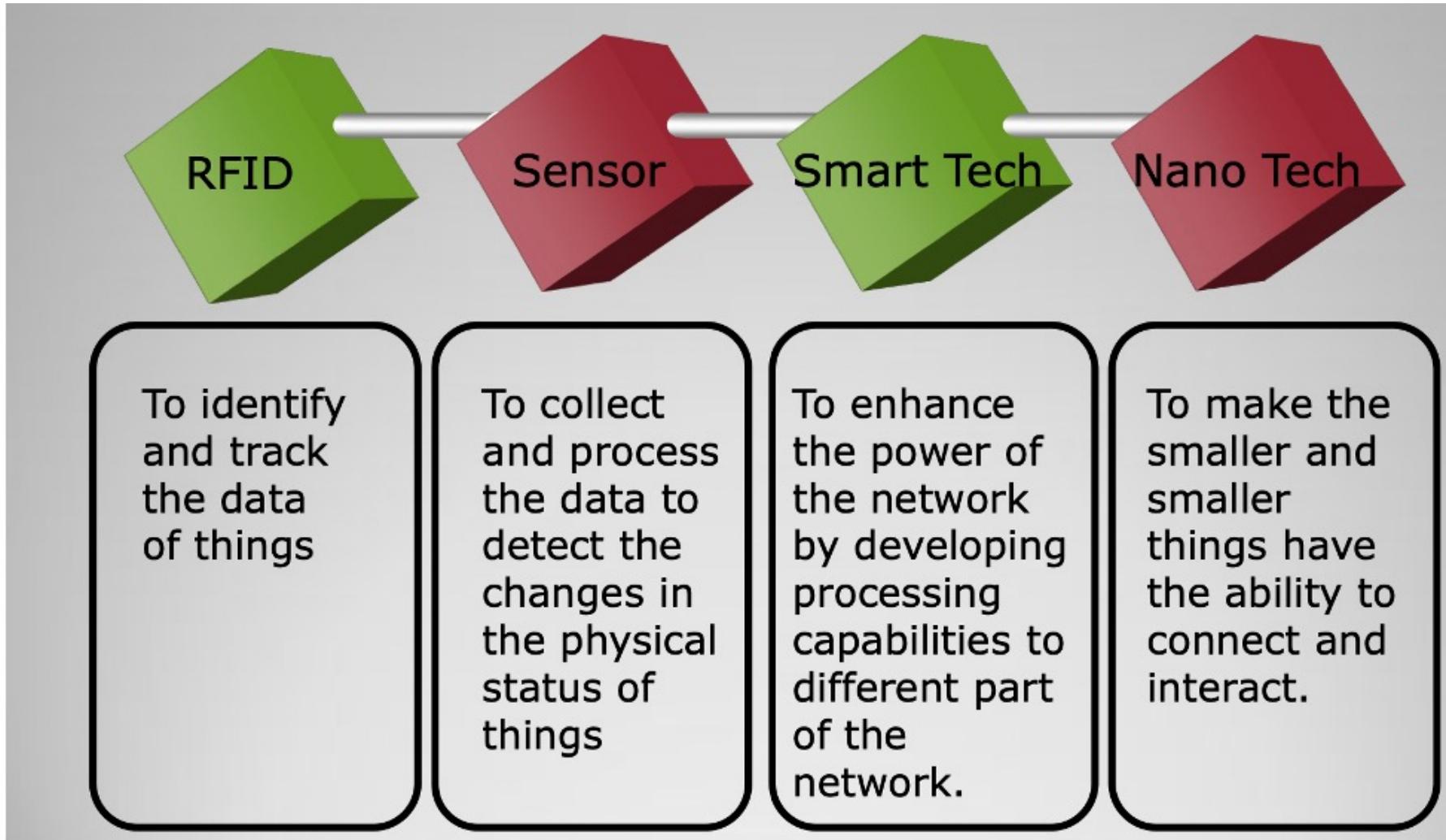
Le composant principal d'un système embarqué est le microcontrôleur.

- ❖ Microcontrôleur est un circuit intégré qui exécute un programme.
- ❖ Lit des entrées des autres composants. Et contrôle les autres composants.
- ❖ Il envoie des commandes et reçoit les données.



# *Chapitre III : Composants hardware, software et les systèmes d'exploitation de l'IOT*

# Comment un système IoT fonctionne?





## Structure de l'IOT

- ❖ **Tagging Things** : Tracabilité et identification des objets par **RFIDs**.
- ❖ **Feeling Things** : collecte des données par **les capteurs**.
- ❖ **Shrinking Things** : technologies de miniaturisation et de nanotechnologies => connexion des petits objets.
- ❖ **Thinking Things** : intelligence intégrée dans les objets.



# *Systemes d'identification des objets: RFID,NFC*



# Tagging Things: Identification des objets

Identifier les objets avant apparition de L'IOT(Identifying things)

- **Radio Frequency Identification (RFID)**
- **Barcode/2D code, IP address**

Ces objets sont capables d'être identifiés(RFID) sur un réseau virtuel

→ Confier une adresse que ça soit un tag ou autre



# Tagging Things: Identification des objets

**Exemple:** Walmart veut déplacer des marchandises d'un emplacement à un autre.

A chaque moment, on doit savoir l'emplacement de cette marchandise, et combien de type d'équipement dans chaque boîte ?

**==>Code à barre**

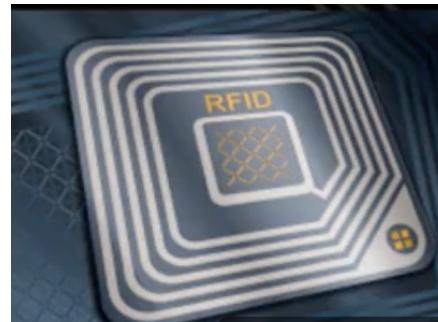
# Tagging Things: Identification des objets

- Chaque box a son propre code à barre qui détermine les informations sur le contenu du box.
- Le scanner pour lire ses informations



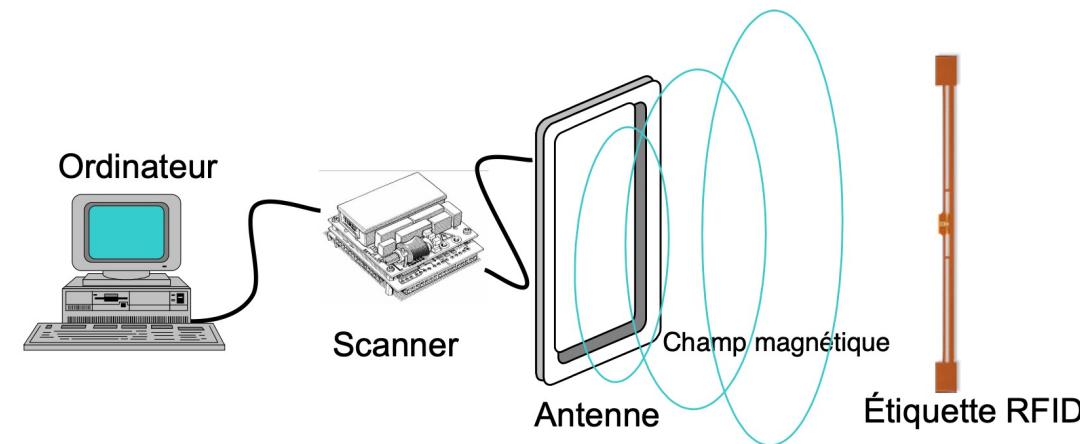
# RFID: Identification des objets

De nos jours, les codes à barres sont de moins en moins utilisés et sont remplacés par **RFID une version électronique du code à barre.**



# TECHNOLOGIE RFID

Est une technologie d'identification sans contact et sans ligne. Elle permet de lire et d'écrire des informations ou des données dans des puces électroniques en utilisant les ondes électromagnétiques.





# RFID

RFID a:

- un processeur dynamique alors que le code à barre contient des informations limitées et fixes.
- une flash memory qui charge les changements et effectue les mises à jour des informations.

# RFID

RFID ( Radio Frequency Identification ) récupérer des données à distances grâce à un système composé :

- 1-Une étiquette radio contenant des informations
- 2-Un lecteur récupère les informations d'une étiquette radio à distance



# Caractéristiques du RFID

- Fonction d'identification (objet, animal, humain)
- Faible volume de données et liaison avec une base de données distante
- Pas ou peu de sécurité





# Exercice

- Citez quelques exemples d'utilisation de RFID dans les aeroports, la sante...., paiement bancaire)

# Exemple d'applications RFID



500 000 poubelles britanniques équipées  
de **puces RFID**  
**Identifier les manquements au recyclage**  
**Réduire le volume d'ordures ménagères**



400 000 bouteilles réutilisables d'Air Liquide  
**avec des étiquettes RFID.**

# Exemple d'applications RFID



**L'étiquette RFID( tag) sur les bagages :**

- ❖ réduire le taux d'erreur d'acheminement.
- ❖ le contrôle par les systèmes d'inspection et de détection d'explosifs

**L'aéroport Charles-de-Gaulle (Paris) (Taxi)**

- ➔ Chaque voiture dispose **d'un étiquetage RFID,**
- ➔ Surveiller en temps réel les mouvements des véhicules

# Paiement mobile RFID



La banque postale teste le paiement mobile pendant un an depuis 2013 à Bordeaux et Caen.

Règlement des achats chez les commerçants équipés d'un terminal sans contact.  
montants >20 euros → un code secret

une application (Visa PayWave)



# Caractéristiques de RFID

RFID est une technologie d'identification sans contact

- ✓ Identification de plusieurs étiquettes à la fois (10-100/s)
- ✓ Chaque étiquette identifiée est unique (unique ID)



## Etiquettes RFID Tag

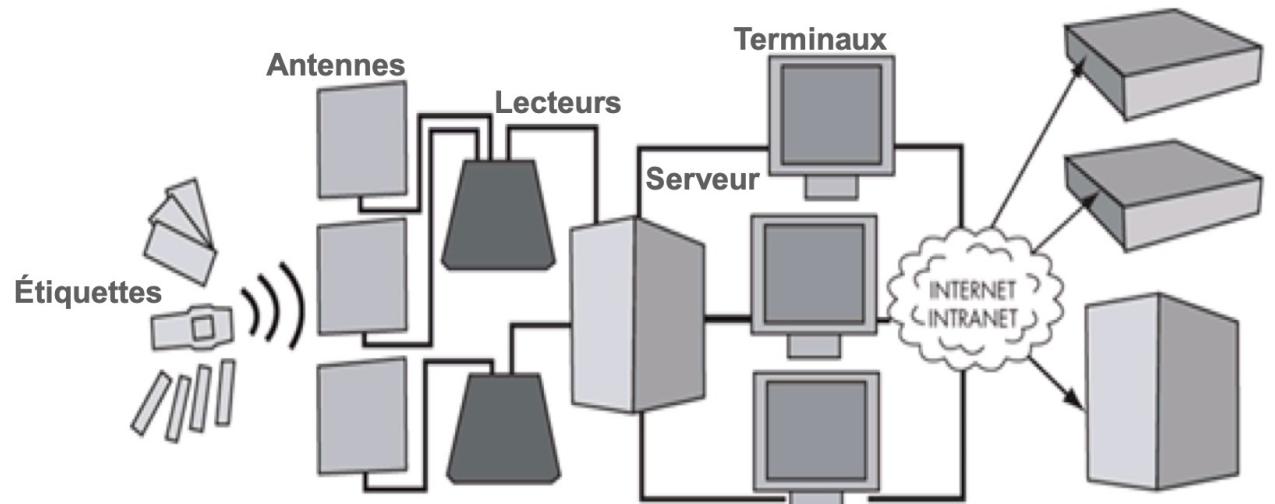
- ✓ Les informations écrites sur l'étiquettes peuvent être modifiées
- ✓ Les étiquettes possèdent une bonne capacité de stockage
- ✓ Les étiquettes peuvent enregistrer des informations dynamiques.

RFID tag peut être connecté facilement à Internet.

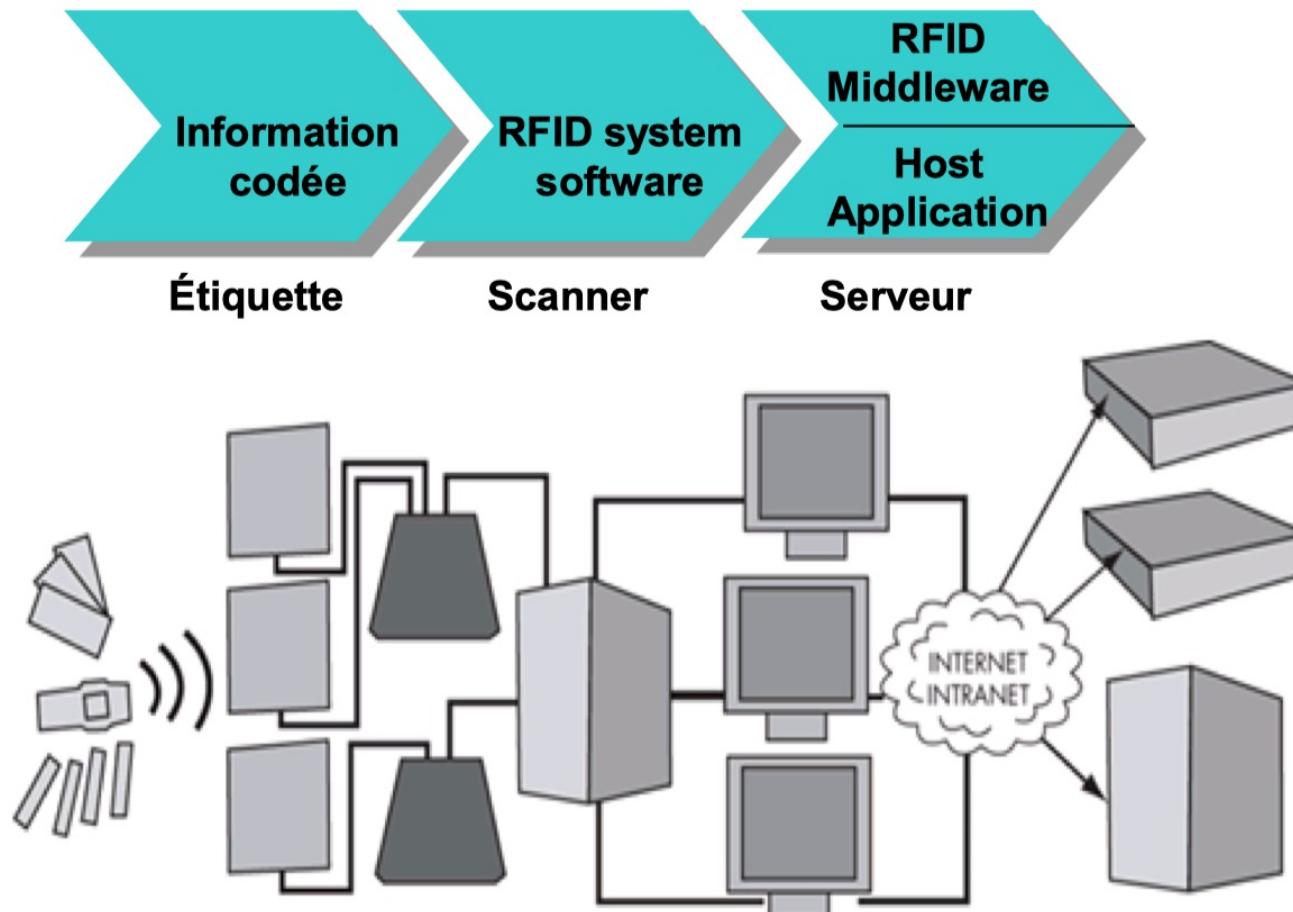
**Exemple : IoT RFID scanner connecté au cloud services de Wal-Mart**

# Les composants de RFID

- 1- Une ou plusieurs étiquettes
- 2- Une ou plusieurs antennes
- 3- Un ou plusieurs scanners
- 4- Un serveur
- 5- Des terminaux
- 6- Un réseau internet ou intranet



# Les composants de RFID





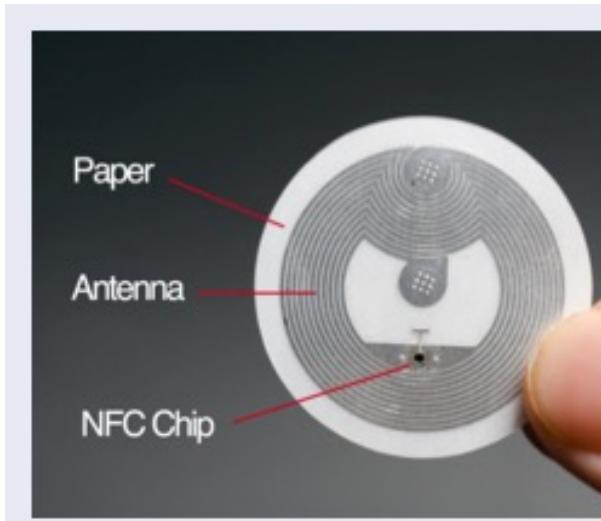
## Avantages RFID

- ❖ Mise à jour facile des données contenues dans les puces RFID
- ❖ Capacité de stockage:
- ❖ Vitesse d'enregistrement (lecture/écriture) rapide
- ❖ Sécurité: possibilité de chiffrer les données
- ❖ Souplesse dans le positionnement de l'étiquette
- ❖ Durée de vie importante (jusqu'à 10 ans)



# NFC

L'interface NFC (Near Field Communication) : une technologie dérivée de RFID permettant une communication à très faible distance entre deux objets



# NFC

NFC (Near Field Communication) est une technologie favorisant des interactions bidirectionnelles simples et sûres entre deux dispositifs électroniques.

- ✓ Effectuer des transactions par paiement sans contact
- ✓ Accéder à des contenus numériques
- ✓ Se connecter à des dispositifs électroniques.





# Architecture d'un système embarqué (IOT)

Une taille limitée (pour offrir un encombrement réduit)

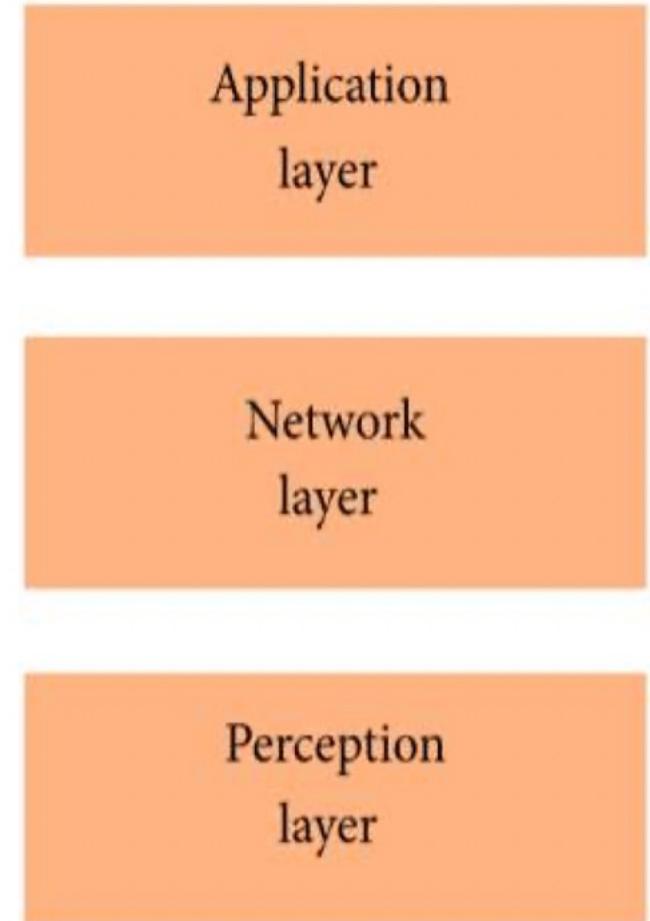
Une consommation énergétique assez faible.

## **Exemple de système embarqué: Robot aspirateur**

Equipé de capteurs pour la détection d'obstacles. ➔ changer de trajectoire pour poursuivre sa mission

# Architecture d'un système embarqué (IOT)

- **La couche perception:** recueil des informations sur l'environnement par les actionneurs et les capteurs.
- **La couche réseau:** la connexion du transport et du traitement des données issues des capteurs et actionneurs.
- **La couche application:** fournir à l'utilisateur des services spécifiques et applications intelligentes.





# Composants d'un systeme IOT

Le systeme complet d'un objet connecte integre 4 composants distincts:

- ✓ Des capteurs
- ✓ Une connectivite
- ✓ Un traitement de l'information( Microcontroleur)
- ✓ Un interface utilisateur ( Systeme d'exploitation)

# Composants d'un objet physique

Un objet connecté fonctionne grâce à un réseau (wifi, bluetooth)

- ✓ Muni de nombreux capteurs pour recueillir les informations (capteurs, actionneurs)
- ✓ Une carte électronique pour prétraiter les informations (microcontrôleur)
- ✓ Un dispositif de transmission et de communication

# Caractéristiques d'un Objet connecté

- **Capteurs** : recueillir des informations
- **Actionneurs** : agir sur le monde physique en modifiant son état.
- **Intelligence** : traitement local des données
- **Communication** : codage et transmission des données, protocoles standards ou dédiés, communication filaire ou sans fil.
- **Énergie** : alimentation de la plateforme en énergie électrique





# *Les capteurs et les actionneurs*



# Les capteurs ou les transducteurs(Sensors)

Un capteur ou un transducteur est un appareil qui convertit un type d'énergie en un autre.

**Les capteurs** détectent les formes d'énergie tel que la lumière ou la force et convertissent cette énergie en informations numériques.

**Les actionneurs** est un transducteur qui absorbe les informations et génère une forme d'énergie.

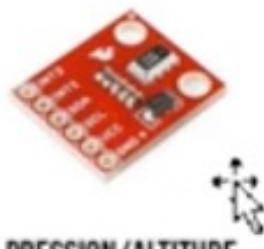
**Exemple:** Votre smartphone vibre dans la poche grâce aux actionneurs

# Exemples de capteurs

## Capteurs d'environnement et de mouvement



CAPTEUR DE FLAMME



PRESSION/ALTITUDE



CAPTEUR DE SON



CAPTEUR D'HUMIDITÉ DES SOLS



CAPTEUR UV



CAPTEURS DE GAZ



THERMOMÈTRE INFRAROUGE



CAPTEUR LUMIÈRE RGB



CELLULE PHOTOÉLECTRIQUE



THERMISTOR



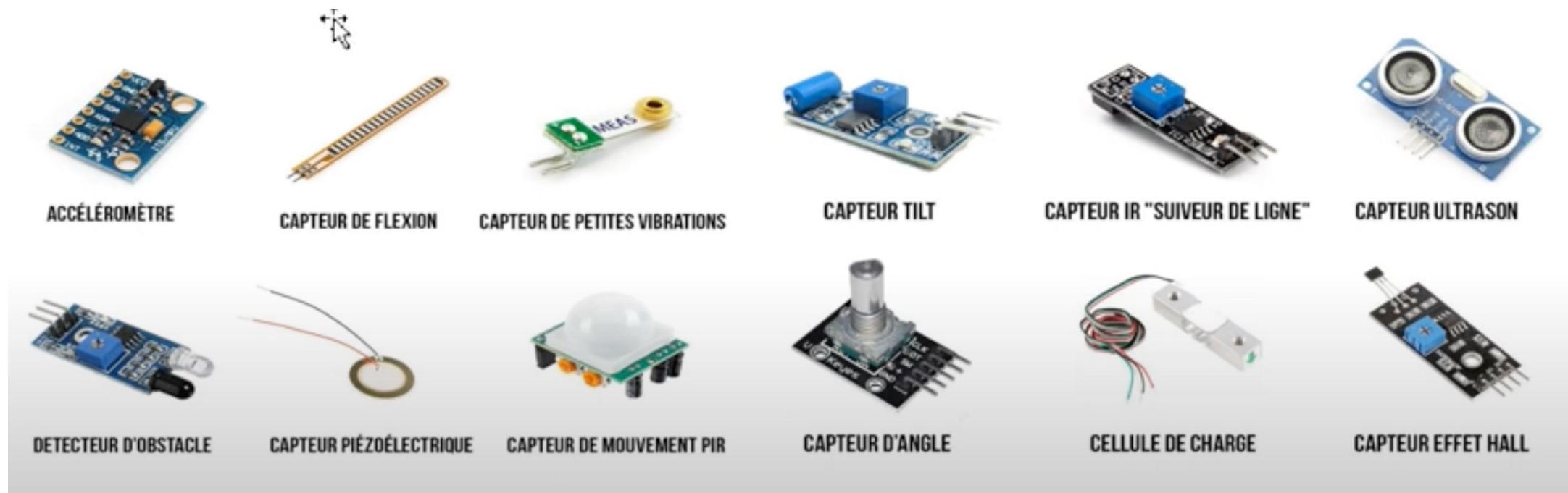
CAPTEUR DE PLUIE



COMPTEUR GEIGER

# Exemples de capteurs

## Capteurs de mouvement



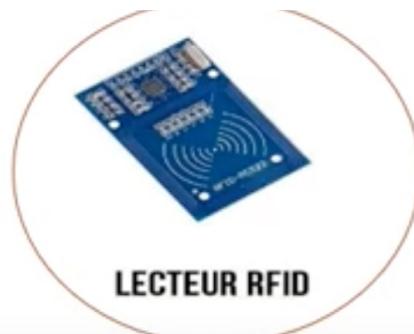
# Autres capteurs



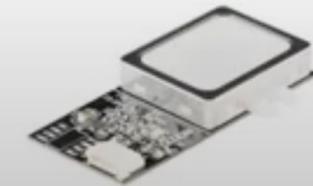
CAMÉRA THERMIQUE



CAMÉRA INFRAROUGE



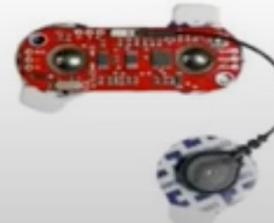
LECTEUR RFID



LECTEUR EMPREINTE DIGITALE

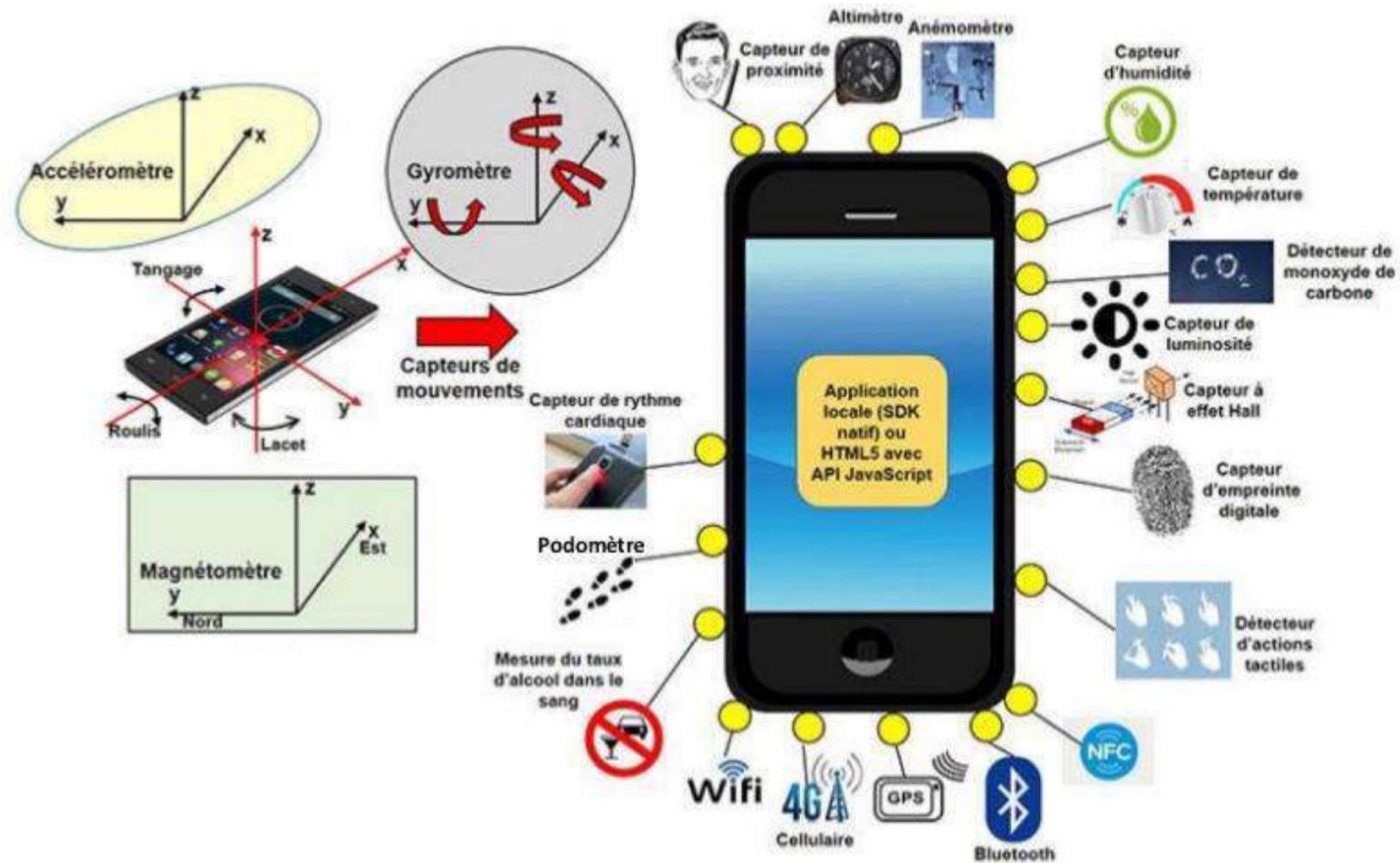


CAPTEUR DE PULSATIONS



CAPTEUR COURANT MUSCULAIRE

# Exemple: capteurs de smartphone





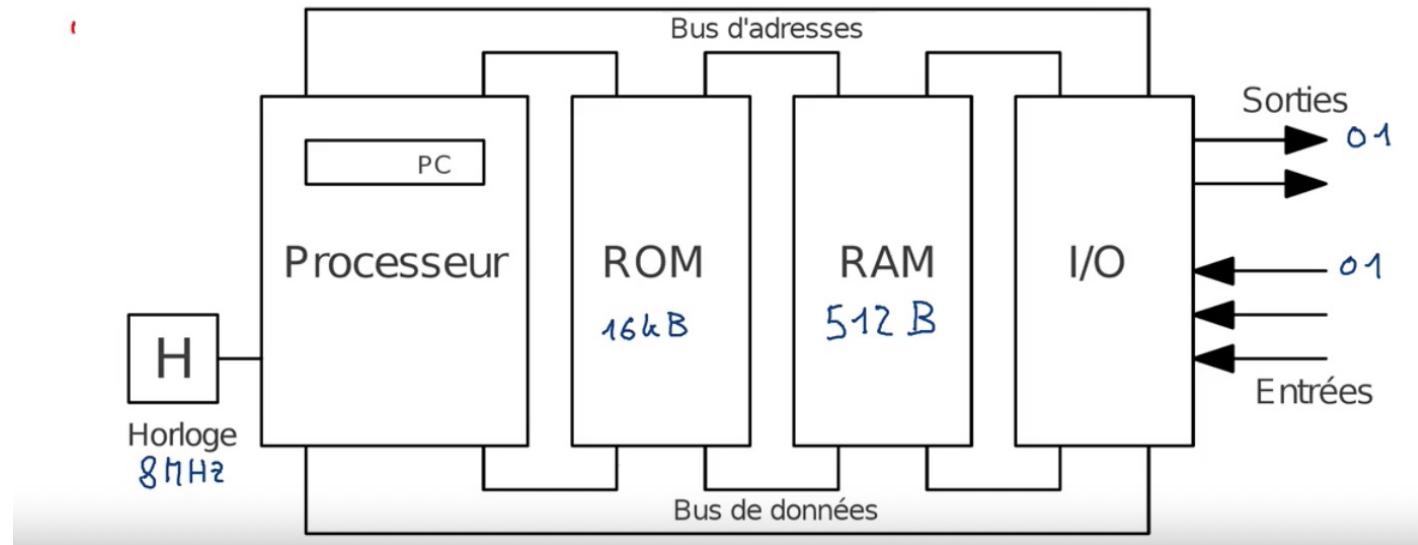
# *Le microcontrôleur*

# Le microcontroleur: le cerveau d'un objet

- Est le composant de IoT responsable du traitement de l'information.
- Comparaison des seuils
- Assurer la connectivité de l'objet au réseau ou à d'autres objets.
- Un objet peut être de taille minuscule d'une pièce de monnaie comme il peut être géant comme un réfrigérateur.
- Effectuer une analyse de données.

# Architecture d'un microcontrôleur

Moins de calculs  
Moins de puissance  
Moins chers





# Definition Microcontrôleur

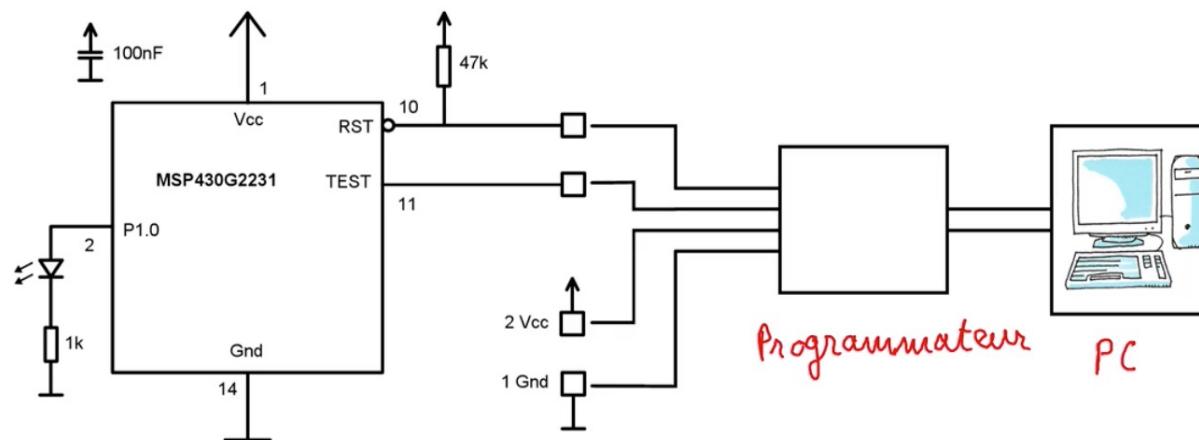
Un seul circuit intégré compose de:

- ❖ micro-processeur.
- ❖ mémoire vive (RAM) pour les données.
- ❖ mémoire morte (ROM) pour les programmes.
- ❖ unités périphériques (timers, convertisseurs analogiques numériques (CAN)).
- ❖ interfaces de communication, séries ou parallèles.
- ❖ Piloter un système embarqué, seul, ou en collaboration avec d'autres.

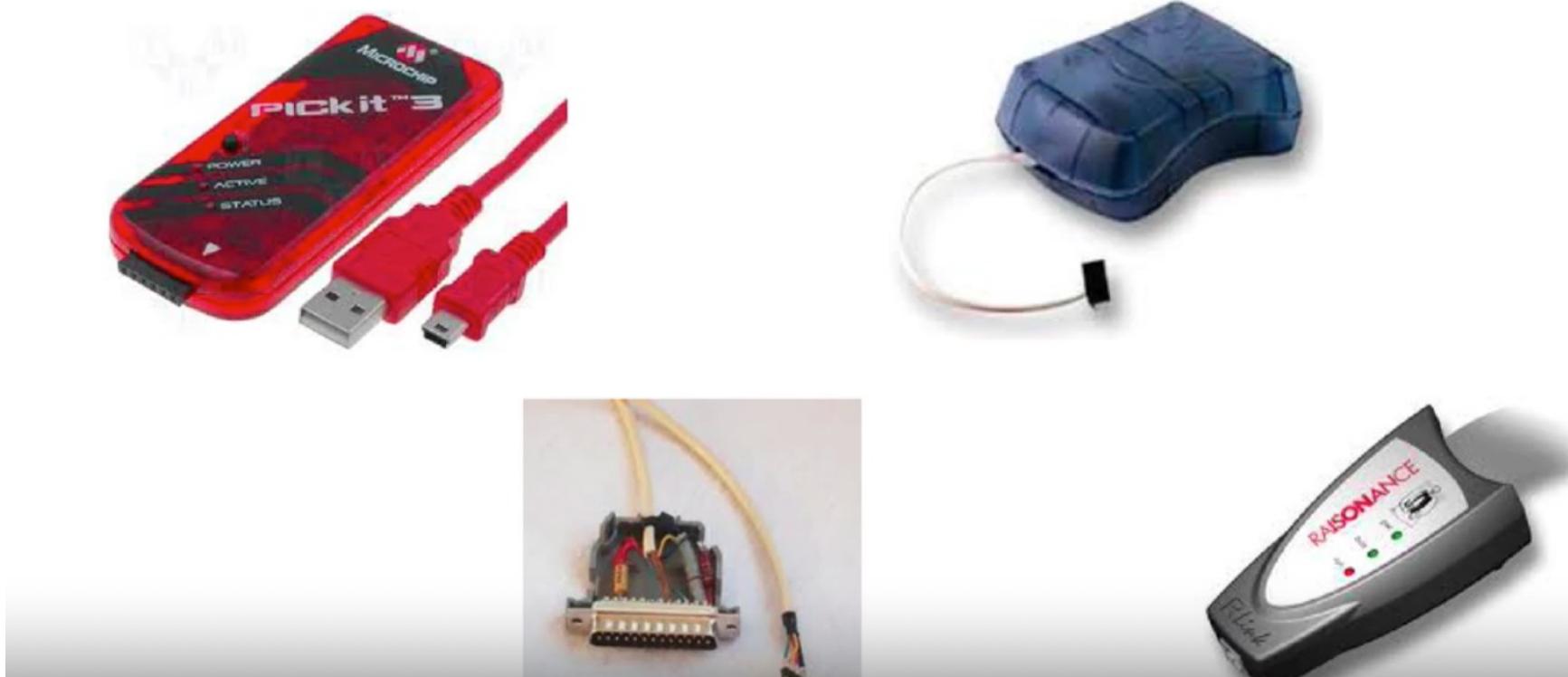
# Definition Microcontroleur

Circuit intégré → Mise en oeuvre materielle (electronique)  
Système informatique → Mise en oeuvre logicielle

Pour programmer un microcontroleur, nous aurons besoin d'un PC et d'un programmateur.



# Programmateur

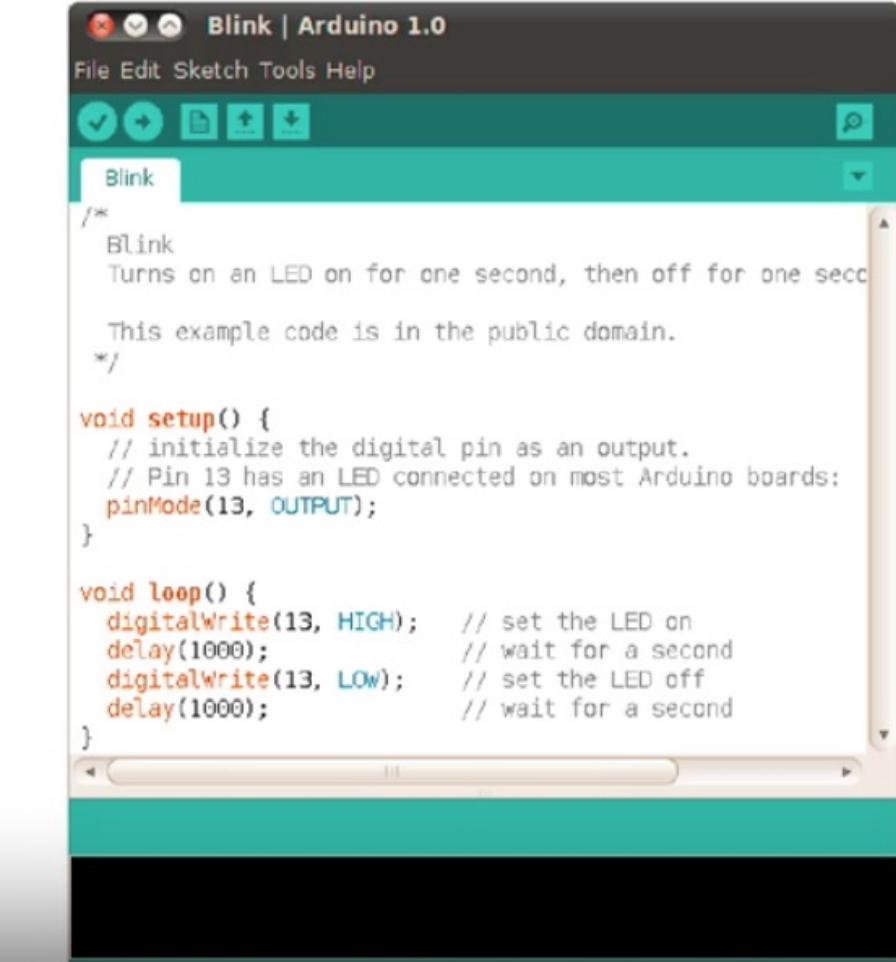


# Carte microcontrôleur

Il existe des cartes électroniques avec des connecteurs qui contiennent à la fois des microcontrôleurs et les programmeurs.



# Environnement de développement( MSP430- Arduino)



```
/*
Blink
Turns on an LED on for one second, then off for one second, repeating the cycle.
This example code is in the public domain.
*/

void setup() {
    // initialize the digital pin as an output.
    // Pin 13 has an LED connected on most Arduino boards:
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);      // set the LED on
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // set the LED off
    delay(1000);                // wait for a second
}
```

Done uploading.  
Erasing...  
Programming...  
Done, 584 bytes written



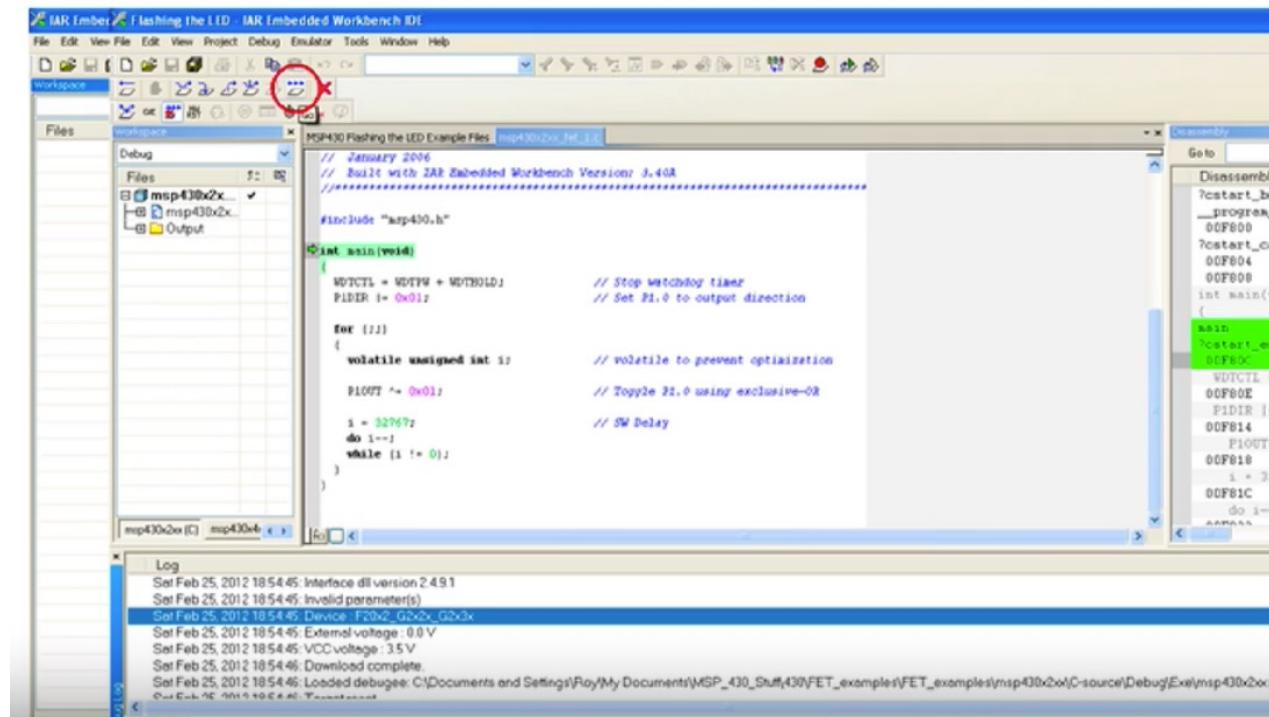
# Environnement de développement( MSP430- Arduino)

**Environnement de développement:** permet d'écrire des programmes et de les convertir sous forme de langage machine, ensuite de les envoyer sur la mémoire du microcontrôleur.

Développer des systèmes à base de microcontrôleurs.

# Environnement de développement

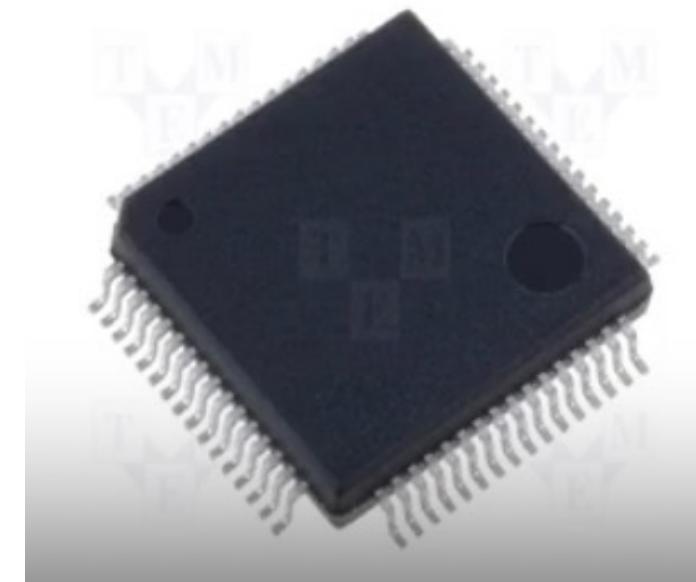
Il existe des environnements de développement plus complexe Code Composer Studio



# Choix du microcontroleurs

Il existe un très grand nombre de microcontroleurs

- ❖ Petits circuits de 6 pattes
- ❖ Circuits avec des centaines de pattes





# Criteres de choix d'un microcontroleur

- ❖ Nombre de pattes d'entrées-sorties
- ❖ Taille de la memoire de programmee (ROM)
- ❖ Taille de la memoire vive(RAM)
- ❖ Consommation electrique(courant consomme)
- ❖ Disponibilite du materiel et de l'environnement de developpement.

# Histoire et evolution des microcontroleurs

Les premiers microcontroleurs datent des années 1970.  
**(Texas TMS1000 et Intel 4048, mask-ROM)**



# Histoire et evolution des microcontroleurs

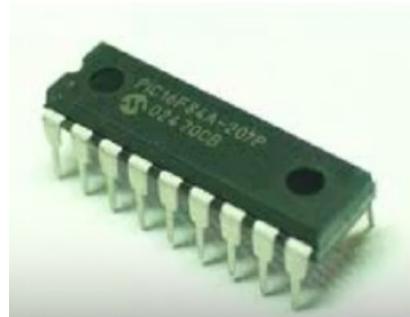
Les microcontroleurs avec des mémoires effacées par Ultra violet  
(Intel 8748, Motorola 68705)

Facile de mise en œuvre, écrire un programme le teste puis l'effacer



# Histoire et evolution des microcontroleurs

EEPROM en 1993. ROM programmable effacable électroniquement





# Familles de microcontrôleurs actuels

## 8bits

- ✓ PIC Microchip
- ✓ AVR Atmel qui a été utilisé pour Arduino
- ✓ Dérivés d'un ancien processeur Intel 80C51

## 16bits

dsPIC

MSP430 Texas Instrument

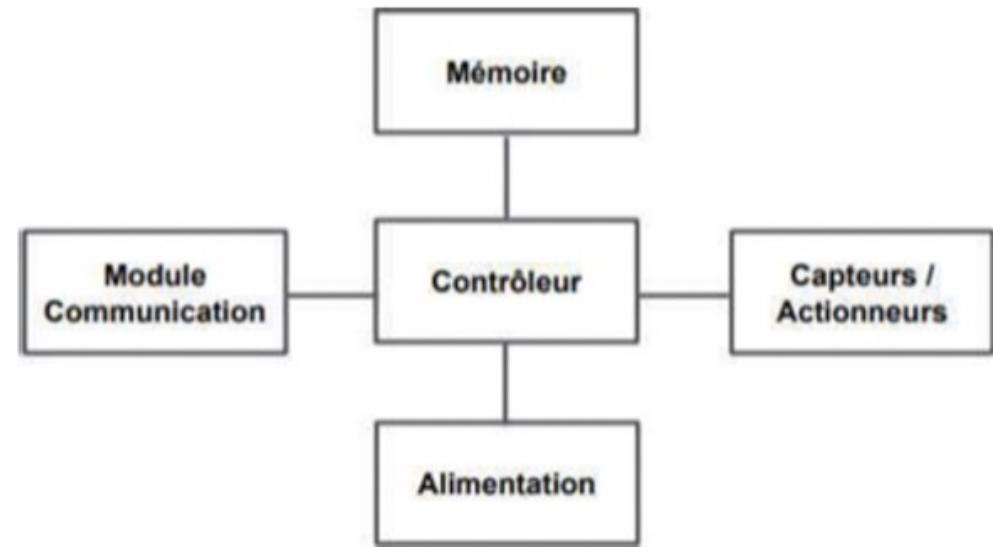
## 32bits

AVR32, PIC32, MIPS, PowerPC

ARM, NXP, Stmicro, Texas, Samsung

# Les cartes microcontrôleurs

Avec les actionneurs et les capteurs, on associe des cartes à microcontrôleurs pour extraire les informations utiles ou commander les actionneurs



# Exemple de carte microcontrôleur



Arduino

- Basé sur un µc Atmega (Single core, 16MHz)
- Connexion simple
- Programmation facile
- Bon choix pour les capteurs



NodeMCU

- Basé sur le µc ESP8266 (Single core, 80MHz)
- Programmation facile
- Intègre WiFi



STM32

- Basé sur un µc ARM 32 bits (24-400MHz)
- Bon choix pour les capteurs
- Bon choix pour le traitement local



Pycom Lopy4

- Basé sur le µc ESP32(Dual core, 160-180MHZ)
- Programmation facile
- Connectivité : WiFi, Bluetooth, Sigfox, LoRa



# *Les systèmes d'exploitation pour les plateformes IOT*



# Solutions technologiques IOT

**Systèmes construits autour d'un système d'exploitation embarqué (Raspberry Pi, Beaglebone, etc.)**

- ✓ Ouverts, puissants, langages de programmation multiples
- ✓ coût relativement élevé, interfaçage plus difficile.



# Solutions technologiques IOT

## **Systèmes dédiés compacts à logiciel propriétaire (ARDUINO, GENUINO, INTEL GALILEO, ESP8266 etc.)**

- ✓ Très réactifs, très faible coût, fonctionnement plus robuste  
interfaçage aisément, prise en main très rapide.
- ✓ Moins puissants, langages de programmation plus limités.

# Solutions technologiques IOT

- Des OS en open source pour les systemes IOT
  - **FreeRTOS** adapté au temps reel
  - **Contiki** pour les capteurs environnementaux
  - **TinyOS** pour les systèmes miniatures disposant de très peu de mémoire
  - **RIOT** qui se présente comme le Linux des objets connectés basé sur une architecture à micronoyau
  - **VxWorks** pour les objets avec une interface visuelle, Ubuntu et sa variante Ubuntu Core



# TinyOS

TinyOS est un système d'exploitation dédié aux capteurs réseaux sans fil.

**Un réseau de capteurs sans fil:** est un réseau ad hoc avec un grand nombre de nœuds qui sont des micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome.



## Etude de cas: Smart Home

### **Conception et réalisation d'une centrale embarquée de la domotique « Smart Home »**

La majorité des individus, et plus particulièrement les personnes âgées, passent beaucoup de leur temps à domicile, d'où l'influence considérable de l'habitat sur la qualité de vie.



# Domotique

La domotique ou encore la maison intelligente est définie **comme une résidence équipée de technologies d'informatique, d'automatisme et d'électronique**, ambiante qui vise à assister l'habitant dans les situations diverses de la vie domestique en améliorant le confort et simplifiant un certain nombre de tâches.



# Domotique

## **Fonction de confort:**

optimiser de l'éclairage

programmer les équipements électroménagers et multimédia.

## **Fonction d'économie de l'énergie**

Mettre en veille les dispositifs de chauffage en absence d'habitants.

Diminuer les gaspillages de ressources énergétiques suivi des consommations et optimisation des tarifs.



# Domotique

## **Fonction de sécurité:**

La protection des individus.

Des systèmes capables d'anticiper des situations potentiellement dangereuses

Réagir aux événements mettant en danger l'intégrité des personnes.

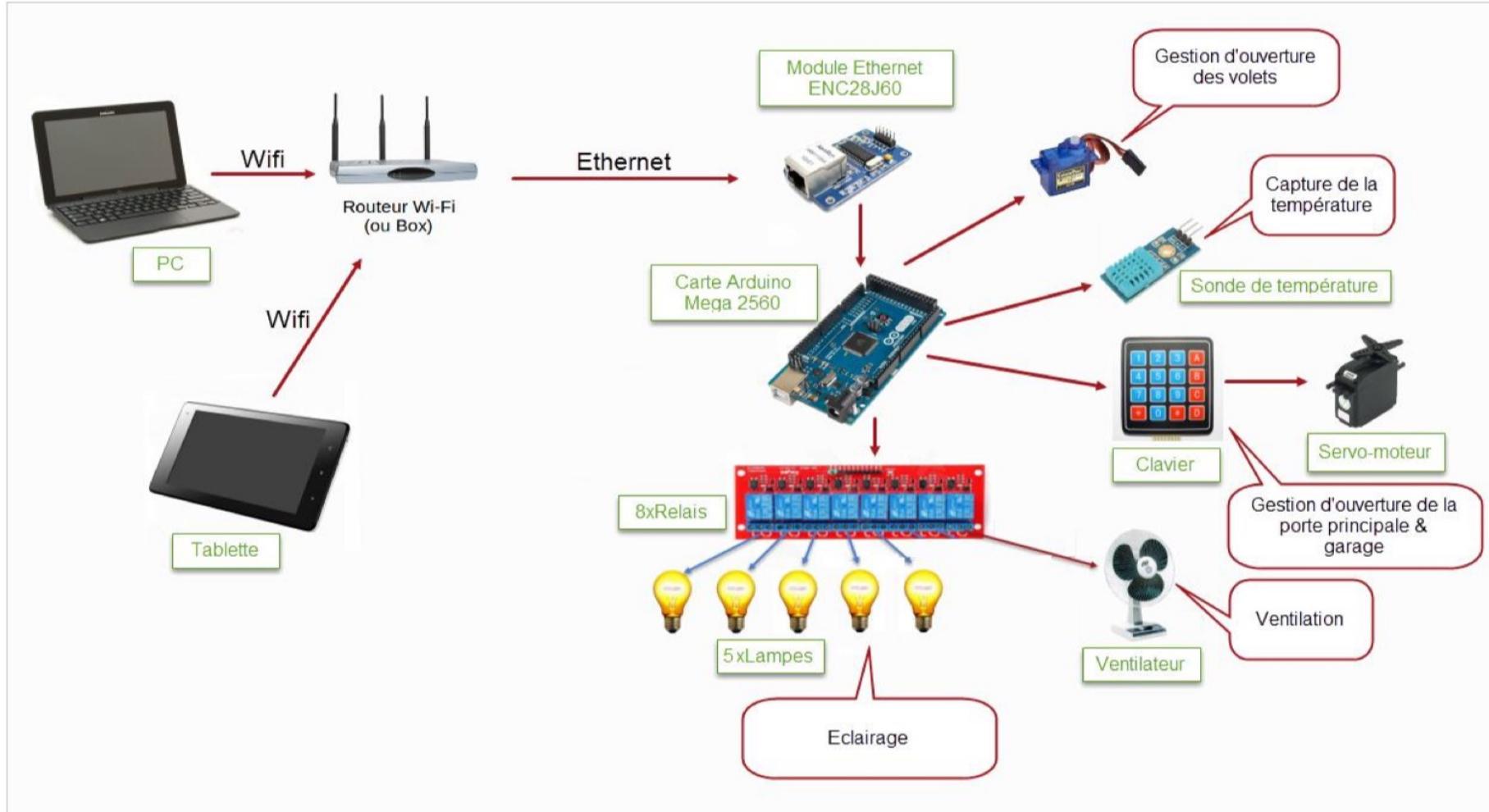


# Domotique

## **La réalisation d'un contrôleur intelligent:**

- ✓ Piloter la domotique à partir d'une application à distance
- ✓ Créer un Web Server entre les différents équipements du système domotique

# Structure du projet proposé



# Fonction de gestion d'éclairage

## **Fonction d'ouverture de la porte principale et du garage**

Cette fonction assure l'ouverture de la porte principale de l'habitat ainsi du garage d'une manière plus sécurisé en adoptant un système d'accès par clavier.





## Fonction de gestion ouverture des volets des fenêtres

L'utilisateur peut contrôler plus aisément l'ouverture des fenêtres tout en appuyant sur un simple bouton sur l'application de commande avec son Smart phone ou sur tablette.



# Fonction d'acquisition de la température et ventilation

L'acquisition de la température se fait via un capteur de température DHT11 pour contrôler le climat à l'intérieur de l'habitat avec de la ventilation.



# Fonction de détection de fuite de gaz

Cette fonction permet de détecter les fuites de gaz via un capteur de gaz MQ-2. Il est apte à détecter le GPL, le butane, le propane, le méthane, l'alcool, l'hydrogène, la fumée.





# Etapes d'élaboration d'un projet IOT

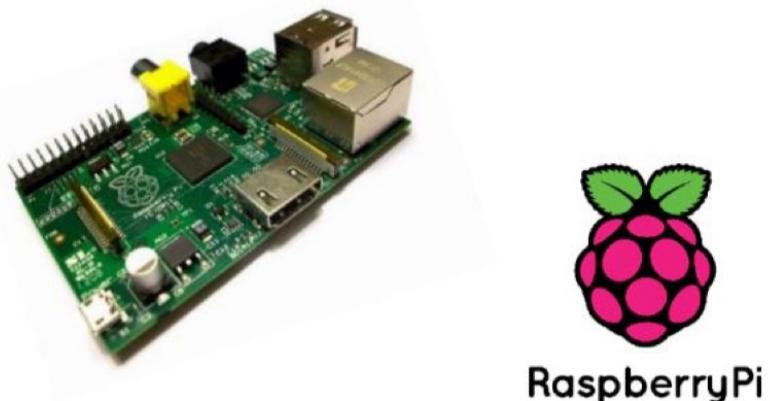
## **Choix de la carte programmable:**

Dans cette partie nous avons cités les différents types de cartes programmables et ses caractéristiques dont le but de sélectionner la plus optimale pour notre projet.

# Choix de carte programmable

## Carte RASPBERRY\_PI

il s'agit d'une carte mère seule avec un processeur ARM11 à 700 MHz. Elle inclut 1, 2 ou 4 ports USB, un port RJ45 et 256 Mo de mémoire vive pour le modèle d'origine (512 Mo sur les dernières versions).





# Choix de carte programmable

## Caractéristiques matérielles:

Taille : 85.60 mm × 53.98 mm

Poids : 45 g

Processeur : 700 MHz ARM1176JZF-S Core (ARM11)

Système sur puce (Soc) : Broadcom BCM2835 ;

Processeur graphique (GPU) : décodeur Broadcom VideoCore IV, API logicielle vidéo OpenGL ES 2.0, MPEG-2 et VC-1, décodage vidéo 1080p30 h.264/MPEG-4 AVC ;



# Choix de carte programmable

## **Caractéristiques matérielles:**

Mémoire (SDRAM) : 256 Mo [Modèle A] ou 512 Mo [Modèle B] partagée avec le processeur graphique ;

Ports USB 2.0 : 1 [Modèle A] ou 2 [Modèle B] ;

Sortie vidéo : RCA Composite (PAL et NTSC) et HDMI (rev 1.3 & 1.4)

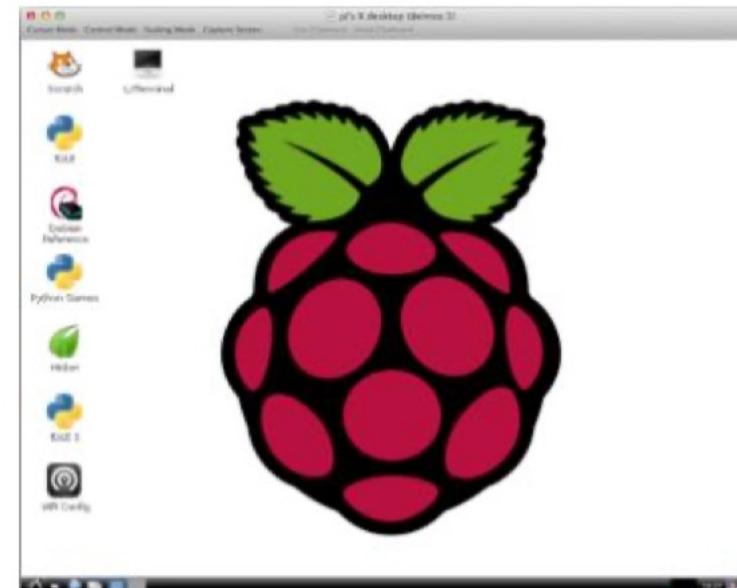
Sortie audio : 3.5 mm jack, HDMI

Unité de lecture-écriture de carte mémoire : SDHC / MMC / SDIO ;

# Choix de carte programmable

## Systemes d'exploitation

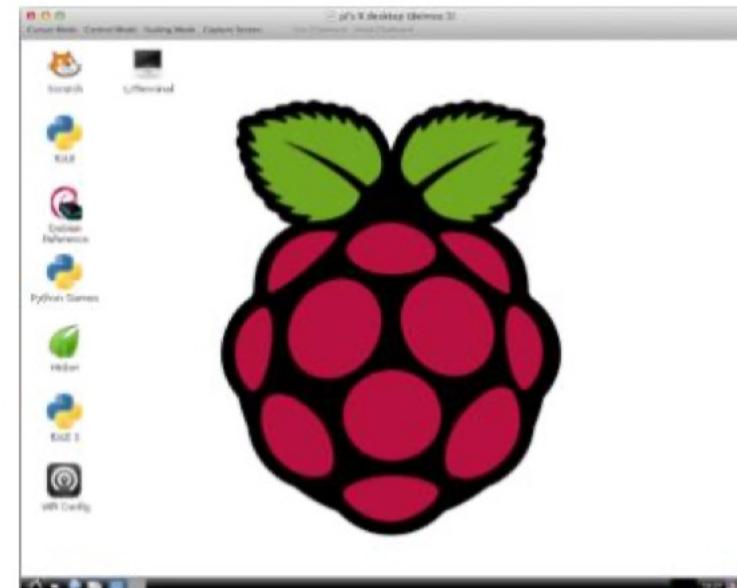
Le systeme d'exploitation utilise est Raspbian



# Choix de carte programmable

## Systemes d'exploitation

Le systeme d'exploitation utilise est Raspbian



# Choix de carte programmable

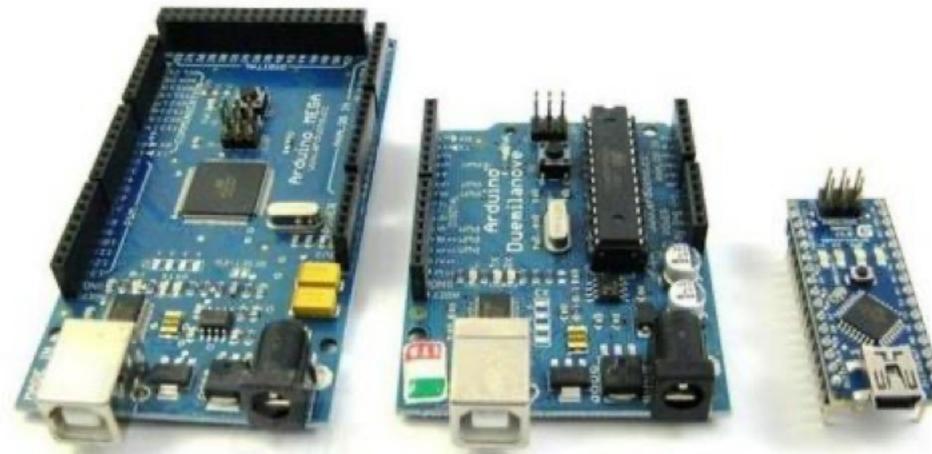
**Boitier:** elle est extrêmement légère

Il est donc conseillé d'avoir un boîtier pour la protéger.



# Carte Arduino

Arduino est un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique.





# Carte Arduino

repose sur un circuit intégré associée à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externs.

**Côté entrées:** des capteurs qui collectent des informations sur leur environnement comme la variation de température.

**Côté sorties:** des actionneurs qui agissent sur le monde physique telle une petite lampe qui produit de la lumière, un moteur qui actionne un bras articulé, etc.



# Series Carte Arduino

- ❖ Arduino UNO
- ❖ Arduino Nano
- ❖ Arduino Lilypad
- ❖ Arduino DUE
- ❖ Arduino Méga 2560

# Arduino Méga 2560

- ❖ L'Arduino Méga 2560 est une carte électronique basée sur le microcontrôleur ATmega2560.
- ❖ 54 broches numériques d'entrée / sortie
- ❖ 16 entrées analogiques
- ❖ un résonateur céramique (Quartz) à 16 MHz
- ❖ une connexion USB
- ❖ une prise d'alimentation
- ❖ un connecteur ICSP, et un bouton de réinitialisation.





# Caractéristiques de Arduino

**Pas cher** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes.

**Multi-plateforme** : tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.

**Un environnement de programmation clair et simple** : facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.



# Caractéristiques de Arduino

**Logiciel Open Source et extensible :** Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmateurs expérimentés.



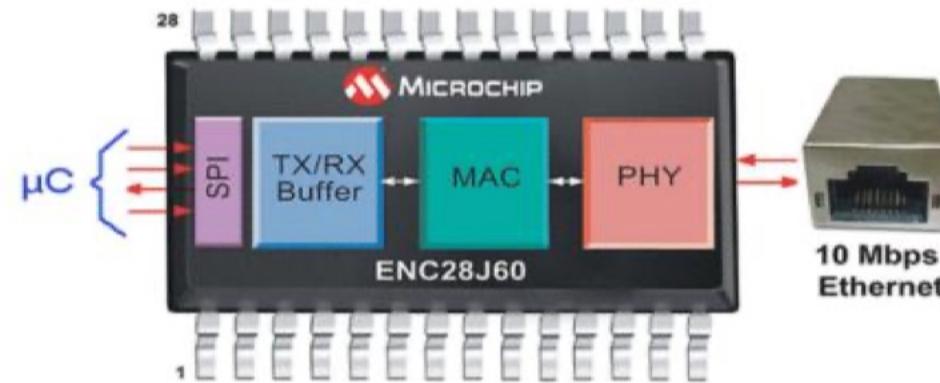
# Choix des organes du système domotique

## **Module Ethernet ENC26J80: Communication Ethernet avec la puce ENC28J60**

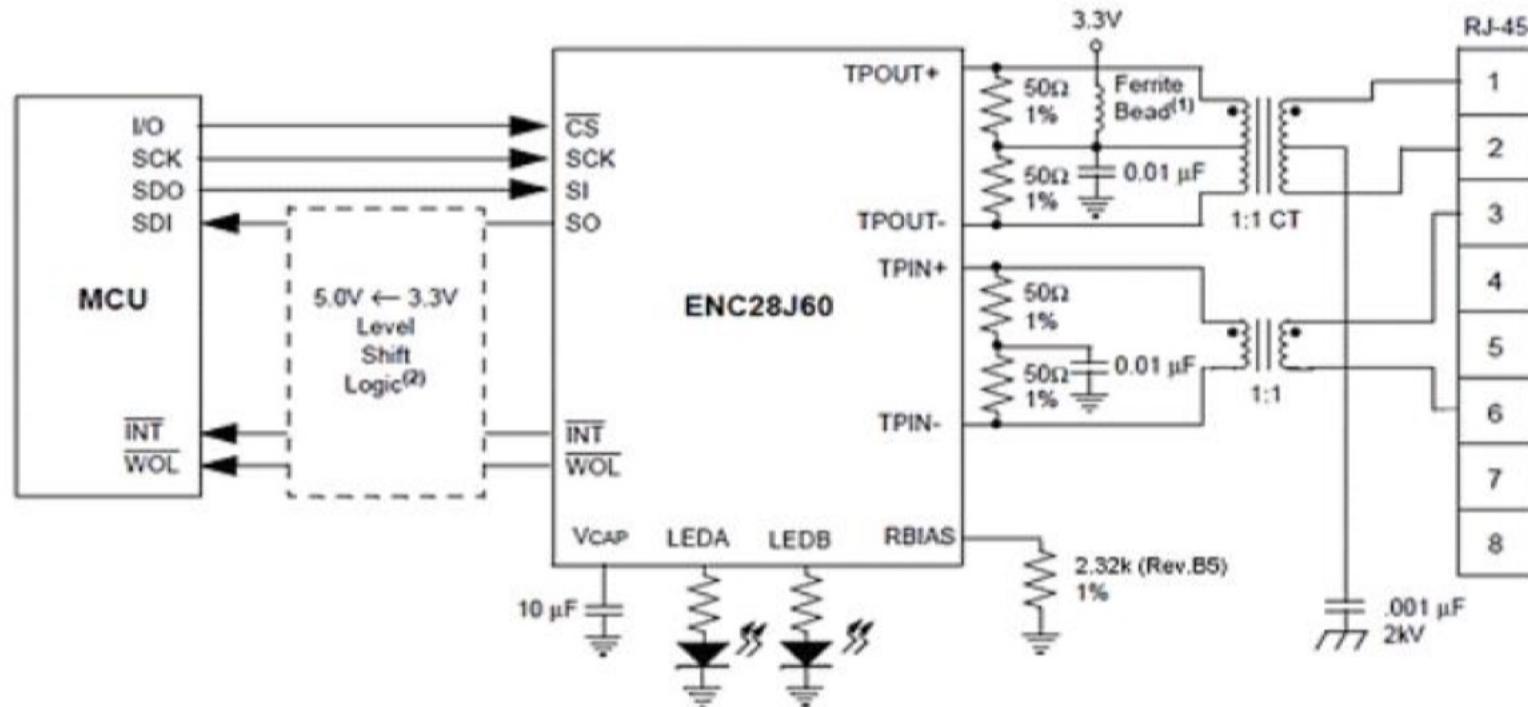
En 2004, Microchip présente **le premier contrôleur Ethernet 10BASE-T** (10Mbits/s) référencé ENC28J60 se laissant facilement piloter via un bus SPI (Serial Programming Interface) par un microcontrôleur et ne comportant que 28 broches pour un boîtier au format DIL.

# Choix des organes du système domotique

l'ENC28J60 permet une intégration sensiblement facilité dans le réseau domestique en lui couplant un microcontrôleur afin de piloter les **fonctionnalités Ethernet**. Disponible en boîtier DIL (DIP), il reste la solution la moins chère pour la mise en œuvre d'un projet connecté au réseau Ethernet.



# Interconnexion physique avec l'exterieur



# Choix des organes du système domotique

Nous avons connecté l'ENC28J60 avec la carte Arduino Méga 2560 (très puissante).



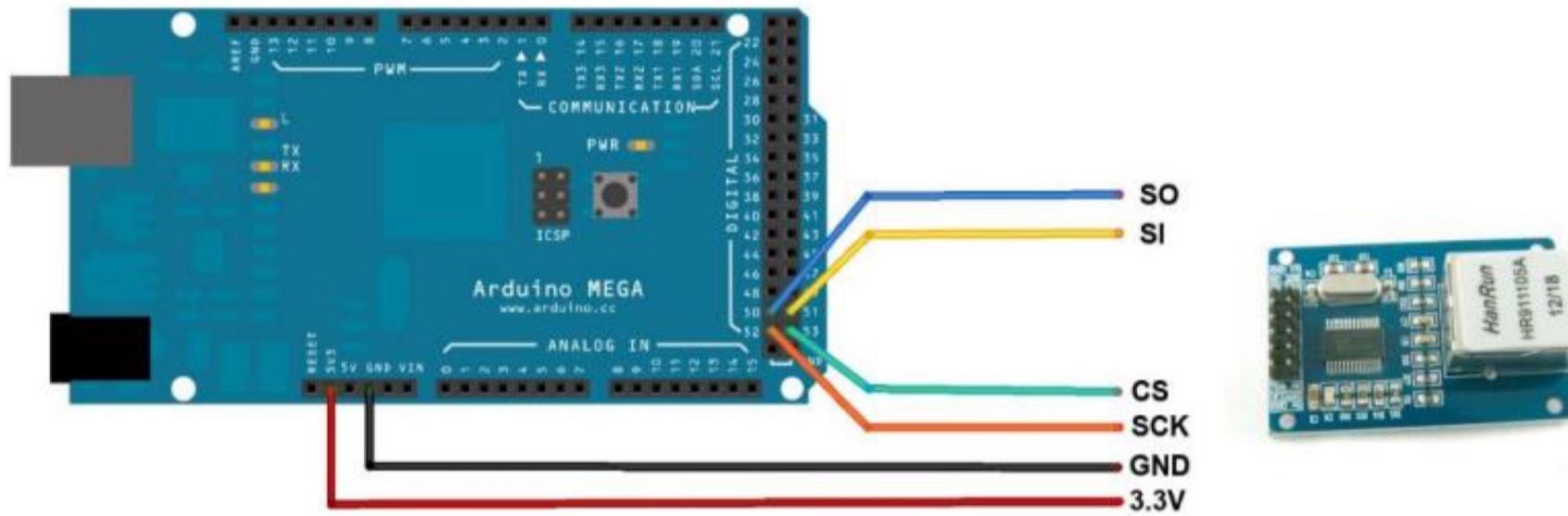
# Le module ENC28J60

Le module ENC28J60 permet de **connecter son Arduino à son réseau local**.

Il permet une intégration sensiblement facilité dans le réseau domestique en lui couplant un microcontrôleur afin de piloter les fonctionnalités Ethernet, **il reste la solution la moins chère pour la mise en œuvre d'un projet connecté au réseau Ethernet**.



# Le module ENC28J60



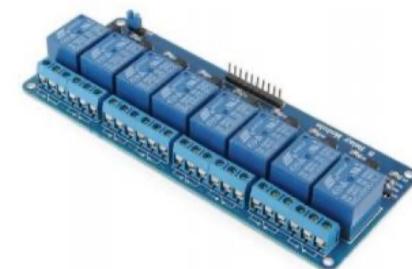
Bronchement avec la carte Arduino Méga 2560

# Module de Relais à 8 canaux

Il s'agit d'une carte d'interface de relais, qui peuvent être contrôlé directement par un large éventail de microcontrôleurs comme Arduino, AVR, PIC, ARM, API, etc.

Le module de relais est largement utilisé pour tout contrôle MCU, le secteur industriel, contrôle PLC, contrôle de la maison intelligente.

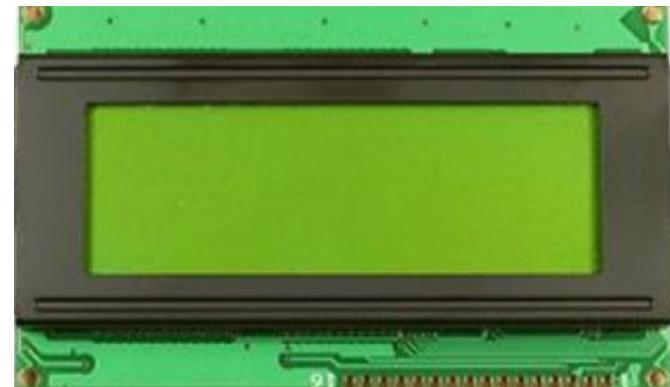
**Contrôler de l'éclairage de 4 pièces ainsi de la ventilation.**



## Module Afficheur LCD

Les afficheurs LCD alphanumériques présentent une solution facile d'emploi et bon marché de doter votre projet d'une interface indépendante de votre PC.

- Afficheur 2 lignes et 16 colonnes d'affichage
- Afficheur 4 lignes et 20 colonnes d'affichage.





## Module Afficheur LCD

Le LCD dispose de 3 registres internes:

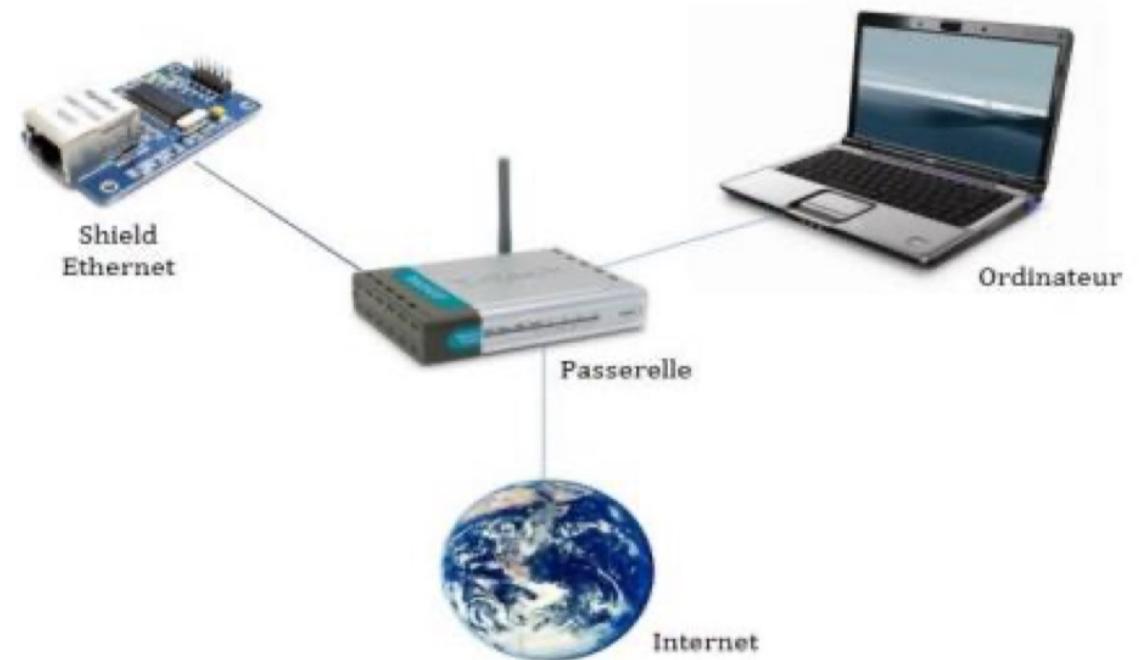
**Le registre de données** :l'envoi des codes des caractères à afficher.

**Le registre de commande** :envoyer des commandes d'effacement de l'écran, de positionnement du curseur, etc.

**le registre d'état** qui permet de consulter notamment la disponibilité du LCD pour recevoir des commandes ou des données.

## Point d'accès (Passerelle)

Pour se connecter au réseau local du système domotique via le **module Ethernet ENC28J60**, **Passerelle** : l'utilisateur puisse accéder à l'application de commande (application HTML).





## Capteur d'humidité et température DHT11

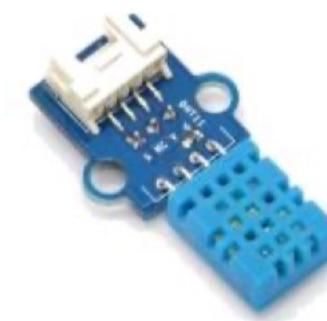
Un sous-système d'acquisition de température avec le capteur de température DHT11.

Cette température acquise va être renvoyée vers l'utilisateur sur l'application sous Smart Phone ou son ordinateur d'une manière automatique.

# Composants Capteur d'humidité et température DHT11

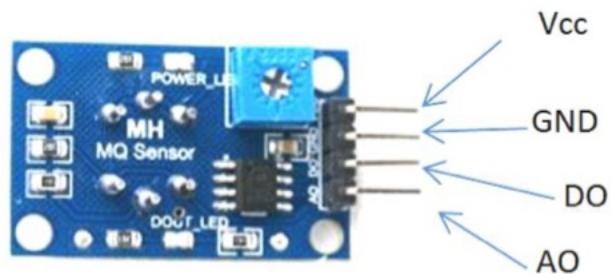
Ce capteur d'humidité et de température est très rependu dans le contrôle de climatisation, il est constitué :

- ✓ capteur de température à base de NTC
- ✓ capteur d'humidité résistif
- ✓ un microcontrôleur s'occupe de faire les mesures, les convertir et de les transmettre.



# Capteur de gaz/fumée (MQ-2)

Le capteur de gaz inflammable et de fumée MQ2 détecte la concentration des gaz combustibles dans l'air et renvoie sa lecture comme tension analogique.



# Bronchement de capteur de gaz/fumée MQ-2 avec Arduino



## Ventilateur

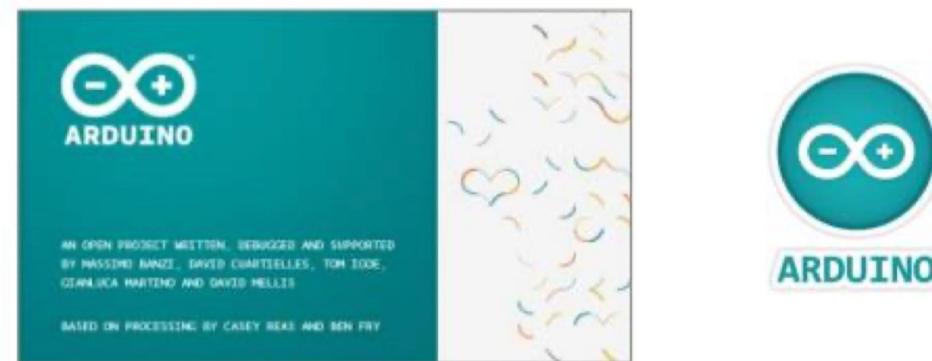
Programmer un degré de température adapté à chaque pièce et notamment en fonction de moments de la journée, nuit et jour.

Ces températures par zone sont contrôlées depuis l'application HTML conçue pour donner de la fraîcheur à l'intérieur de l'habitat.



# Etude de la partie logicielle

L'interface de l'IDE Arduino est plutôt simple, il offre une interface minimale et épurée pour développer un programme sur les cartes Arduino.





# Langage Arduino

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le C, le C++, le Java et le Processing. Le langage impose une structure particulière typique de l'informatique embarquée.



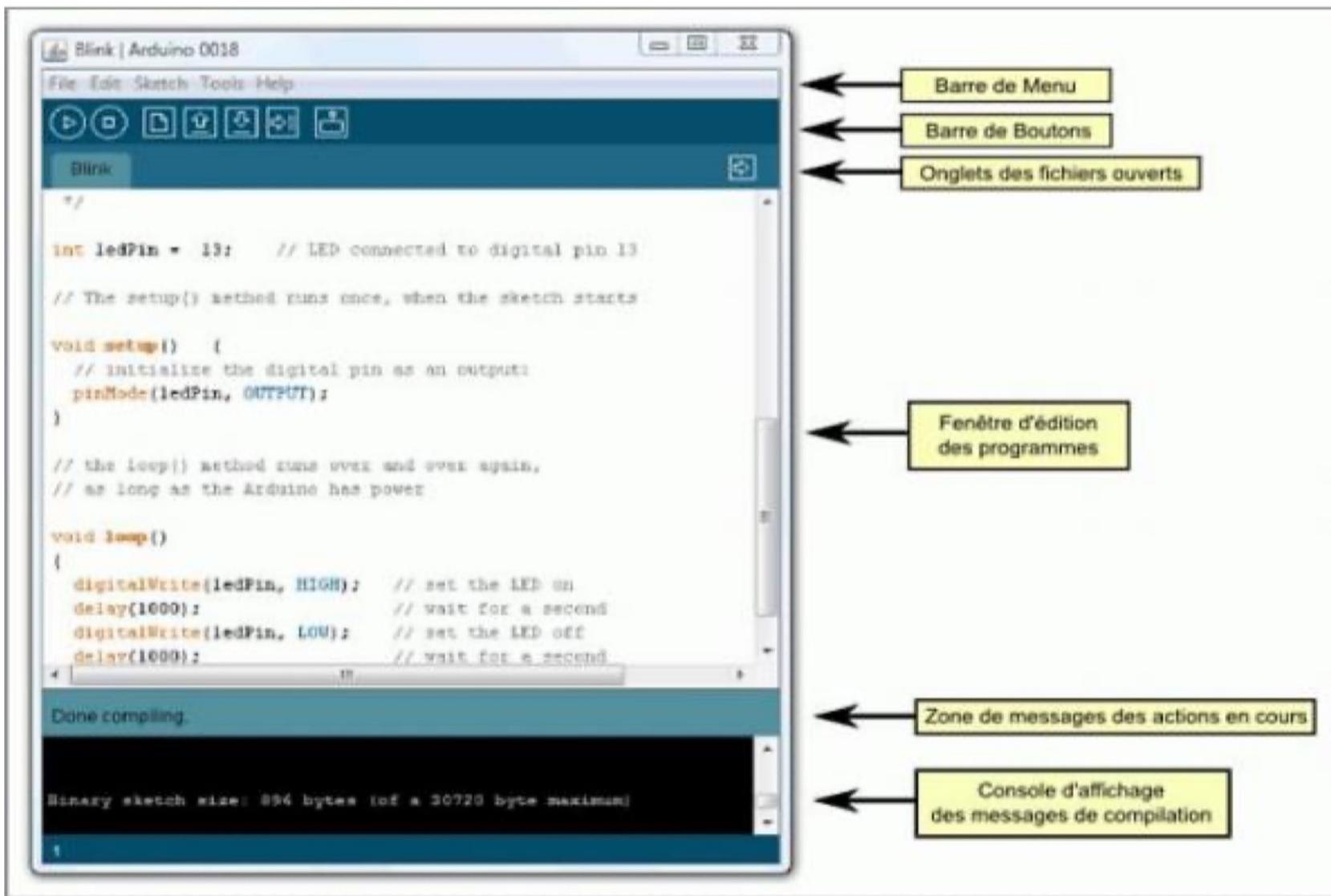
# Langage Arduino

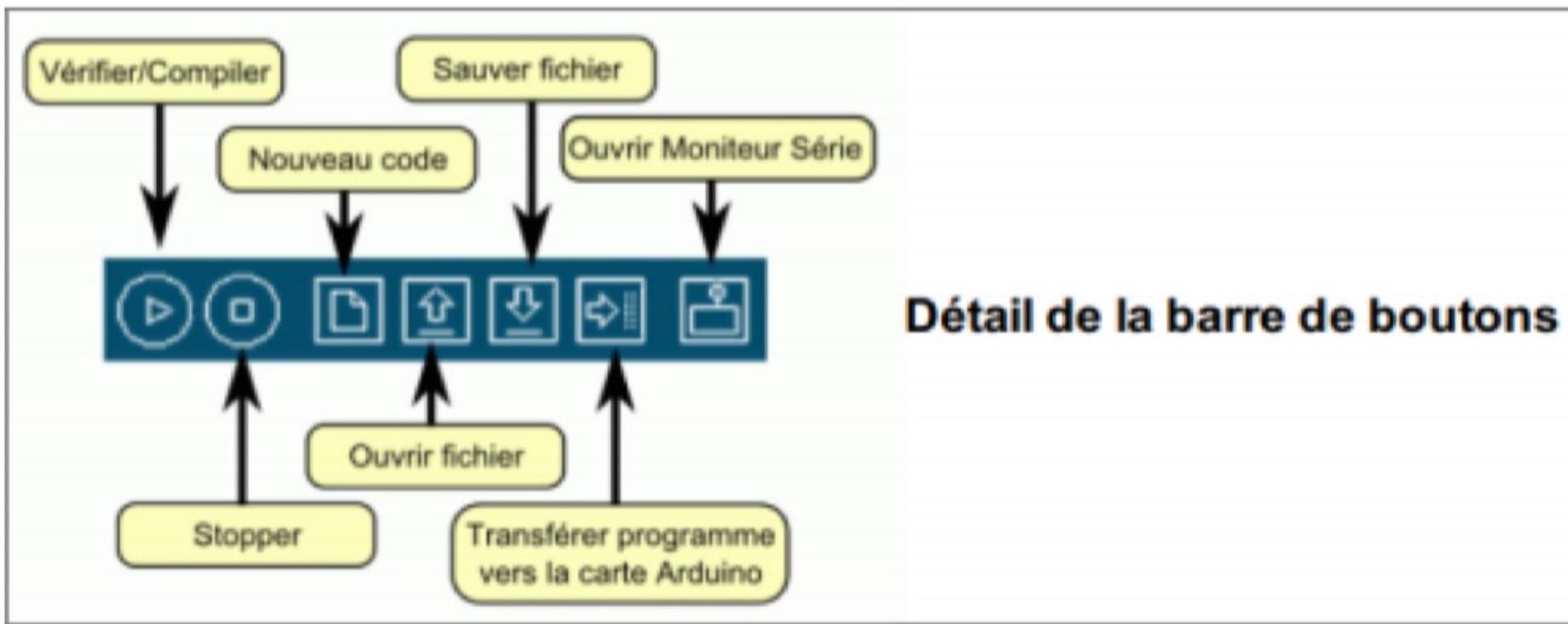
## **La fonction « setup »:**

toutes les opérations nécessaires à la configuration de la carte  
(directions des entrées sorties, débits de communications série, etc.).

## **La fonction « loop » :**

exécutée en boucle après l'exécution de la fonction setup. tant que la carte n'est pas mise hors tension, redémarrée (par le bouton reset)..





Commentaires multilignes pour se souvenir du patch ==>

```

sketch_061111a §
/*
 * Ce programme fait clignoter une LED branchée sur la broche 13
 * et fait également clignoter la diode de test de la carte
 */

```

I/La définition des constantes et des variables	<code>int ledPin = 13;</code>	// LED connectée à la broche 13
2/La configuration des entrées et sorties <code>void setup()</code>	<code>void setup() {     pinMode(ledPin, OUTPUT); }</code>	// configure ledPin comme une sortie
3/La programmation des interactions et comportements <code>void loop()</code>	<code>void loop() {     digitalWrite(ledPin, HIGH);     delay(3000);     digitalWrite(ledPin, LOW);     delay(1000); }</code>	// met la sortie à l'état haut (led allumée) // attente de 3 secondes // met la sortie à l'état bas (led éteinte) // attente de 1 seconde

Une fois la dernière ligne exécutée, la carte revient au début de la troisième phase et recommence sa lecture et son exécution des instructions successives. Et ainsi de suite.

Cette **boucle** se déroule des milliers de fois par seconde et anime la carte.

Done compiling.



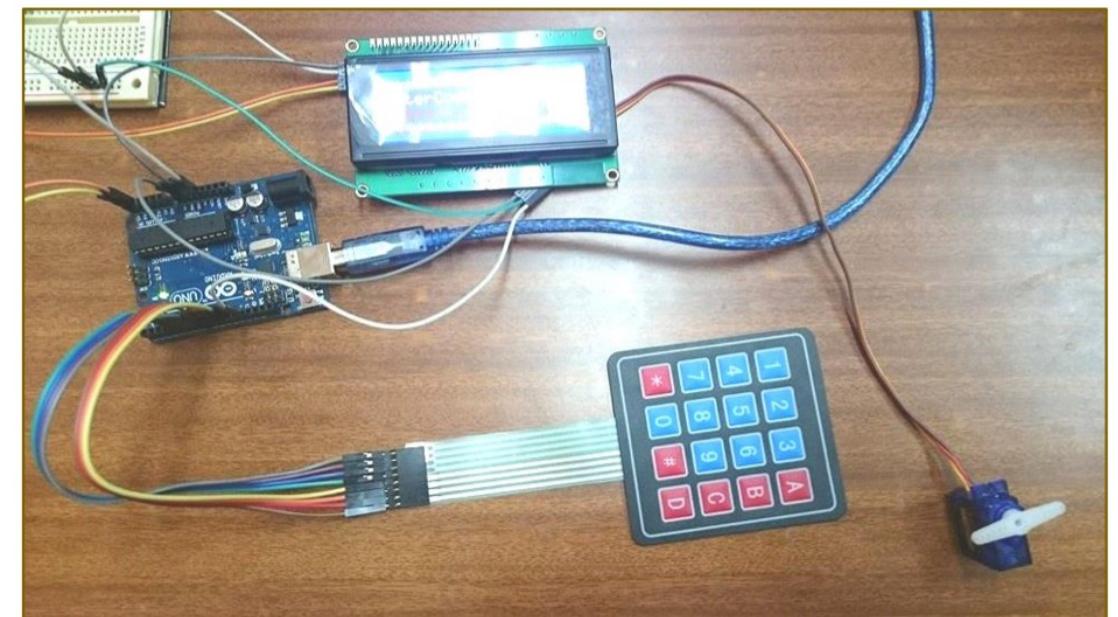
## Realisation de la maison intelligente

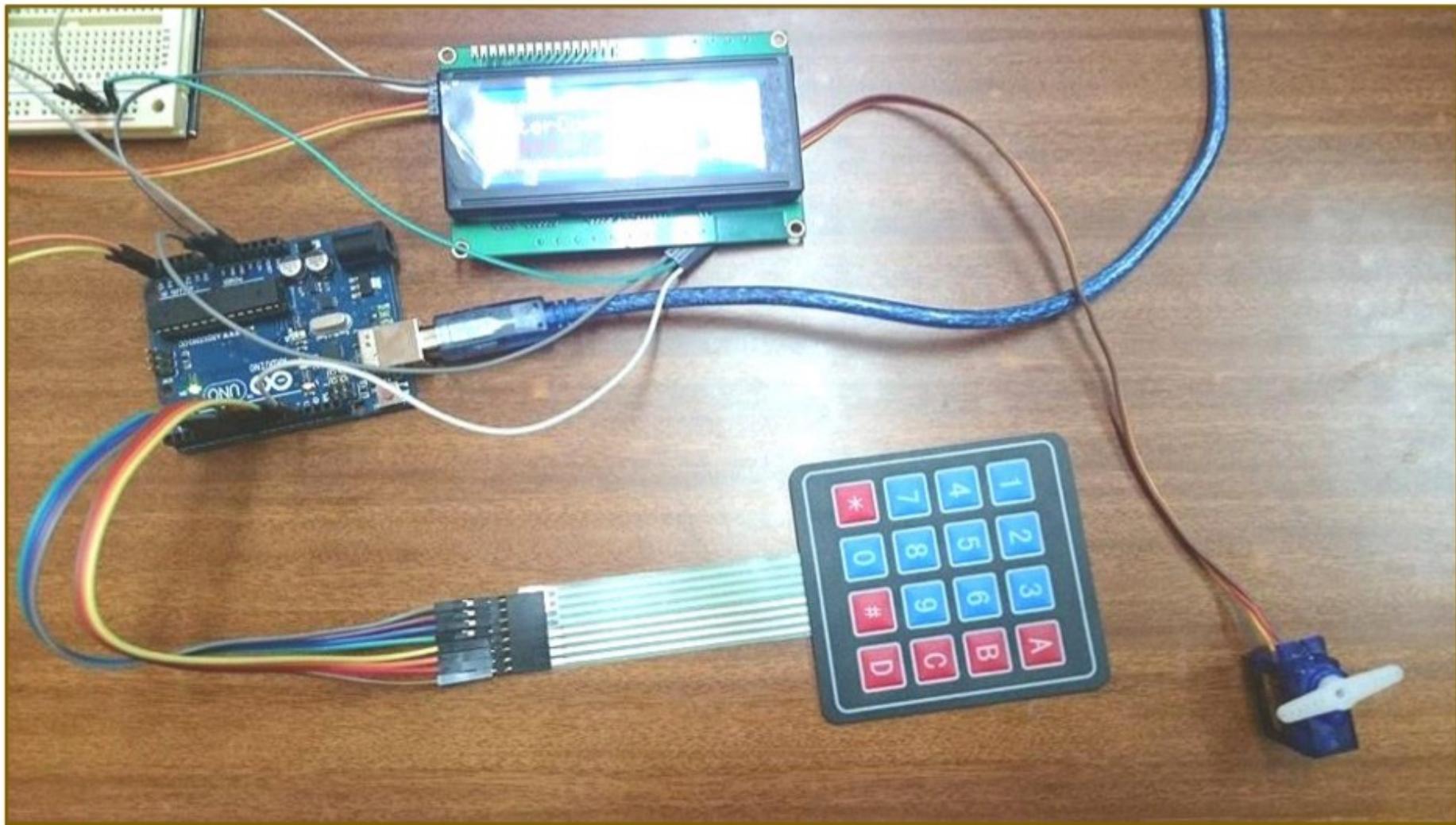
Nous avons concevoir le dessin d'ensemble de notre maison intelligente sur Catia V5 avant de passer à l'étape de la réalisation sur Terrain :

## Realisation de la maison intelligente

### Fonction d'accès sécurisé à l'habitat

Cette fonction sera indépendante des autres fonctions qui seront commandées à distance par l'intermédiaire de l'application HTML.





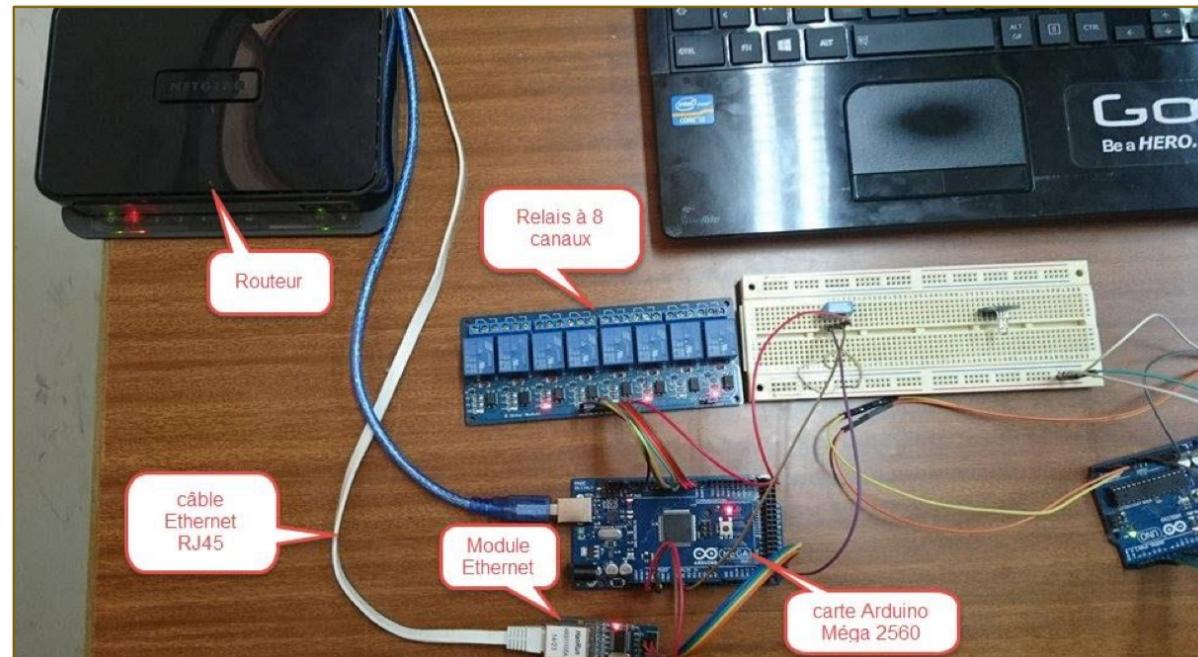


## Fonction d'ouverture du garage

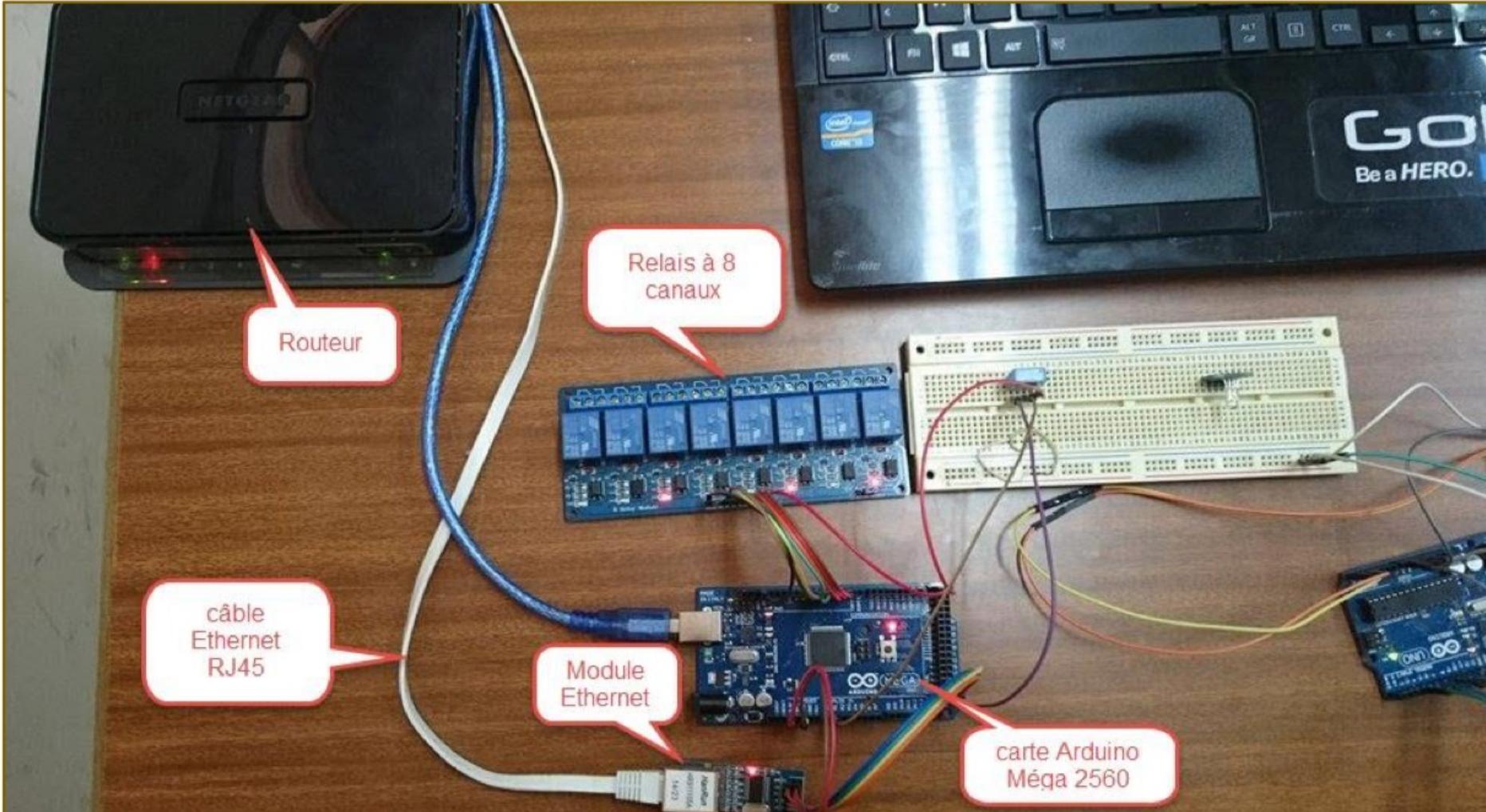
La commande d'ouverture du portail du garage est réalisée à distance via l'application HTML de commande en agissant sur le contrôle du servomoteur pour faire monter/descendre le volet du garage. (voir l'application de commande dans les annexes).

## Fonction d'éclairage

La fonction d'éclairage est assurée via l'application HTML qui va être commandée à travers le module Ethernet branché sur la carte Arduino Méga et lié au routeur qui va créer un réseau local entre le PC (ou Smart phone) et les lampes de l'éclairage.

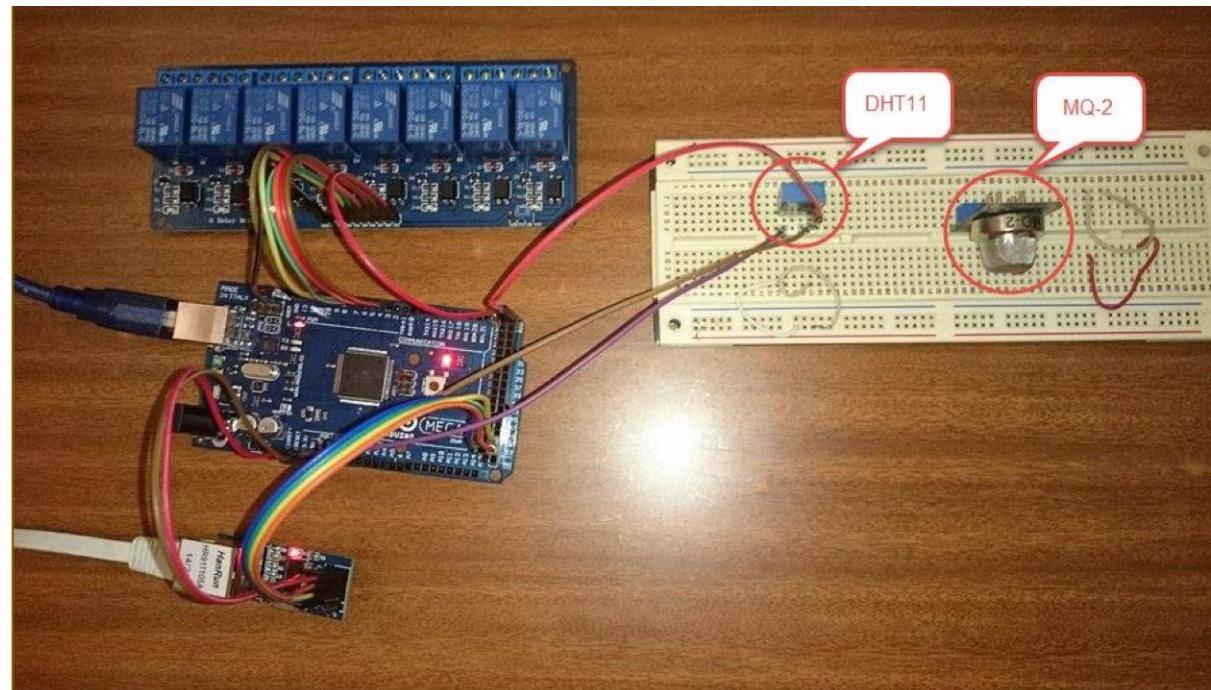


## Fonction d'éclairage



## Fonction de l'acquisition de la température

La fonction de l'acquisition de la température est réalisée via le capteur DHT11 par la suite les valeurs des degrés seront affichées sur la page HTML.





## Fonction de ventilation

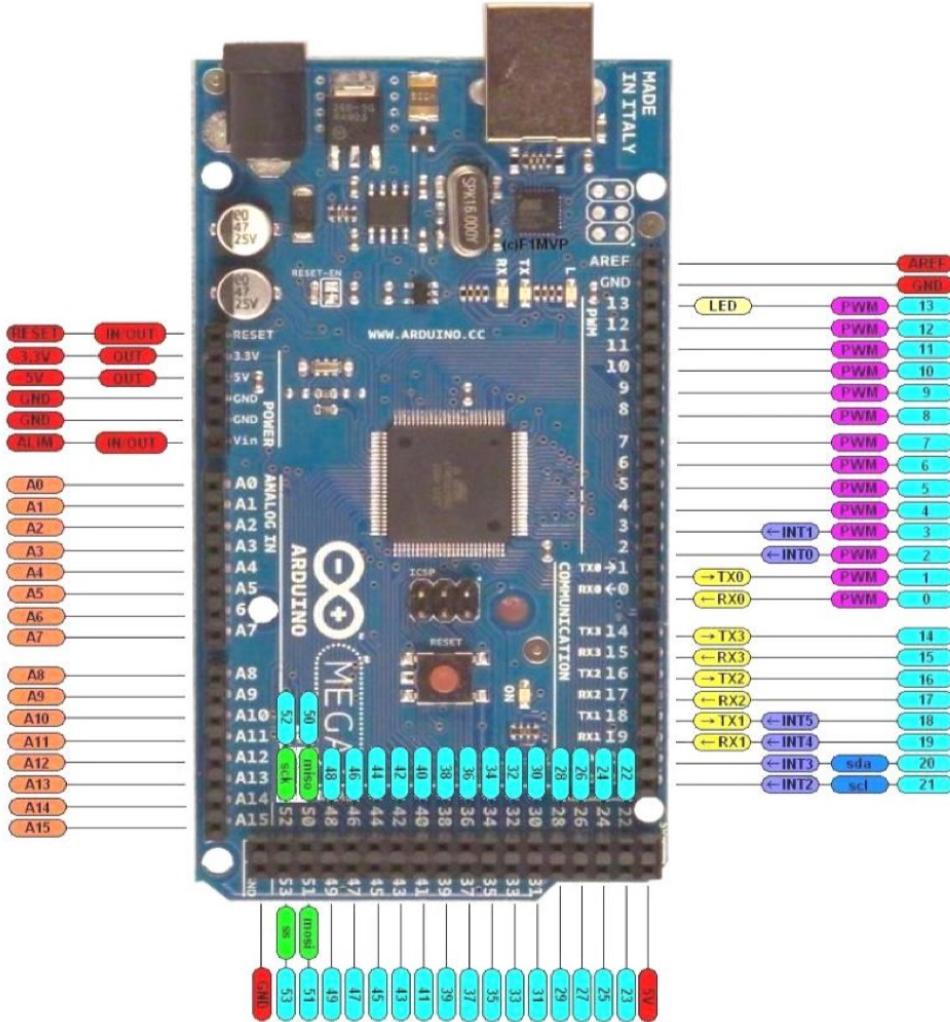
La ventilation de l'intérieur de l'habitat est assurée par l'intermédiaire de l'application HTML ou automatiquement vis-à-vis la variation de la température (déclaration d'un seuil de température pour lancer la ventilation).



## Fonction de détection de gaz/fumée

Cette fonction permet de détecter s'il y a des fuites de gaz dans la cuisine à l'aide du capteur MQ-2 en affichant sur l'application de commande le degré de concentration de gaz. Par la suite, on peut lancer une alarme pour informer l'utilisateur en cas de danger !

# Fonction de détection de gaz/fumée





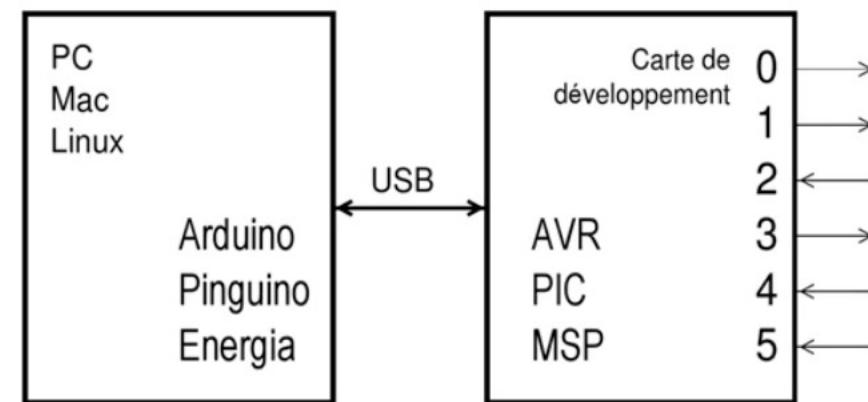
# *Programmation Microcontrôleurs Arduino*

# Arduino

Microcontroleur en open source le plus utilisé → Tourne sur différents systèmes d'exploitation (Mac, Linux..)

Il existe plusieurs constructeurs qui utilisent Arduino sous différents noms

- ❖ Pinguino
- ❖ Energia





# Arduino

L'intérêt de Arduino réside dans le fait de cacher la complexité du processeur.

- ✓ Programmer les pins de sortie et d'entrée.
- ✓ Courant va rentrer dans le processeur pour allumer la LED mais la pin3 va produire une sortie vis à vis du processeur.

---

---

```
pinMode (3,OUTPUT) ;    digitalWrite (3, HIGH) ;
pinMode (4,INPUT);     digitalRead (4)
```

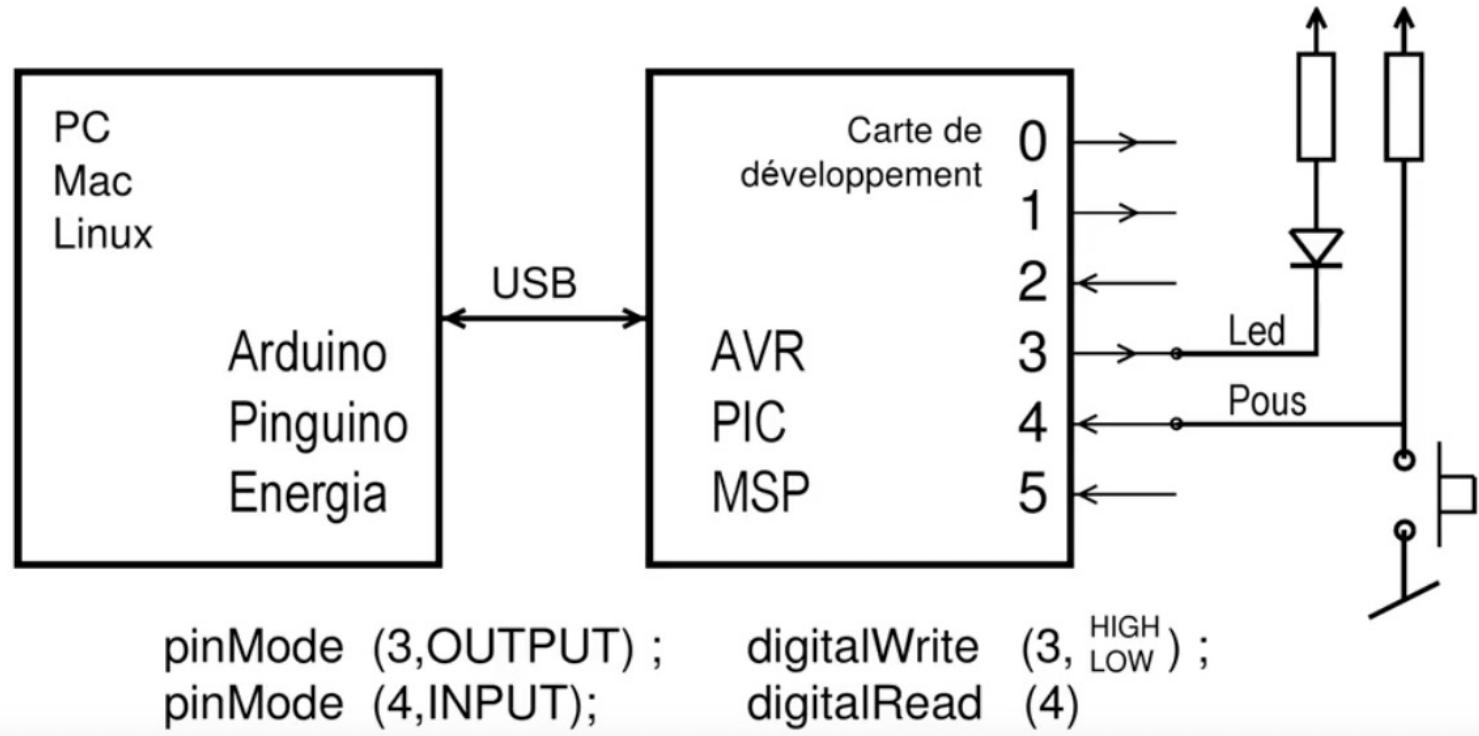
# Arduino

Programmer des pins en entrée soit en sortie

**Pin 3 : brancher une Led avec sa resistance.**

**→Ecrire HIGH=> Etat logique 1(Allumer La Led)**

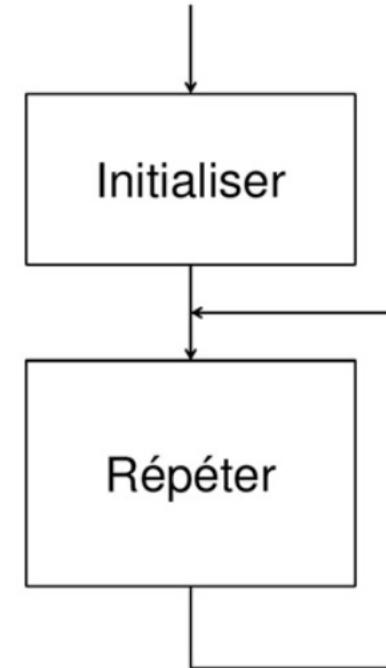
Pin4: Lire Poussoir



# Programme Arduino

- ✓ **Setup()**: fonction qui permet d'initialiser
- ✓ **Loop()**: fonction qui s'exécute en permanence

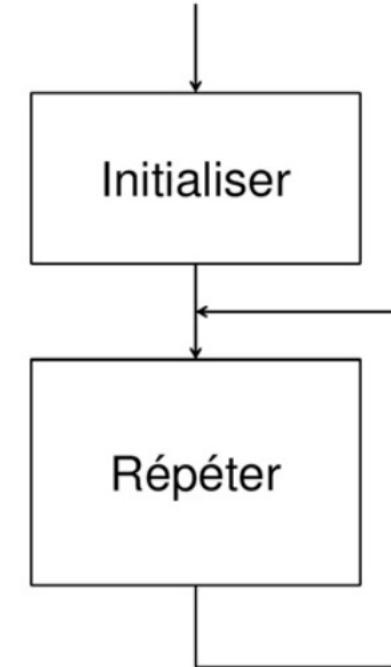
```
void setup () {  
    pinMode (3,OUTPUT) ;  
}  
  
void loop () {  
    digitalWrite (3,LOW) ;  
    delay (1000) ;  
    digitalWrite (3,HIGH) ;  
    delay (1000) ;  
}
```



# Etapes d'exécution d'un programme Arduino

- ✓ **La phase d'initialisation** avec La fonction Setup()  
**La boucle loop()** pour exécuter ce qu'on lui de faire

```
void setup () {  
    pinMode (3,OUTPUT) ;  
}  
  
void loop () {  
    digitalWrite (3,LOW) ;  
    delay (1000) ;  
    digitalWrite (3,HIGH) ;  
    delay (1000) ;  
}
```





# Programme Arduino

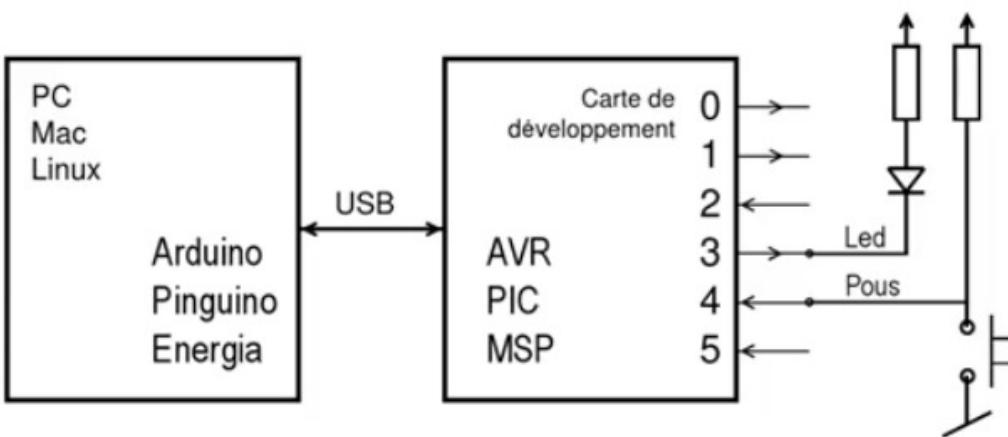
```
#define Led 3
```

```
#define LedOn digitalWrite (Led,LOW)  
#define LedOff digitalWrite (Led,HIGH)
```

```
void setup () {  
    pinMode (Led,OUTPUT) ;  
}
```

```
void loop () {  
    LedOn ;  
    delay (1000) ;  
    LedOff ;  
    delay (1000) ;  
}
```

# Programme Arduino



```
pinMode (4, INPUT) ;
```

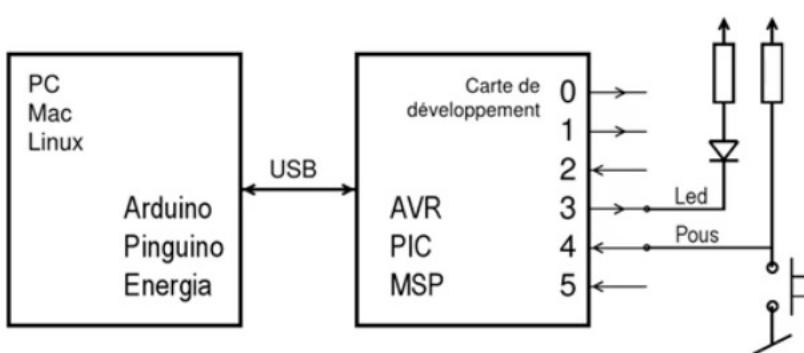
```
digitalRead (4)
```

= 1 si relâché  
= 0 si pressé

```
if ( )
```

```
if (digitalRead (4) == 0)  
{  
    faire ceci si c'est pressé  
}
```

# Programme Arduino



```
#define Pous 4
```

```
#define PousOn digitalRead (Pous) ==0
```

```
pinMode (Pous, INPUT) ;
```

```
digitalRead (Pous)
```

= 1 si relâché  
= 0 si pressé

```
if ( )
```

```
if (PousOn) {  
    faire ceci  
}
```



# Programme Arduino

Faire marcher le programme avec la carte Arduino, on branche la led sur le pin 3 et le poussoir sur le pin 4

## Question:

Si on enlève la structure 'else', que se passe-il?

- a) Cela ne change rien
- b) La Led ne s'allume plus
- c) La Led reste allumée

```
void loop () {  
    if (PousOn) {  
        LedOn ;  
    }  
    else {  
        LedOff ;  
    }  
}
```

# Programme Arduino

Faire marcher le programme avec la carte Arduino, on branche la led sur le pin 3 et le poussoir sur le pin 4

## Question:

Si on enlève seulement le 'else {}', que se passe-il?

- a) La Led clignote
- b) La Led ne s'allume plus
- c) La Led ne s'éteint plus

```
void loop () {  
    if (PousOn) {  
        LedOn ;  
    }  
    else {  
        LedOff ;  
    }  
}
```



# Definir la fonction PoussOff

```
// Allume la Led si on n'agit pas sur le poussoir  
  
#define Pous 4  
#define PousOn digitalRead (Pous) == 0  
#define PousOff digitalRead (Pous) == 1  
  
#define Led 3  
#define LedOn digitalWrite (Led,LOW) ;  
#define LedOff digitalWrite (Led,HIGH) ;  
  
void setup () {  
    pinMode (Pous, INPUT) ;  
    pinMode (Led,OUTPUT) ;  
}
```

```
void loop () {  
    if (PousOff) {  
        LedOn ;  
    }  
    else {  
        LedOff ;  
    }  
}
```

```
void loop () {  
    if ( ! PousOn) {  
        LedOn ;  
    }  
    ...  
}
```

! pour inverser  
l'état 0-1, la  
condition vrai/faux



# L'instruction While

```
void loop () {  
    while (PousOn) {  
        LedOn ;  
    }  
    LedOff ;  
}
```

while (condition)  
{  
 faire tant que la  
 condition est vraie  
}

while (1) {  
 faire éternellement  
}

while (1) {}  
 ne plus rien faire



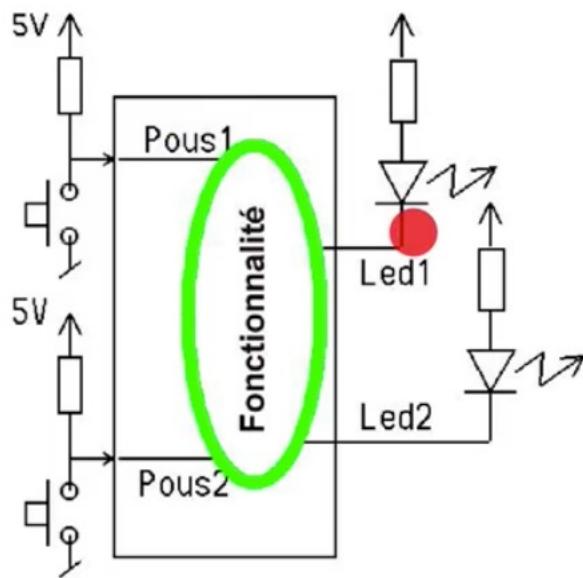
# Inverser l'état de la LED

Inverser l'état de la Led (toggle)

```
if (digitalRead(Led) == 0)  {  
    LedOn  
}  
else  {  
    LedOff  
}
```

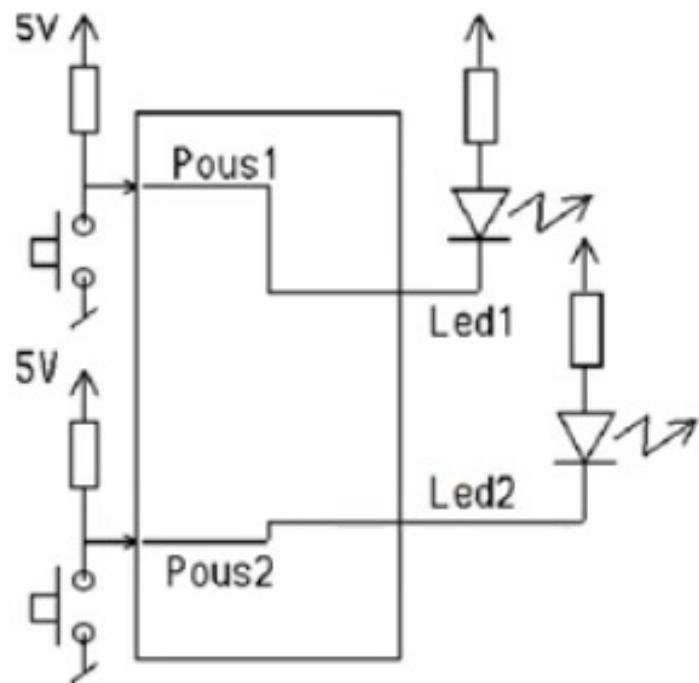
```
#define EtatLed digitalRead(Led)  
    if (EtatLed==0) {LedOn}  
ou  
#define LedEstEteinte digitalRead(Led) ==0  
    if (LedEstEteinte) {LedOn}
```

# Exemple : 2 poussoirs et 2 LEDs



```
#define Led1 5 // actif à 0
#define Led2 6 // actif à 0
#define Pous1 2 // actif à 0
#define Pous2 3 // actif à 0
#define Pous1On !digitalRead (Pous1) // actif à 0
#define Pous2On !digitalRead (Pous2) // actif à 0
#define Led1On digitalWrite (Led1, LOW)
#define Led1Off digitalWrite (Led1, HIGH)
#define Led2On digitalWrite (Led2, LOW)
#define Led2Off digitalWrite (Led2, HIGH)
```

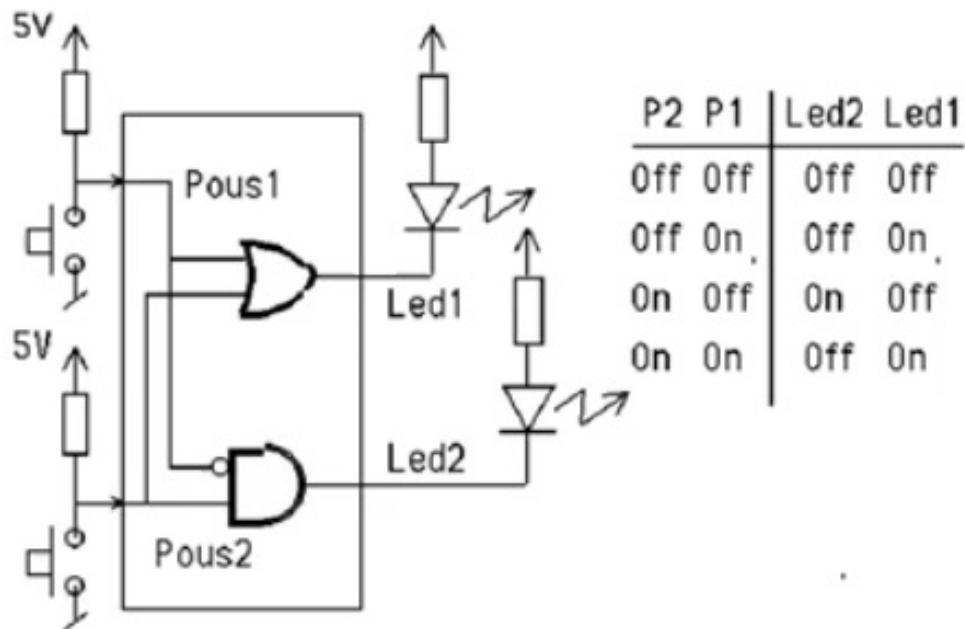
# Exemple : 2 poussoirs et 2 LEDs



On copie le poussoir 1 sur la LED1 et  
on copie le poussor 2 sur la LED2

```
void loop() {
    if (Pous1On) { Led1On; }
    else { Led1Off; }
    if (Pous2On) { Led2On; }
    else { Led2Off; }
}
```

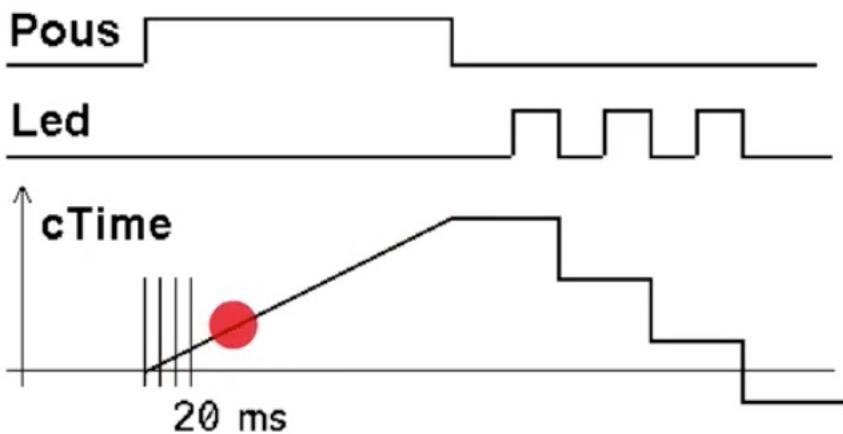
# Exemple : 2 poussoirs et 2 LEDs



```
void loop() {
    if (Pous1On || Pous2On) { Led1On; }
    else { Led1Off; }
    if (Pous2On && !Pous1On) { Led2On; }
    else { Led2Off; }
}
```

# Exemple : 2 poussoirs et 2 LEDs

Nombre de clignotements  
selon durée poussoir

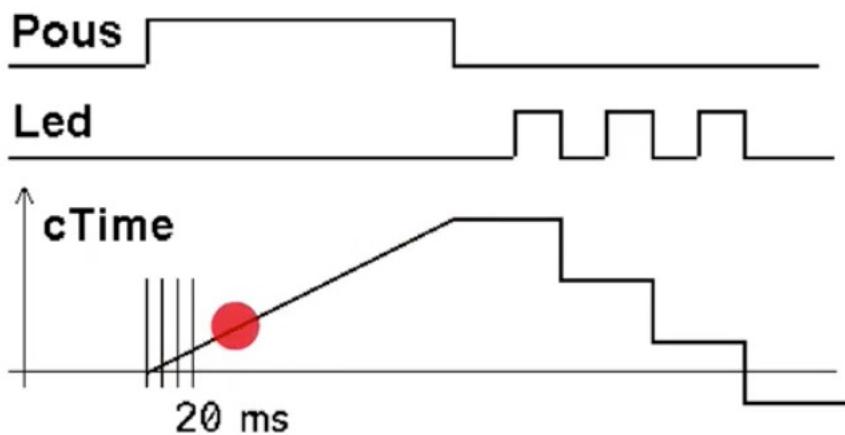


Tant que je presse un bouton, je  
vais incrementer un compteur,  
augmenter toutes les 20  
miliisecondes.

Une fois, nous avons relâché le  
bouton, nous avons décompté

# Exemple : 2 poussoirs et 2 LEDs

Nombre de clignotements  
selon durée poussoir



Definir une variable compteur  
du temps **cTime**

```
int cTime ;  
void loop() {  
    cTime = 0;  
    while (PousOn) {  
        cTime++;  
        delay (20);  
    }  
    while ( cTime >0) {  
        LedOn; delay (200);  
        LedOff; delay (200);  
        cTime -= 25;  
    }  
}
```



# *Chapitre IV : Mise en réseau et architectures de l'Internet des objets*



# Web et l'IOT

La mise en oeuvre d'un systeme IOT necessite:

L'exploitation de toutes les ressources materielles: les objets, les devices et logiciels comme les applications et les services offerts par les fournisseurs IOT.

Faire parler ces objets a travers un reseau avec differents types de connexion ( Modele OSI ou TCP/IP) **→ Interoperabilite( faire communiquer)**



# Le Web et le IOT

## Difference entre le Web et IOT

- ❖ Pour le web, nous avons **le modèle OSI ,TCP/IP** avec des couches
- ❖ Pour chaque couche nous avons un certain nombre de standards et de protocoles.
- ❖ Pour la couche réseau on a le **IPV4 et IPV6**
- ❖ Pour la couche transport, on a **TCP/IP**
- ❖ Protocole Ethernet, **3G, 4G**



# Le web et l'IOT

- ❖ La partie applicative , nous avons **http**
- ❖ Est ce qu 'il existe une architecture universelle IOT?
- ❖ Est ce au 'il ya la possibilite d'utiliser TCP/IP? (**NON**)

**Une capacite reseau enorme**

**un echange de message enorme qui n'existe pas forcement pour les modeles IOT.**



# Le web et l'IOT

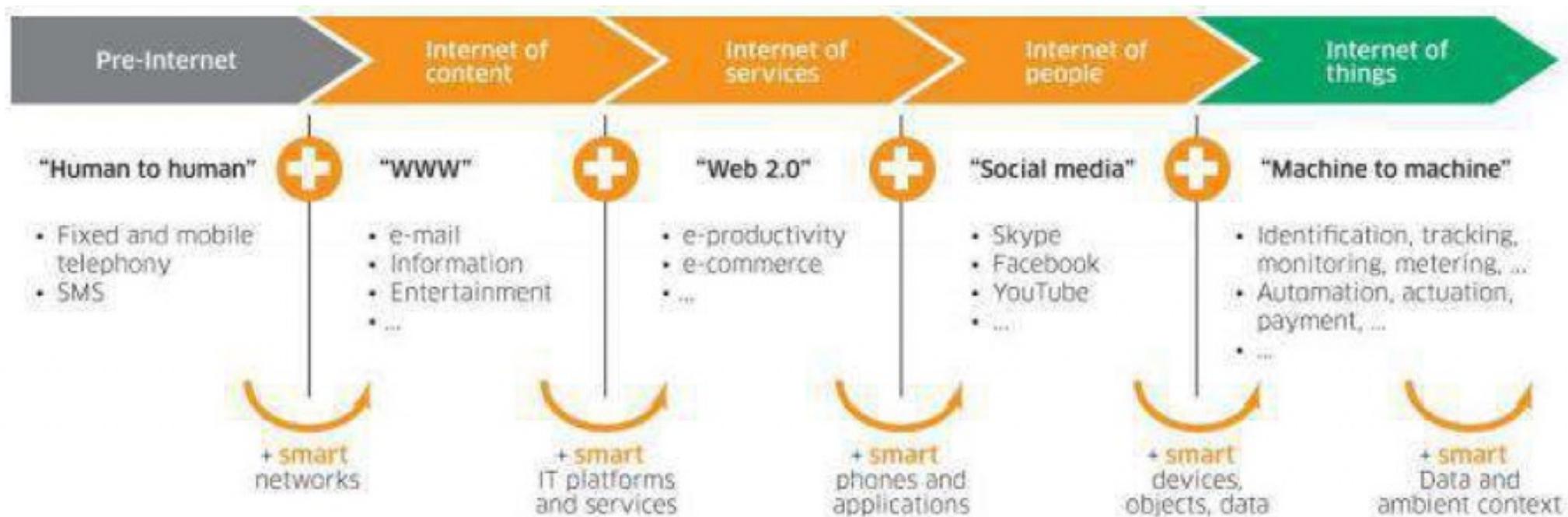
Simitude entre le web et le IOT.

- Couche dedie aux objets
- Couche dediee a la connectivite reseau.
- Couche dediee aux services, a applicatif.

**Multitude d'architecture IOT qui peuvent varier d'un constructeur a d'autres:**

- CISCO-AMAZON-IBM-GOOGLE (**Busniss specific architecture**).

# Evolution de l'internet



# Classes du reseaux

**1-Local Area Network ( LAN)**:small network = one local Area network

The way you access your network is through the wall socket and ethernet jack, routeurs Wifi.

**2- Wide Area Network (WAN) :** est plus complexe de Local Area Network (LAN), il existe une hiérarchie de machines qui communiquent entre elles. Il a différents protocoles de LAN.

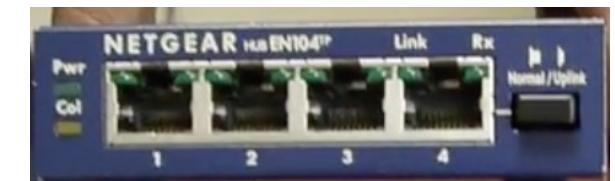
**3- Mobile Ad Hoc Networks MANET:** est compose de mobile devices, ad hoc network.

# Local Area network

Local Area Network contient des objets connectés et des ordinateurs qui communiquent entre eux avec un hub ou switch.



**Hub** : composant réseau avec plusieurs ports d'entrée.Une fois des données sont envoyées à un port elles seront copiées dans les autres.



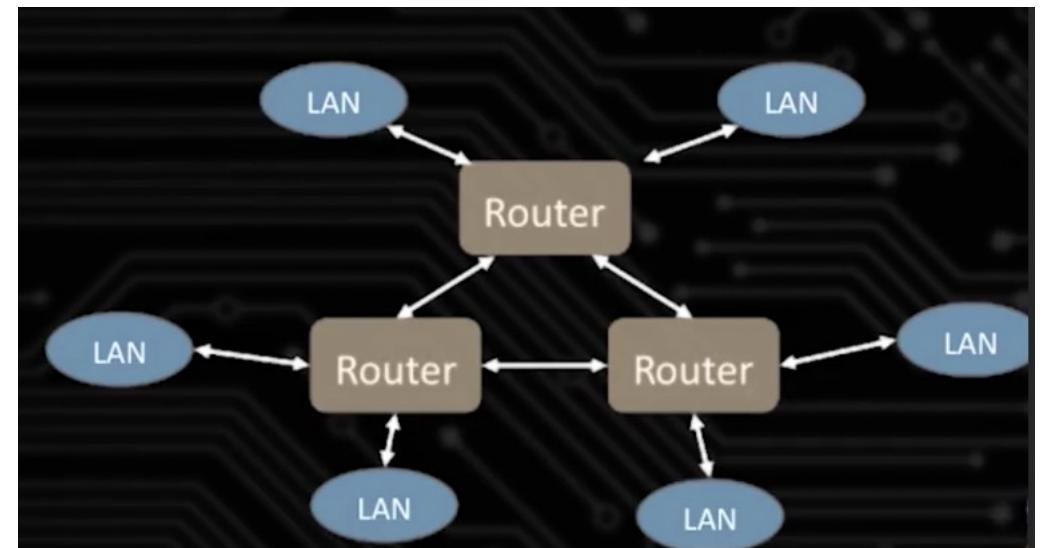
**Switch** : plus intelligent, il consulte l'entête du packet pour savoir sa destination (host3)



# Wide Area network

Wide area Networks sont composés de plusieurs LANs qui sont regroupes grâce a des routeurs

Les routeurs communiquent entre eux pour transmettre les données



# Modele OSI

Application

Presentation

Session

Transport

Networking

Datalink

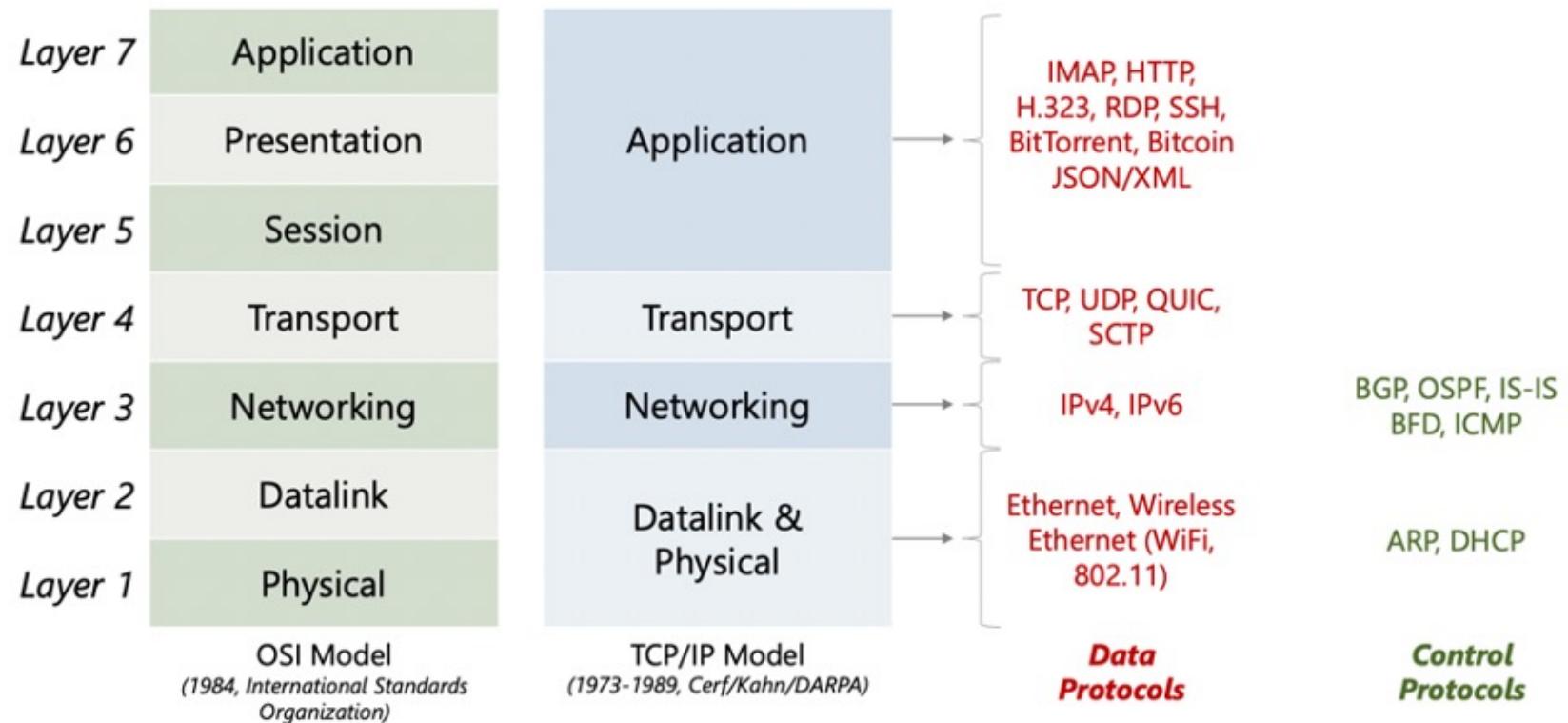
Physical

OSI Model  
(ISO/IEC 7498-1)

Définit de quelle manière les ordinateurs et les périphériques en réseau doivent procéder pour communiquer :

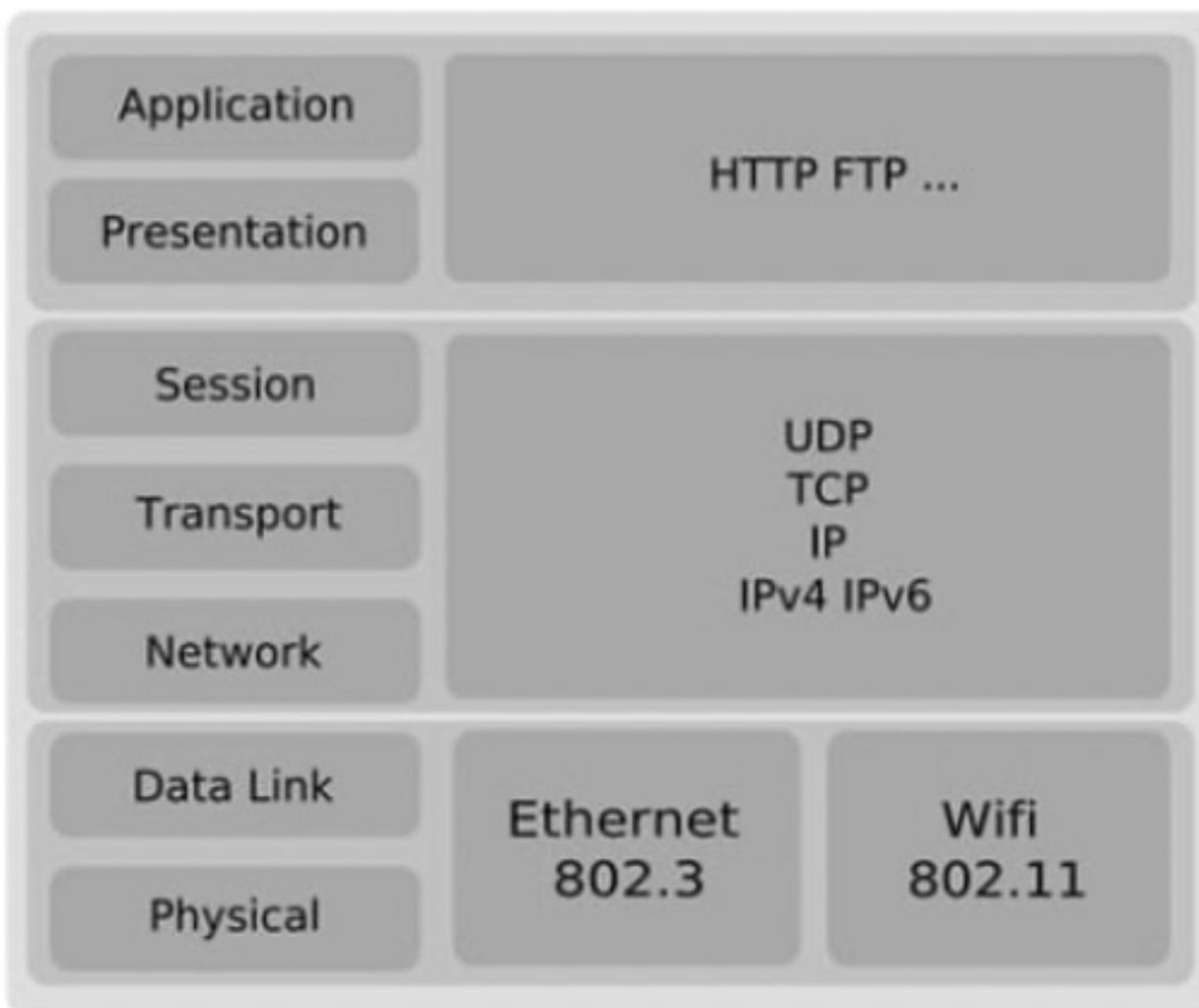
- ✓ Il spécifie le comportement d'un système dit ouvert
- ✓ Les règles de communication constituent les protocoles normalisés.

# Protocoles TCP/IP

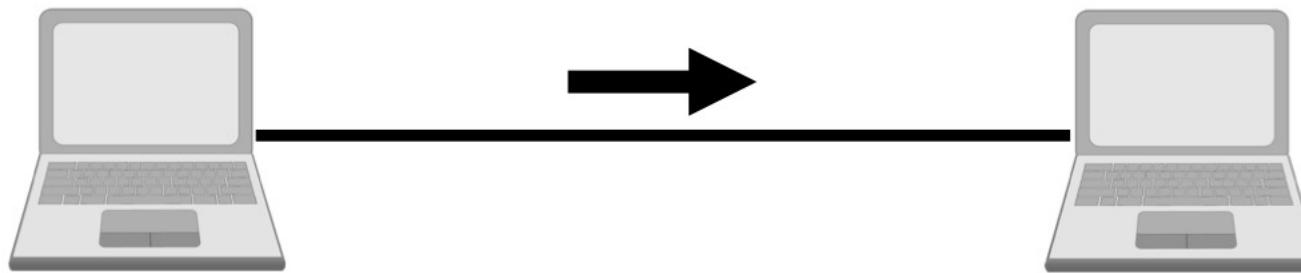
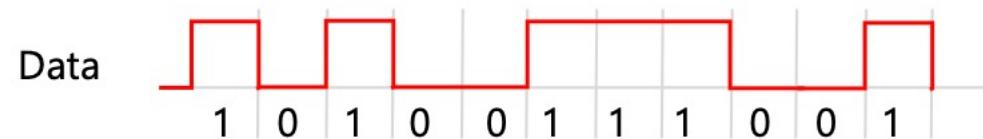




# Protocoles de communication de l'IOT



# Communication entre 2 machines





## Etude de protocole de communication sans Fil

**La communication sans fil** utilise les ondes électromagnétiques pour transmettre des données en utilisant l'air comme canal de transfert :



## Etude de protocole de communication sans Fil

- L'émetteur applique une certaine variation de courant à son antenne.
- La variation de courant induit une onde électromagnétique.
- L'onde électromagnétique se propage à une vitesse proche de celle de la lumière dans l'air.



## Etude de protocole de communication sans Fil

- Un courant électrique est induit dans l'antenne du récepteur par la variation de champs magnétique.
- Le récepteur lit la variation de courant et l'interprète selon le protocole de communication.

# ZigBee

Zigbee est un protocole de haut niveau (au même titre que le FTP, HTTP, etc.).

Il permet à de petites radios de communiquer sur de faibles distances.

Ce protocole est utilisé dans des radios à consommation réduite. Il est basé sur la norme IEEE.





# Bluetooth

La technologie Bluetooth dans le monde des télécommunications et dans les appareils sans fil.

Etablir la communication à partir d'un ordinateur portable, un Smart phone ou une Tablette.



# Étude des différents protocoles de communication

## Wifi

Le Wi-Fi est une technologie de réseau informatique sans fil mise en place pour fonctionner en réseau interne → norme IEEE 802.11 (ISO/CEI 8802 -11).





# Protocoles de communication de l'IOT



HaLow





# Protocoles réseaux couche application

**1. La couche application** est la couche la plus proche de l'utilisateur. La majorité des protocoles utilisés par les utilisateurs se situent dans cette couche (**HTTP, SMTP, FTP**).



# Protocoles réseaux couche application

## le transfert de fichiers:

Les protocoles FTP (IETF), NFS (Sun Microsystems) et AFS

## les services de messagerie

Les protocoles Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP)

## les sessions distantes

Les protocoles Telnet, rlogin, Secure Shell (SSH)

## le transfert de ressources hypermedia (les documents HTML)

Le protocole HTTP

## la résolution d'adresses.

Des protocoles d'exploitation et de gestion Domain Name System (DNS)

# Protocoles reseaux couche presentation

**La couche présentation** est chargée du formatage des données de la couche applicative afin qu'elles puissent être envoyées à travers le réseau puis être lues à nouveau par les applications.

**La couche session** contrôle les connexions entre les ordinateurs. Cette couche permet l'ouverture et la fermeture de session et gère la synchronisation des échanges ainsi que les transactions.



# Protocoles reseaux couche presentation

**La couche transport** fournit les moyens concrets pour transférer des données de taille variable d'une source vers une destination en conservant la qualité du service.

# Protocoles couche reseau

La couche réseau fournit les moyens concrets pour transférer des données de taille variable (appelés “paquets”) entre différents réseaux de nœuds.

Effectuer le routage et l’adressage des paquets dans cette couche  
Définir la route que vont emprunter les paquets pour aller d’un point de départ à un point d’arrivée (d’un interlocuteur à l’autre).

# Protocoles liaison des données

- Gère les communications entre deux machines directement connectées entre elles.
- Découper des données brutes en frames de tailles variables puis les envoyer de manière séquentielle.



# Protocole Ethernet

Ethernet est une technologie et protocole de connexion et de communication pour les réseaux locaux LAN et les réseaux étendus WAN de **la couche 1 du modèle OSI( couche physique)**.

Ethernet fonctionne dans les deux couches inférieures du modèle OSI: la couche liaison de données et la couche physique.

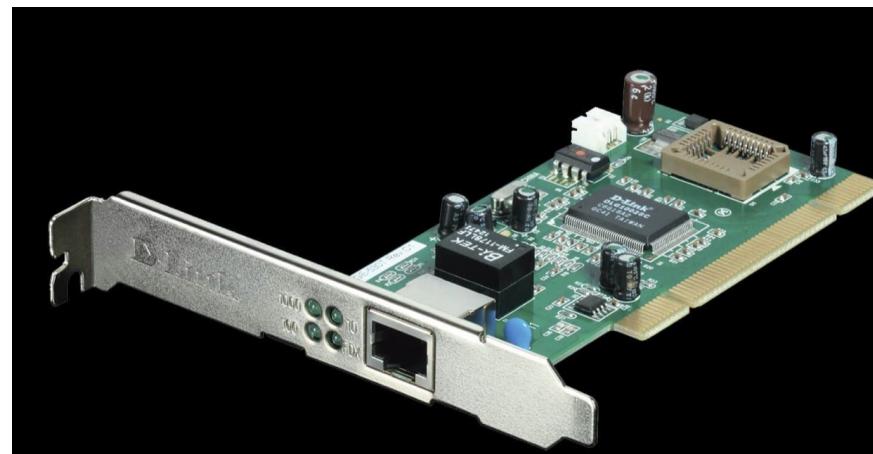


# Protocole Ethernet dans OSI

Ethernet Layer 1 joue un rôle clé dans la communication qui a lieu entre les périphériques, mais chacune de ses fonctions a des limites.

# Protocole Ethernet dans OSI

**Une carte Ethernet** de votre ordinateur remplit une fonction de base : transmettre des données du réseau à votre ordinateur





# Protocole Ethernet dans OSI

La carte réseau Ethernet possède un identifiant unique que l'on nomme **adresse MAC**.

Cette adresse est essentielle dans le fonctionnement du protocole Ethernet.



# Modèle TCP/IP

Le modèle TCP/IP est une approche réaliste ou pratique.

Le modèle OSI est un modèle idéalisé ou théorique.

→ le modèle TCP/IP qui est utilisé comme modèle de réseau de référence pour Internet.

Il contient 4 couches

- ❖ La couche application
- ❖ La couche transport
- ❖ Internet
- ❖ Accès réseau.



# Modèle TCP/IP

N° de la couche	OSI	TCP/IP	Unité de données
7	Application		
6	Présentation	Application	Données non transformées
5	Session		
4	Transport	Transport	Segments
3	Réseau	Internet	Paquets
2	Liaison des données	Accès réseau	Frames
1	Physique		Bits



# Couche application de TCP/IP

Le modèle TCP/IP regroupe les trois couches de session, présentation et application du modèle OSI dans une seule couche application.

les protocoles de haut niveau :

- ✓ FTP pour le transfert de fichiers.
- ✓ SMTP pour les mails, HTTP pour le WWW.
- ✓ DNS pour les noms de domaine.



# Couche transport

La couche transport assure la communication logique entre processus. Cette couche détermine comment les données doivent être envoyées : de manière fiable ou pas.

Protocoles dans la couche transport :

- ✓ TCP (Transmission Control Protocol)
- ✓ UDP (User Datagram Protocol).



# Couche transport

**TCP** est un protocole de transfert fiable orienté connexion pour s'assurer qu'il n'y ait ni perte ni corruption de données.

**UDP** est un protocole de transfert non fiable et qui ne nécessite pas de connexion préalable.

les échanges où la perte de quelques données n'est pas grave (appel vidéo, jeu en ligne, etc.)



# Couche internet

Le but principal de la couche Internet est d'assurer la communication logique entre hôte, c'est-à-dire de transmettre coûte que coûte les paquets d'un hôte à un autre et de faire en sorte qu'ils arrivent à destination.

Le protocole principal de cette couche est IP (Internet Protocol ou Protocole Internet).



# La couche accès réseau

La couche accès réseau du modèle TCP/IP regroupe les couches physique et de liaison des données du modèle OSI. Cette couche définit comment envoyer des paquets IP à travers le réseau (via des protocoles comme Ethernet ou Wireless entre autres)

# IOT et les protocoles de communication

Pour connecter un objet à Internet, plusieurs choix s'offrent à vous.

## Exemple

Wifi, le Bluetooth, le réseau cellulaire (3G, 4G).

Des technologies comme LPWAN **Low Power Wide Area Network** ont été développées spécifiquement pour connecter les objets connectés.

- minimiser la consommation énergétique
- maximiser la portée
- s'adapter au volume de données échangées.

## Les éléments fondamentaux d'un réseau IoT

Sa consommation d'énergie (combien d'énergie consommée pour envoyer 1 Mo).

Sa bande passante (combien de temps nécessaire pour envoyer 1 Mo).

Sa portée (sur combien de mètres ou kilomètres la connexion reste fiable).

A ces critères s'ajoute la fréquence à laquelle vous devez récupérer les données captées par votre objet (en temps réel, une fois par heure ou par jour).



## Les types de réseaux par topologie

**WAN (Wide Area Network)** : un réseau de plusieurs dizaines de kilomètres carrés

**LPWAN (Low Power Wide Area Network)** :

réseau de plusieurs dizaines de kilomètres carrés mais utilisant peu d'énergie (car peu de bande passante)

**PAN (Personal Area Network)** : le réseau de quelques mètres (Bluetooth)

**LAN (Local Area Network)**: le réseau Internet privé de votre domicile ou de votre entreprise (Wifi)

**Satellite** : partout dans le monde pour peu de ne pas être dans un tunnel (GPS)



# Reseau LPWAN

LPWAN (Low Power Wide Area Network) : un réseau à longue portée et à faible consommation énergétique

Réseau dédié à l'IOT et M2M

- Portée : de quelques kilomètres dans les zones les plus denses à des dizaines de kilomètres dans les zones rurales.
- Autonomie : Plusieurs années.  
Bande passante : De 0.1 à plusieurs centaines de kbits/sec.

## Catégories

LPWANs cellulaires : Nb-IoT et LTE-M

LPWANs non cellulaires : sigfox et LoRaWAN



# Reseau LPWAN

4 principaux paramètres pour choisir un réseau LPWAN

- ✓ Portée
- ✓ Débit
- ✓ Consommation d'énergie
- ✓ Coût de déploiement



# Reseau sigfox

Mis en place par la startup française au début des années 2010

- Repose sur une technologie brevetée de bande ultra-étroite UNB (Ultra Narrow Band)
  - Utilise des fréquences sans licence ISM (Industriel, Scientifique et Médical)



# Reseau LoraWAN

- Créé par la startup française Cycleo en 2009
  - Depuis 2015, le LoRaWAN est porté par une association à but non lucratif appelé The LoRa Alliance, qui regroupe plus de 500 entreprises
  - LoRa, comme le SigFox, opère sur une plage de fréquences ISM sans licence



# Reseaux capteurs sans Fil

## Technologie des réseaux sans-fil

- ✓ Les réseaux pour téléphones mobiles
- ✓ Les réseaux Ad-hoc.
- ✓ Les réseaux de capteurs sans-fil: échangent l'information sur l'environnement afin d'établir une vue globale de la région surveillée ( routage et l'acheminement des données)

# Réseaux capteurs sans Fil

- ✓ Les réseaux de capteurs sont le fruit des dernières avancées technologiques dans les domaines des réseaux sans-fil.
- ✓ concevoir des composants, à dimension réduite, intégrant un dispositif de captage et pouvant communiquer sur de faibles distances via des liaisons sans-fil.
- ✓ La mise en réseau de ces composants, appelés micro-capteurs, permet de réagir à des événements et d'analyser les données captées sur des zones étendues.

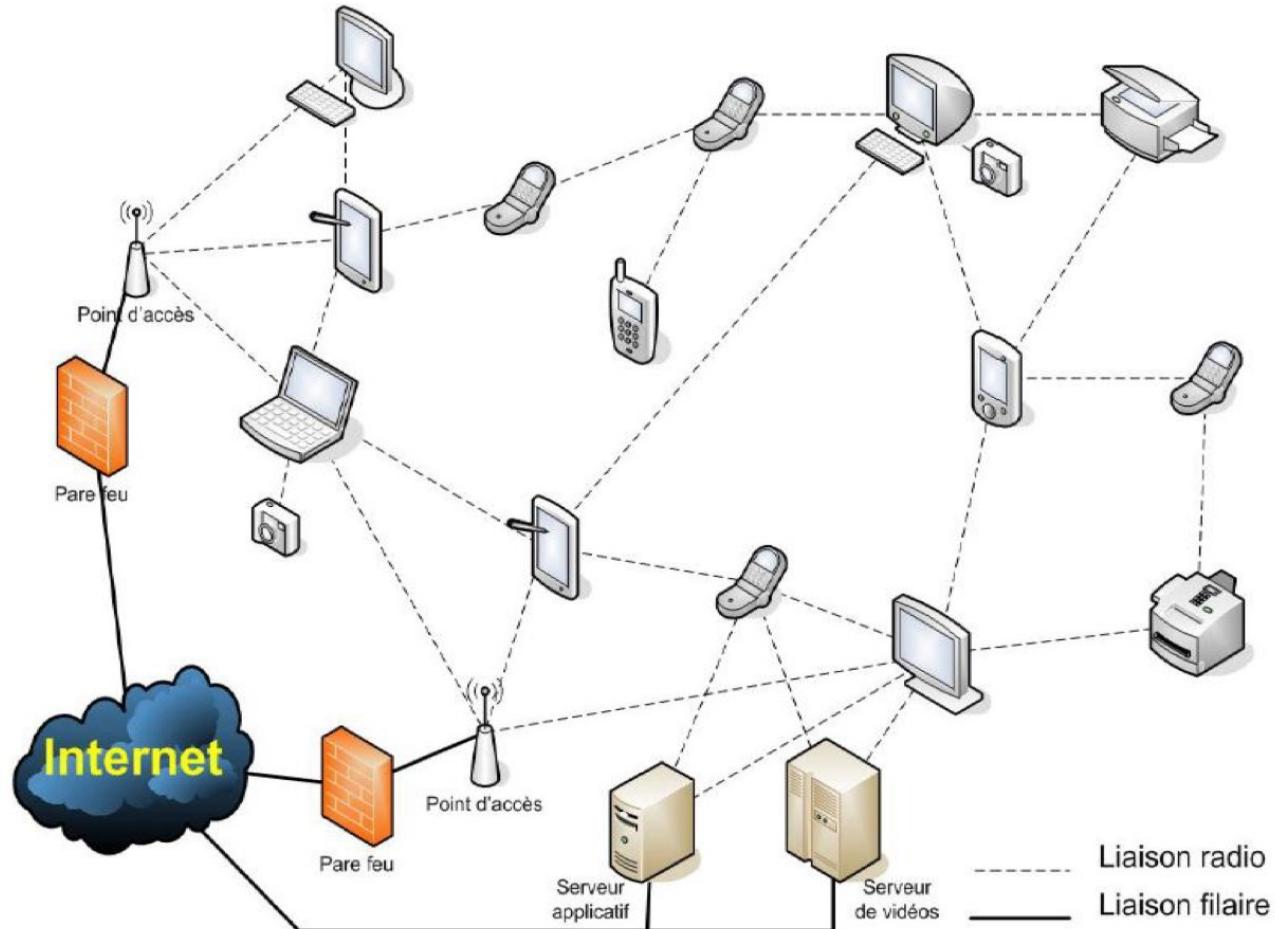


# Reseaux Ad hoc ou MANET

Un réseau Ad hoc, appelé généralement MANET (Mobile Ad hoc Network):

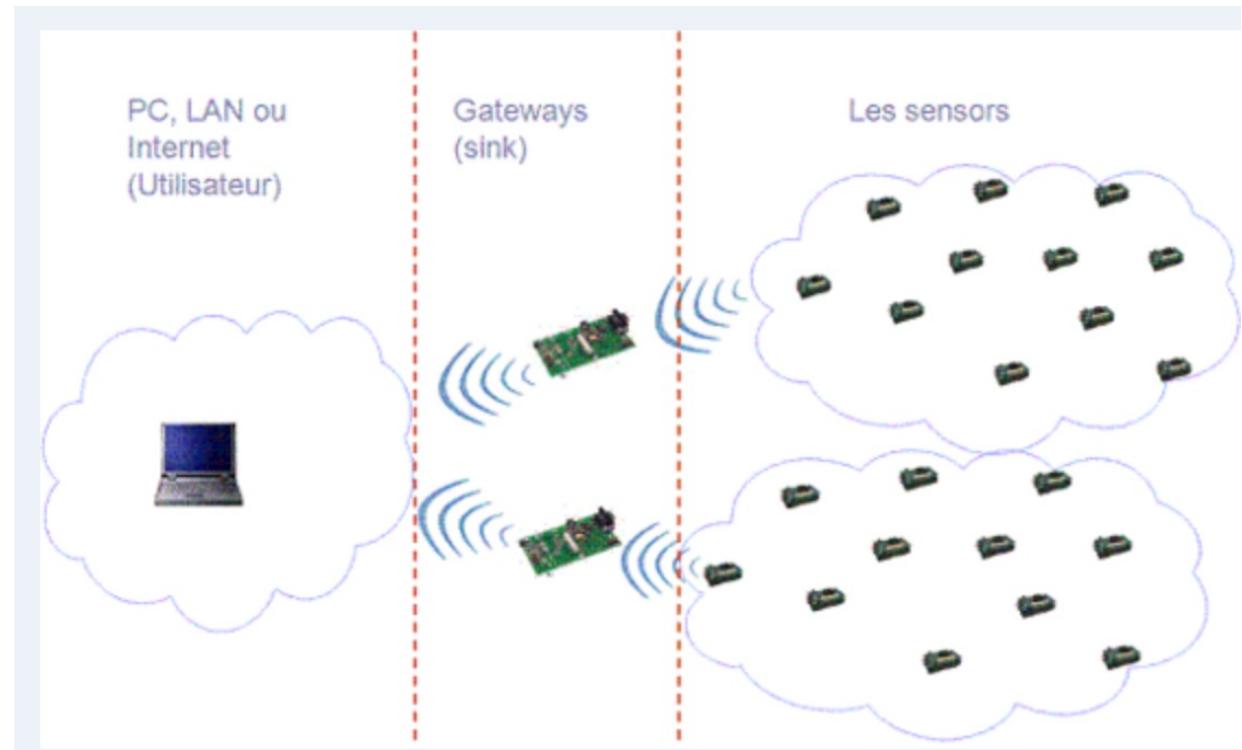
- ✓ Une collection d'unités mobiles munies d'interfaces de communication sans-fil
- ✓ Forme un réseau temporaire sans recourir à aucune infrastructure fixe ou administration centralisée.
- ✓ Dans de tels environnements, les unités se comportent comme des hôtes et/ou des routeurs.

# Reseaux Ad hoc ou MANET



# Reseau WSN

**WSN** collecte un ensemble de paramètres de l'environnement telles que la température ou la pression de l'atmosphère.



## Protocoles de connexion Réseau:Wifi

IEEE : 802.11a/b/g/n/ac

Besoins en mémoire : 1Mo +

Autonomie avec Pile : Jours

Vitesse de transfert : 11, 54, 108, 320, 1000 Mb/s

Portée (environ) : 300m

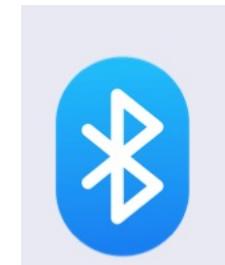
Bandes: 2,4 à 6Ghz



## Protocoles de connexion Reseau: Bluthooth

IEEE : 802.15.1

- Besoins mémoire : 250 ko +  
Autonomie avec Pile : Mois Vitesse de transfert : 1 Mb/s
- Portée (environ) : 10m
- Bande:2,4 à 2,5Ghz



## Protocoles de connexion Réseau :ZigBee

IEEE : 802.15.4

Besoins en mémoire : 4-32 ko

Autonomie avec Pile : Années

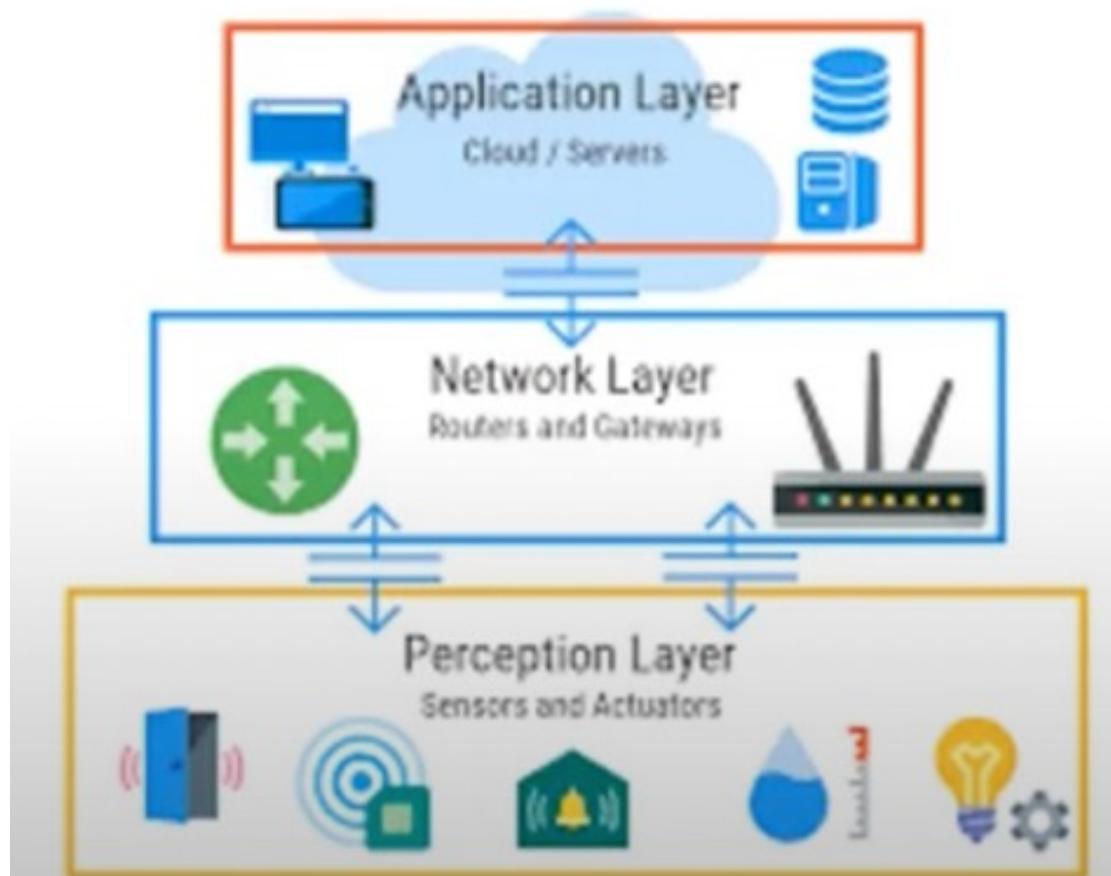
Vitesse de transfert : 20-250 kb/s

Portée (environ) : 100m

Bandé : 868 Mhz



# Architecture IOT IEEE-3-layers





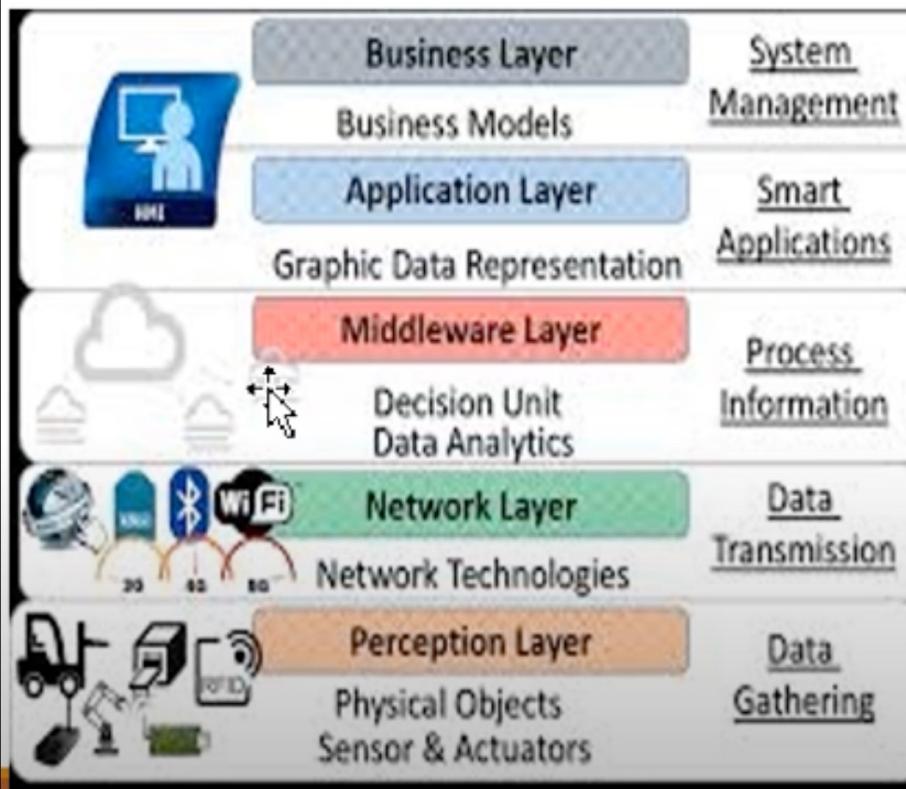
# architecture IOT IEEE-3-layers

**Couche Perception :** est similaire a la couche physique de OSI, elle contient les actionneurs et les capteurs.

**Couche reseau:** responsable de la connexion aux autres objets intelligents et serveurs. Elle utilise pour l'analyse et la transmission des donnees capteurs.

**Couche Application:** les applications mobiles- E health- les applications de gestion l'énergie

# Architecture 5 couches

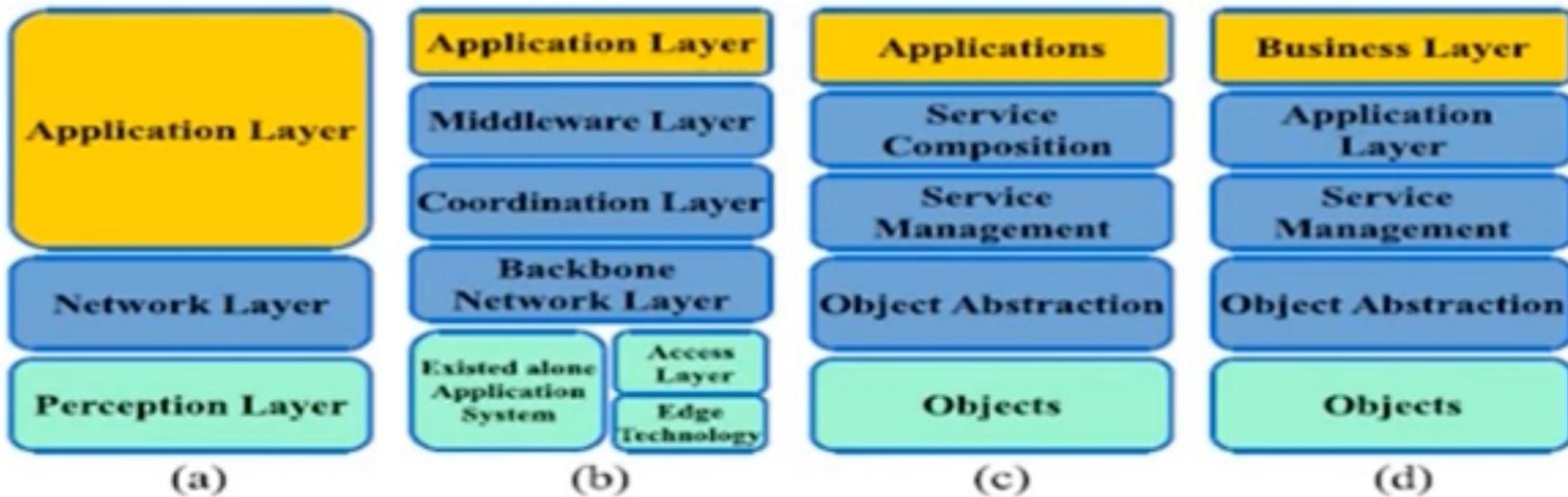


## Couche Transport( Reseaux):

Transfert les donnees capteurs de la couche Perception a la couche de traitement wireless,3G,LAN,Bluthooth,RFID et NFC.

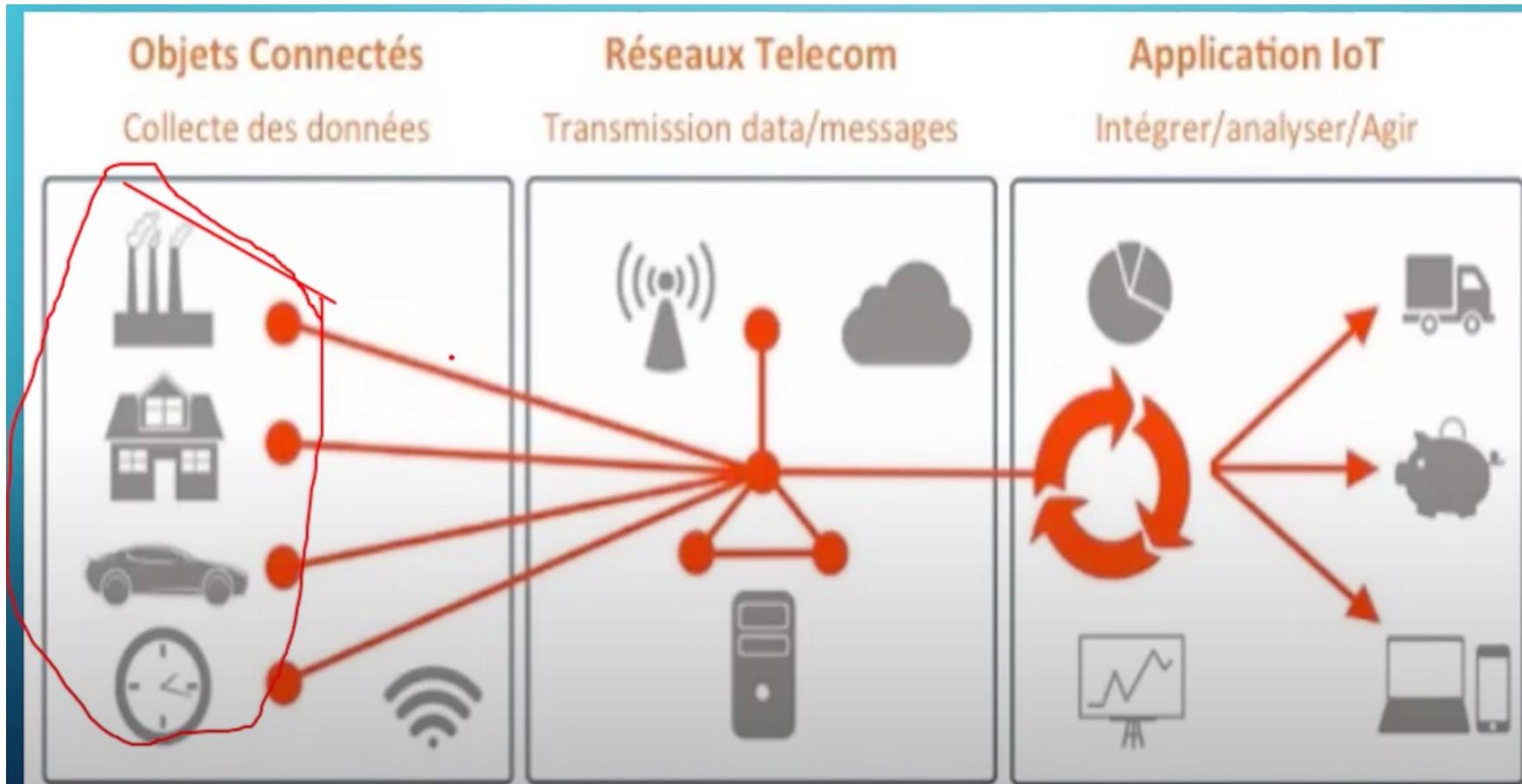
**Couche Analyse :** stocker, analyser et traiter un volume important de donnees, elle emploie plusieurs technologies cloud computing,databases, et big data processing.

**Couche Business:** gere tout le systeme IOT, les applications, le volet securite.

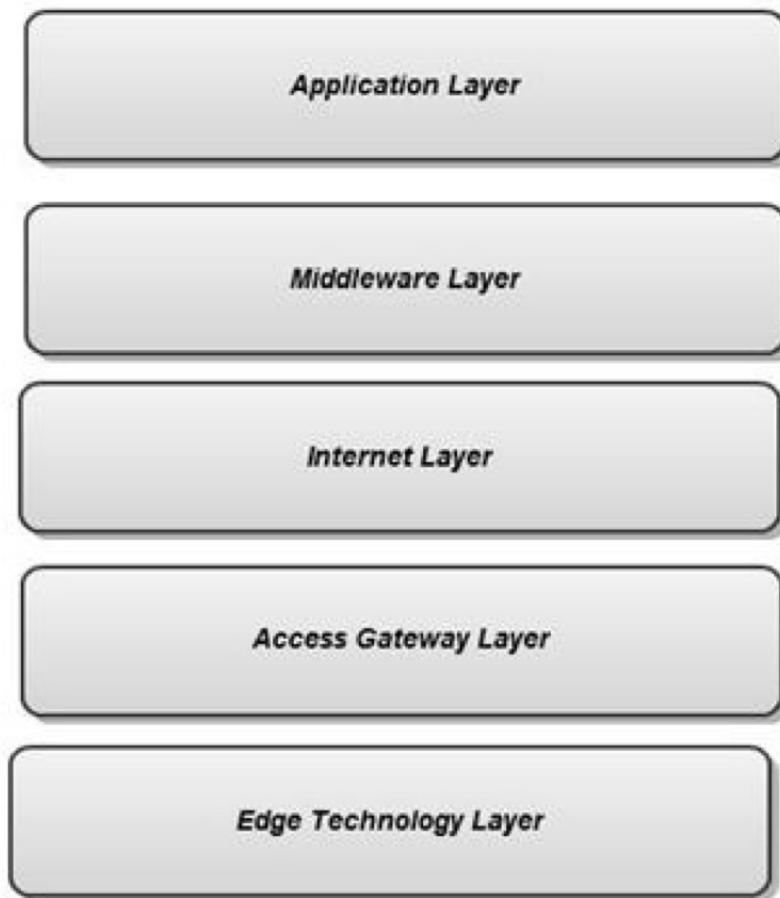


The IoT architecture. (a) Three-layer. (b) Middle-ware based. (c) SOA (d) Five Layer





# Architecture IoT sous forme de couches



# Architecture IoT sous forme de couches

**Couche *edge technology*** : la couche matérielle qui traite les entrées/sorties et gère les communications.

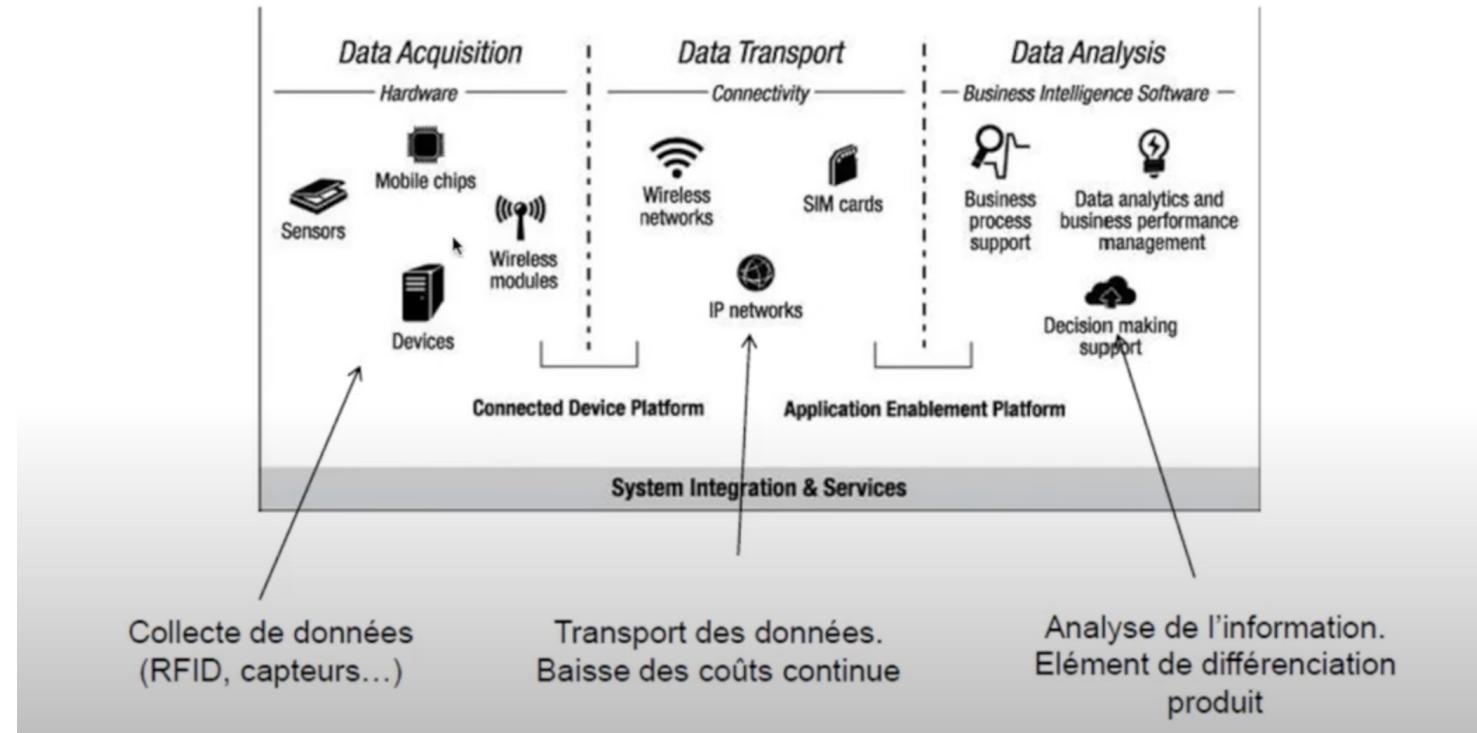
**Couche *access gateway*** : la couche qui gère le transport des données et leur routage vers Internet

**Couche *middleware*** : c'est la couche qui gère le traitement de données et fournit aux applications l'accès aux objets

**Couche *application*** : C'est la couche qui fournit différents services aux applications IOT.

# Internet des objets (couches)

Il n'existe pas une architecture commune pour tous les systèmes IOT.





# Couche acquisition

Permet de collecter les informations a partir de l'environnement

Les données captees de l'environnement physique et transformées en données numériques

Des capteurs intégrés dans la montre intelligente.





# Couche Transport

La couche qui permet de transmettre l'information de la partie capteur jusqu'à la partie stockage:

**Technologies de transmission au niveau local : LAN, PAN,ZigBee, bluthooth**

Exemple la ferme intelligente , on doit d'abord transmettre l'information au niveau local avant de la transmettre à l'application mobile

**Technologies de transmission à grande distance: service internet, réseau cellulaire**



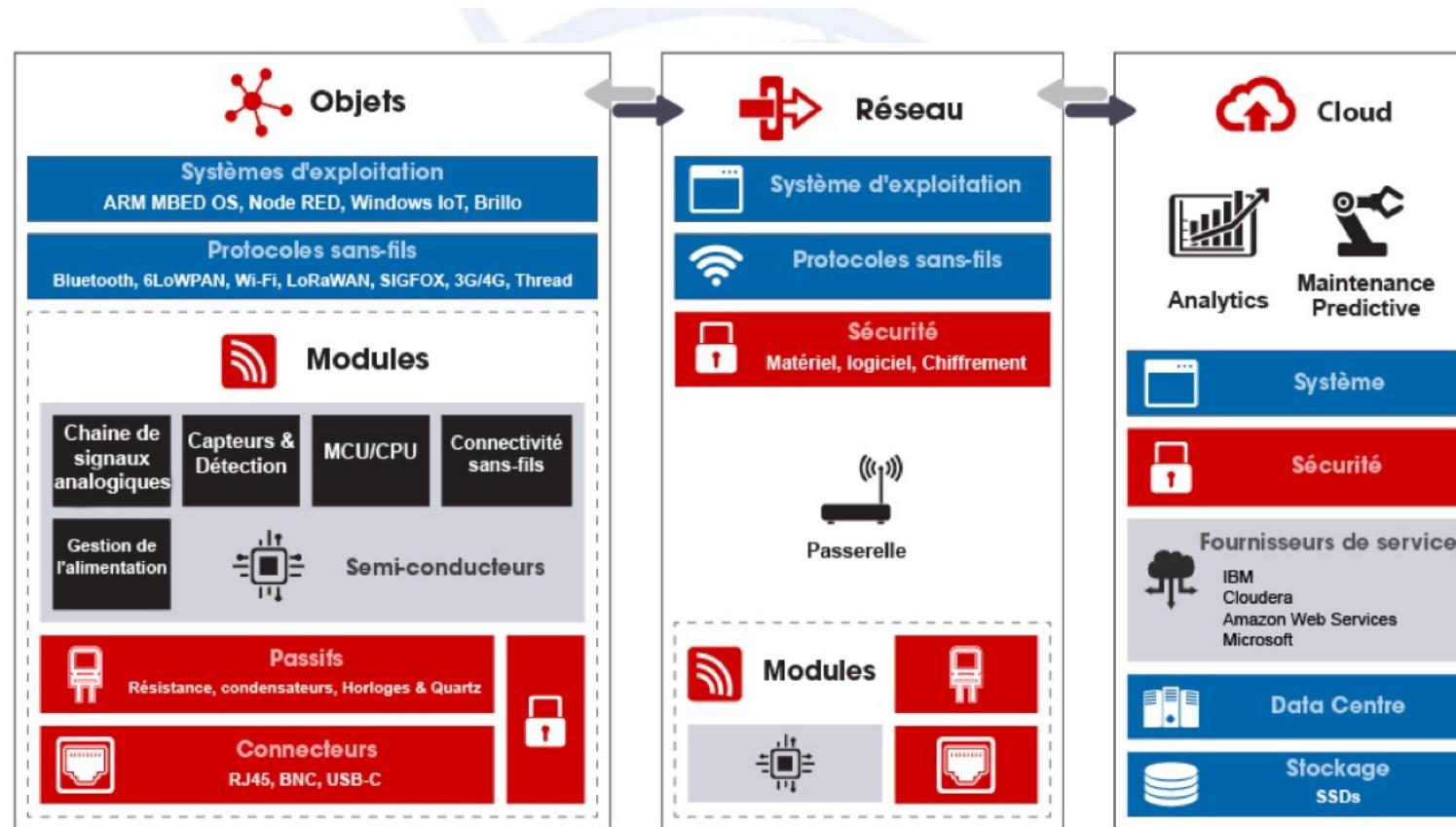
# Couche Analyse

Analyse les données collectées:

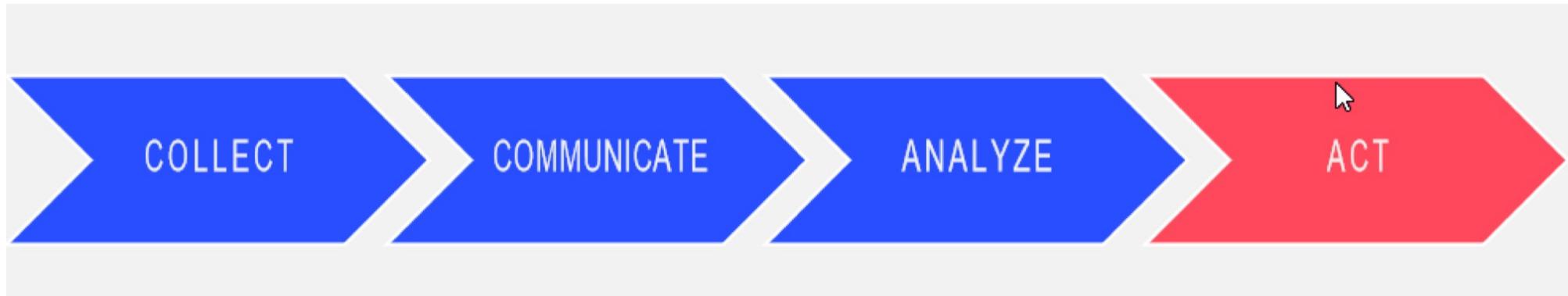
Une composante de stockage( cloud)

Une composante de prise de décision selon les données collectées et les paramètres enregistrés.

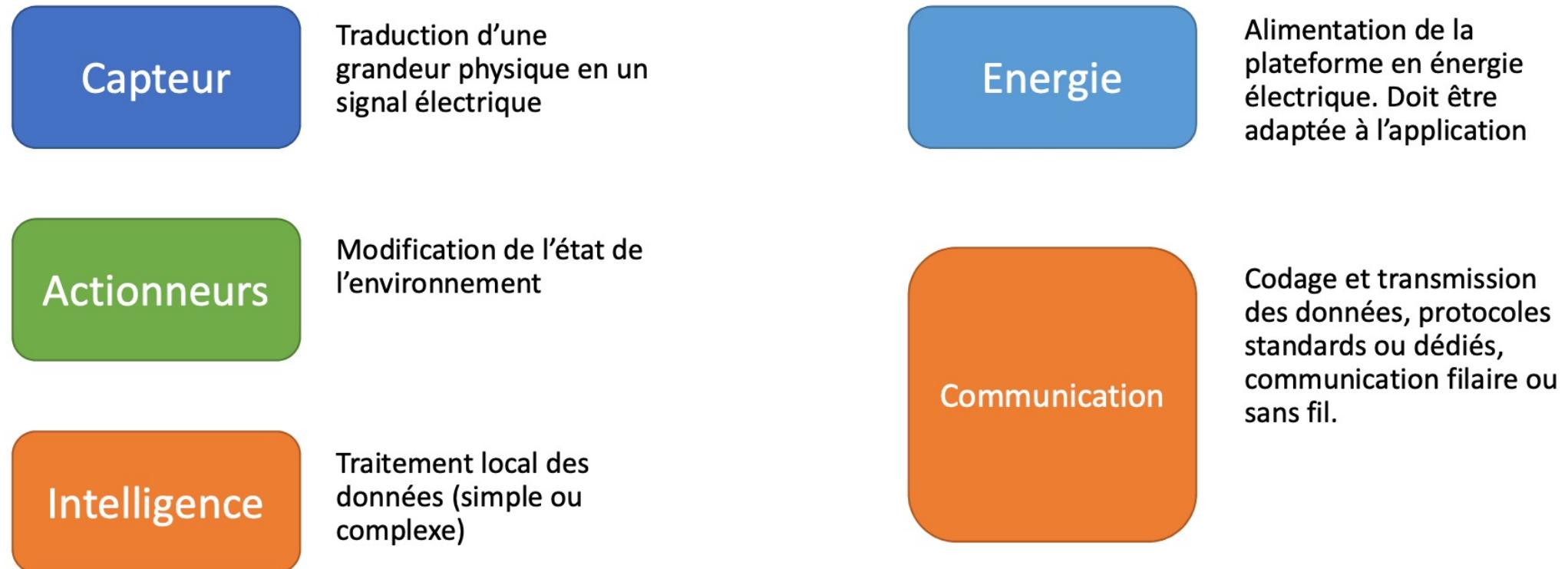
# Architecture fonctionnelle de L'IoT



# Cycle de vie de l'IoT



# Caractéristiques générales d'une plateformes pour l'IoT



# ZigBee / IEEE 802.15.4

