المدرسة الوطنية العليا للذكاء الاصطناعي وعلوم المعطيات - تارودانت
ÉCOLE NATIONALE SUPÉRIEURE DE L'INTELLIGENCE ARTIFICIELLE
ET SCIENCES DES DONNÉES - TAROUDANT

Module/Eléments: Systèmes Embarqués

Filière: Sécurité IT et confiance numérique (S2)

AU: 2024/2025

Pr. MOUDOUD

Plan

Chapitre I: Généralités sur les Systèmes Embarqués

Chapitre II: Les capteurs et les actionneurs

Chapitre III: Les systèmes programmables: microcontrôleurs, processeurs, …

Chapitre IV: Les protocoles de communication et internet des objets (IoT)

Système embarqué (SE)

❑ Définition:

✓ Les systèmes embarqués sont des composants électroniques issus de la combinaison de **matériel**, de **logiciel** qui interagissent avec l'**environnement**. Ils sont modélisés pour résoudre un problème ou effectuer des tâches spécifiques.

✓ Un système embarqué peut être défini comme un système de calcul **électronique (Hardware)** et **informatique (Software)** autonome conçu pour réaliser une tâche particulière. Souvent, un SE ne possède pas des **entrées/sorties** standards comme un clavier ou un écran d'ordinateur (PC).

✓ Un système embarqué désigne un matériel électronique comprenant au moins un **microprocesseur** (microcontrôleur) et un **logiciel** dédié à sa gestion;

✓ Le système matériel et l'application sont intimement liés et noyés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type PC.

Système embarqué

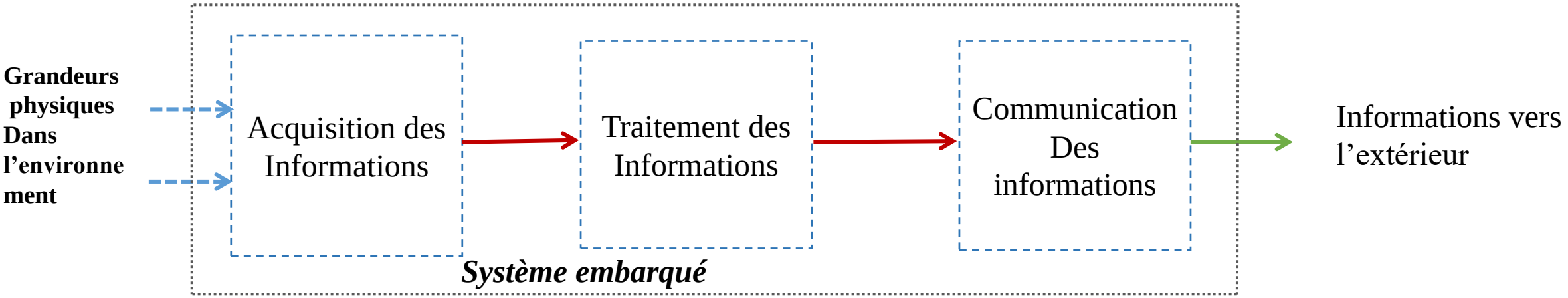
De nos jours, les systèmes embarqués (SE) sont donc **partout**, **discrets**, **efficaces et dédiés** à ce à quoi ils sont destinés.

Ils sont bourrés **d'électronique** plus ou moins complexe et **d'informatique** plus ou moins évoluée.

Un système embarqué est un **système informatique** qu'on va intégrer dans le **monde réel**. par exemple : Tout le traitement électronique qu'il y a dans une voiture, dans un appareil photo, une cafetière, une machine à laver.

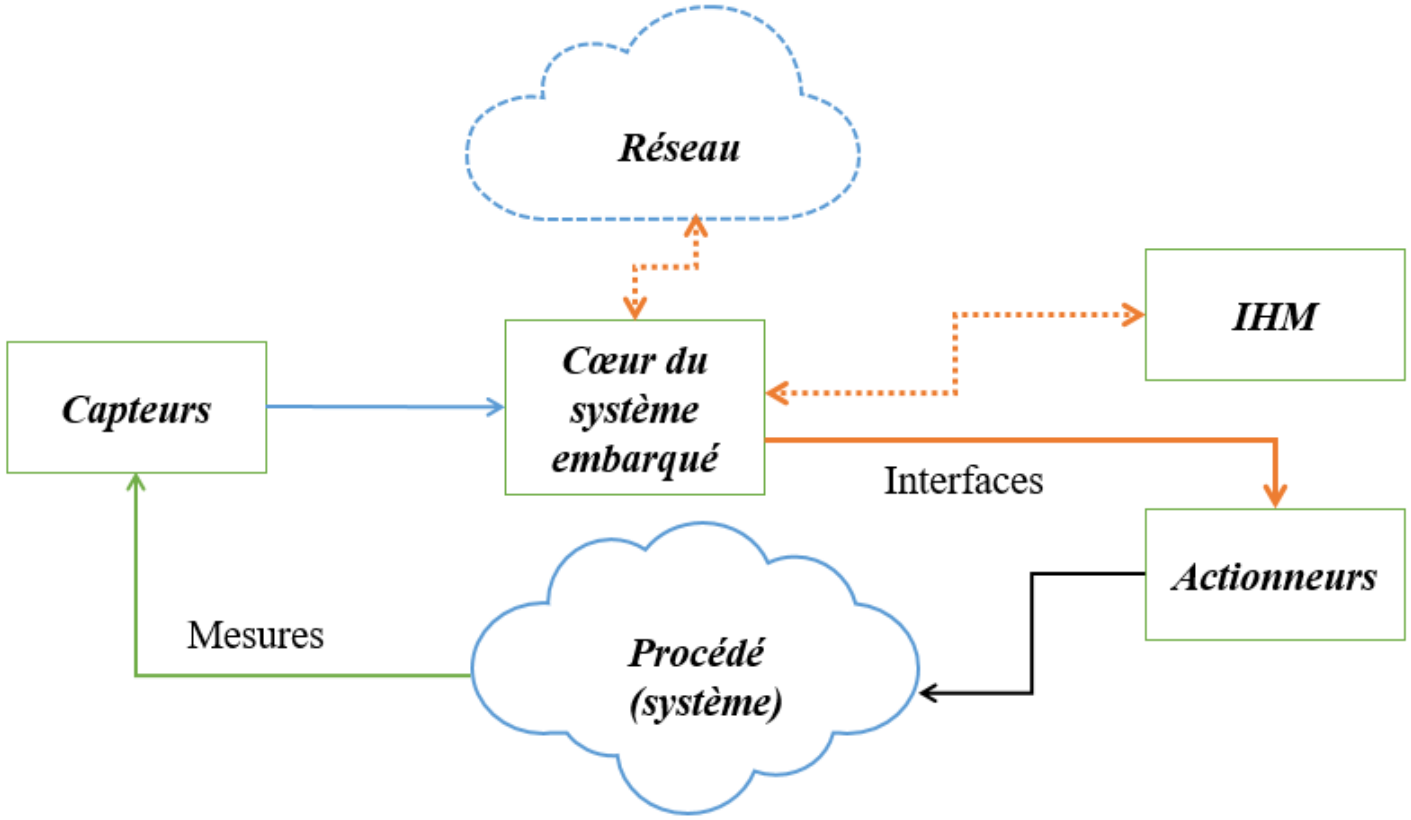
Système autonome qui peut fonctionner sans intervention humaine : caisse automatique de supermarché, remontée mécanique automatique, portail automatique pour accès à une résidence, voiture automatique

❖ Architecture des systèmes embarqués



Système embarqué

❖ Architecture des systèmes embarqués

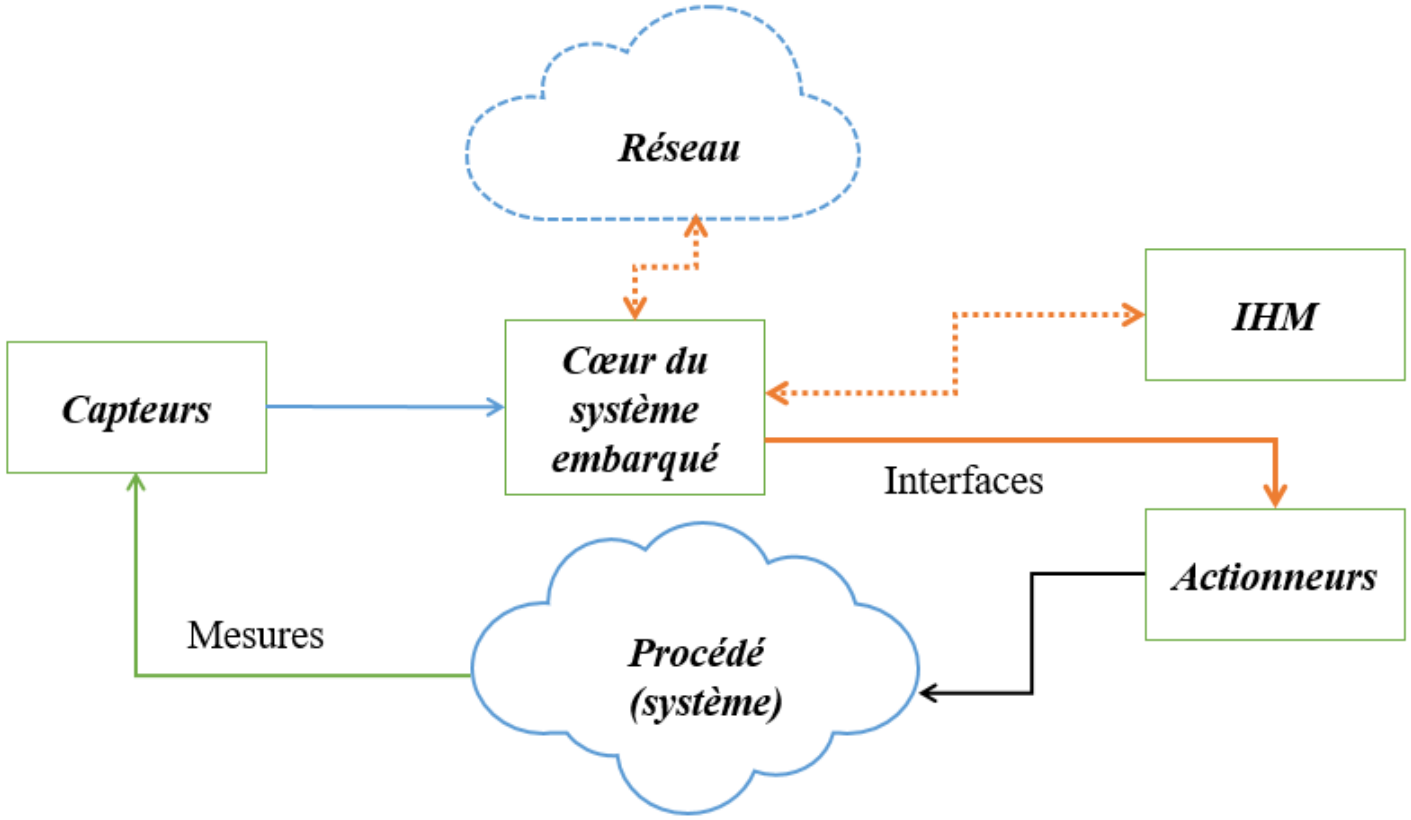


- **Capteurs** : Dispositifs qui mesurent des grandeurs physiques (température, pression, lumière, etc.) et les convertissent en signaux exploitables par le système.

- **Cœur du système embarqué** : Il s'agit du microcontrôleur ou du microprocesseur qui exécute le programme embarqué.

Système embarqué

❖ Architecture des systèmes embarqués



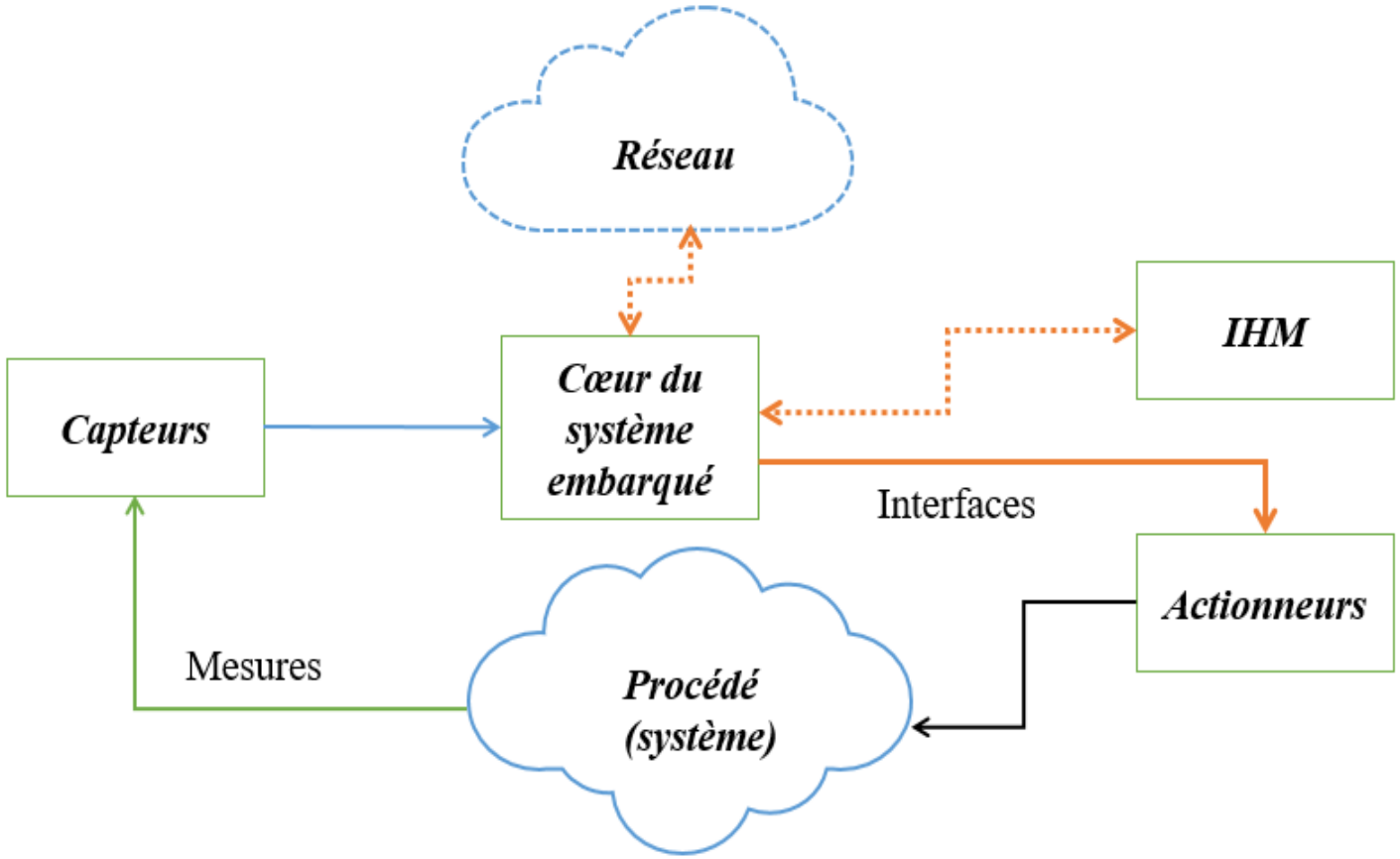
- **Actionneurs** : Composants qui effectuent une action en fonction des commandes du système (moteurs, relais, LED, etc.).

- **Interface Homme Machine (IHM) :** Moyens de communication et interaction entre le système et l'homme.

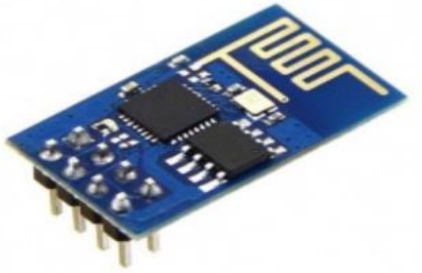
- **Différentes interfaces:** Moyens de communication entre le système et d'autres dispositifs (IHM, réseaux, autres systèmes embarqués).

Système embarqué

❖ Architecture des systèmes embarqués



❑ **Réseau :** Permet la transmission de données vers un serveur ou d'autres systèmes ou objets connectés.



Module Wi-Fi

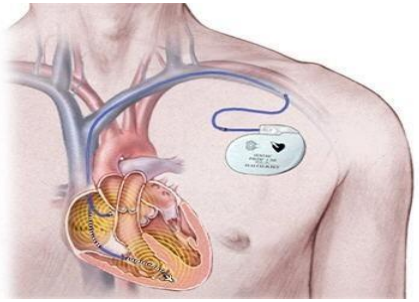
Module Bluetooth

❑ **Procédé (système) :** L'application finale dans laquelle s'insère le système embarqué.

Système embarqué

❖ Domaines d'application (par exemples)

❖ L'agriculture



❖ La santé

❖ Finance et banques

Système embarqué

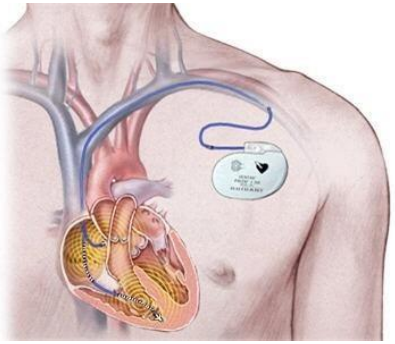
❖ Domaines d'application (par exemples)

➢ **<u>Domaines « traditionnels »</u>**

- ✓ **Avionique**
- ✓ **Robotique**
- ✓ **Automobile**
- ✓ **Militaire**

➢ **<u>Domaines « nouveaux »</u>**

- ✓ **Jeux et loisirs**
- ✓ **Téléphonie, Internet mobile**
- ✓ **Implants (santé sécurité)**
- ✓ **Immeubles intelligents**
- ✓ **Villes intelligentes**
- ✓ **Vêtements…**



Système embarqué

Classification des systèmes embarqués

❖ *Système Transformationnel :*

C’est un système embarqué qui traite ses données en une **seule exécution**, sans interaction continue. Il **lit** ses entrées, effectue un calcul, **génère** une sortie, puis s'arrête (Par exemple l’imprimante).

❖ *Système Interactif :*

Système en interaction **quasi permanente** avec son environnement, y compris après l'initialisation du système; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passés); **le rythme de l'interaction est déterminé par le système et non par l'environnement** (souvent utilisé dans les système de surveillance)

❖ *Système Réactif ou Temps Réel :*

Système en interaction **permanente** avec son environnement, y compris après l'initialisation du système; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passées); **mais le rythme de l'interaction est déterminé par l'environnement et non par le système.**

Système embarqué

Caractéristiques des systèmes embarqués

Les systèmes embarqués possèdent plusieurs caractéristiques spécifiques qui les distinguent des systèmes informatiques classiques.

Caractéristiques principales

- **Fonctionnement en temps réel :**
  pour garantir un bon fonctionnement les systèmes embarqués doivent respecter des délais précis.
- **Faible consommation d'énergie:**
  Utilisation de processeurs basse consommation et gestion efficace de l'alimentation.
- **Combinaison matériel-logiciel:**
  Le logiciel est souvent développé sur mesure pour une architecture matérielle spécifique.
- **Communication/connectivité:**
  Les systèmes embarqués utilisent des protocoles de communication, permettant le contrôle à distance et les mises à jour logicielles.

Aperçu général sur les capteurs

Souvent la grandeur physique à capter (pression par exemple) est transformée à l'intérieur du capteur en d'autres grandeurs physiques intermédiaires (résistance par exemple) avant d'être transformée en une grandeur électrique utilisable (tension analogique par exemple).

On peut classer les capteurs en fonction :

- ✓ **du signal de sortie** : capteurs tout ou rien ou TOR (signal logique), numériques, analogiques.

- ✓ **de l'élément sensible en interne** : capteurs résistifs, inductifs, capacitifs, optiques (ou photosensibles), à semi-conducteur, piézo-électrique, …

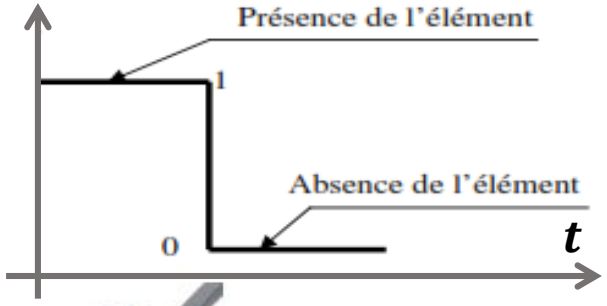
- ✓ **de la grandeur physique d'entrée** : capteurs de position, de vitesse, de température, de force, de pression, de débit, …

Aperçu général sur les capteurs

Classification selon du signal de sortie

Capteurs tout ou rien (T.OR):

Ces capteurs génèrent une information électrique de type **binaire** (Vrai ou faux) qui caractérise le phénomène à détecter ou capter.

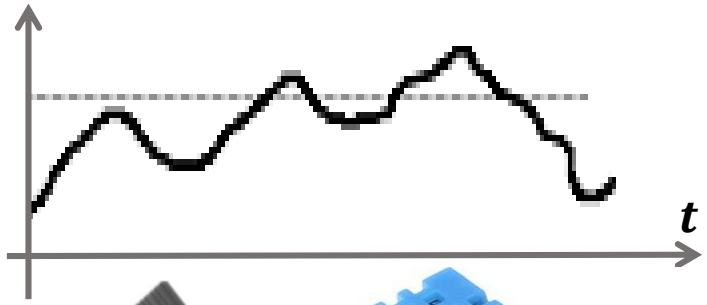


Capteur inductif

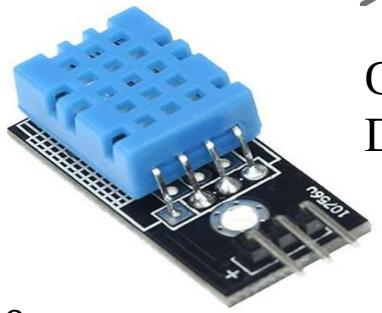
Détecteur Du gaz (MQ-2)

Capteurs analogiques:

Pour ces capteurs la grandeur électrique de sortie est fonction de la grandeur physique à mesurer. Le signal de sortie est analogique (continu) en fonction du temps (voir la figure suivante)



Capteur de température Pt100



Capteur D'humidité DH11

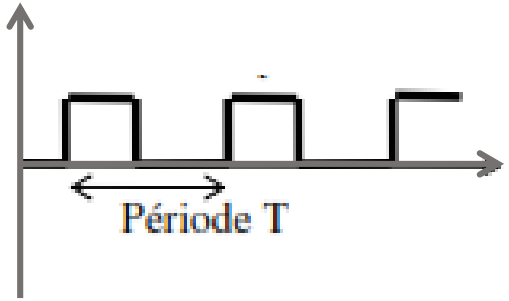
Capteurs numériques:

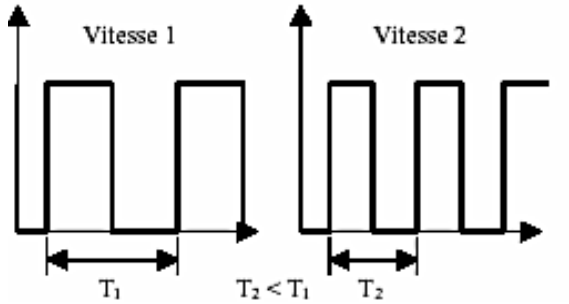
Aperçu général sur les capteurs

❑ Capteurs numériques:

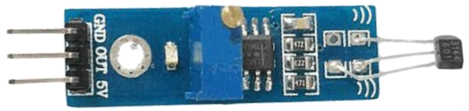
Ce type de capteur délivre en sortie une information électrique à caractère numérique, image de la grandeur physique à mesurer, c'est à dire ne pouvant prendre qu'un nombre limité de valeurs distinctes.
L'information délivrée par ces capteurs peut être représentée par :

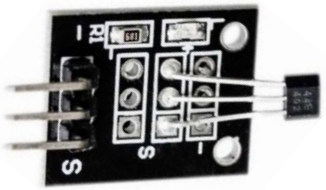
- Soit un signal électrique **périodique** (Signal carré) à période variable





Exemple d'un capteur de vitesse

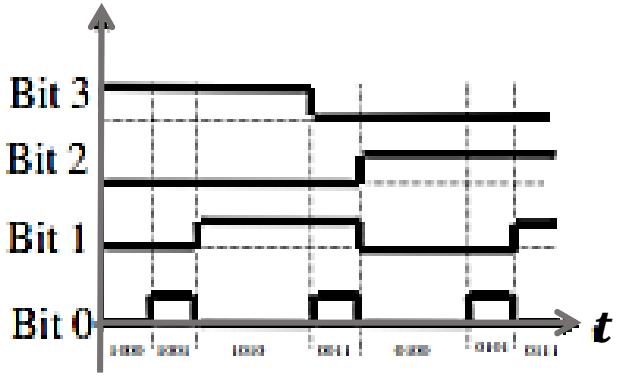




Capteur à effet Hall

- Soit un signal **numérique** codé sur n variables binaires (n Bits).

Exemple d'un codeur dont le signal de sortie (codé sur 4 bits)caractérise la grandeur d'entré



Exemple: capteur rotatif de position

Aperçu général sur les capteurs

Aussi la grandeur de sortie peut être soit : une charge, une tension ou un courant, dans ces 3 cas, le capteur est **actif** ; si la grandeur est une impédance : R, L ou C le capteur est **passif**.

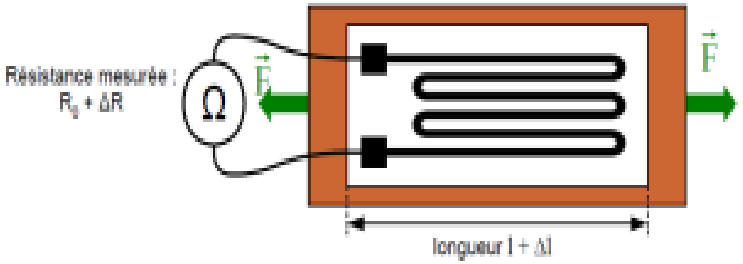
❖ Capteurs passifs

Il s'agit généralement d'**impédance** dont l'un des paramètres déterminants est sensible à la grandeur mesurée. La variation d'impédance résulte :

- Soit d'une **variation de dimension** du capteur, c'est le principe de fonctionnement d'un grand nombre de capteur de position, potentiomètre, inductance à noyaux mobile, condensateur à armature mobile.

Exemple de capteur inductif et capacitif

- Soit d'une **déformation résultant** de force ou de grandeur s'y ramenant, pression accélération (armature de condensateur soumise à une différence de pression, jauge d'extensomètrie liée à une structure déformable).



Capteur de force

Capteur de pression

Aperçu général sur les capteurs

Aussi la grandeur de sortie peut être soit : une charge, une tension ou un courant, dans ces 3 cas, le capteur est **actif** ; si la grandeur est une impédance : R, L ou C le capteur est **passif**.

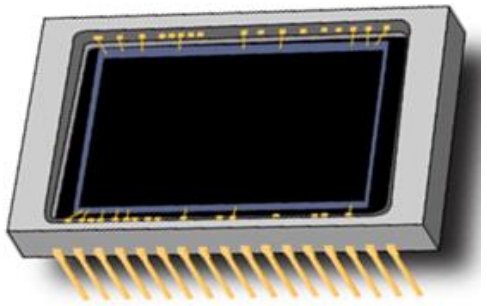
❖ Capteurs actifs

Fonctionnant en générateur, un capteur actif est généralement fondé dans son principe sur un **effet physique** qui assure la conversion en énergie électrique de la forme d'énergie propre à la grandeur physique à prélever, énergie thermique, mécanique ou de rayonnement. Les plus classiques sont :

Effet photo-électrique :

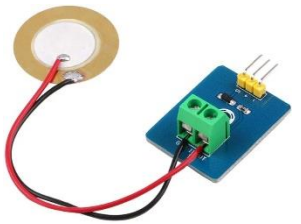
La libération de charges électriques dans la matière sous l'influence d'un rayonnement lumineux ou plus généralement d'une onde électromagnétique dont la longueur d'onde est inférieure à un seuil caractéristique du matériau.

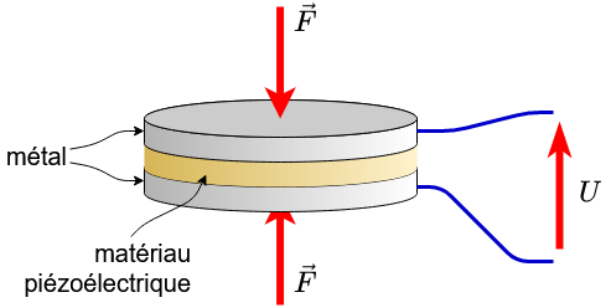
Exemple: Capteur CCD utilisé Dans les appareils photos



Effet piézo-électrique :

L'application d'une contrainte mécanique σ à certains matériaux dits piézo-électrique (le quartz par exemple) entraîne l'apparition d'une déformation et d'une même charge électrique de signe différent sur les faces opposées.





Aperçu général sur les capteurs

Exemples des capteurs d'environnement

THERMOMÈTRE INFRAROUGE

CAPTEUR LUMIÈRE RGB

CELLULE PHOTOÉLECTRIQUE

CAPTEUR DE FLAMME

PRESSION/ALTITUDE

CAPTEUR DE SON

CAPTEUR D'HUMIDITÉ DES SOLS

CAPTEUR UV

CAPTEURS DE GAZ

THERMISTOR

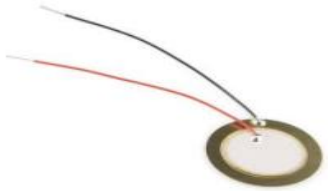
CAPTEUR DE PLUIE

COMPTEUR GEGER

Aperçu général sur les capteurs

Exemples des capteurs du mouvement

DETECTEUR D'OBSTACLE



CAPTEUR PIÉZOÉLECTRIQUE

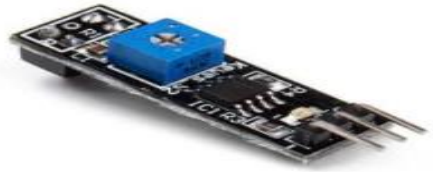
CAPTEUR DE MOUVEMENT PIR

ACCÉLÉROMÈTRE

CAPTEUR DE FLEXION

CAPTEUR DE PETITES VIBRATIONS

CAPTEUR TILT



CAPTEUR IR "SUIVEUR DE LIGNE"

CAPTEUR ULTRASON

CAPTEUR D'ANGLE

CELLULE DE CHARGE

CAPTEUR EFFET HALL

Aperçu général sur les capteurs

Exemples des capteurs d'interaction avec l'humain

LECTEUR EMPREINTE DIGITALE

CAPTEUR DE PULSATIONS

CAPTEUR COURANT MUSCULAIRE

CAMÉRA THERMIQUE

CAMÉRA INFRAROUGE

CAPTEUR CAPACITIF

JOYSTICK

ECRAN TACTILE

LECTEUR RFID

Aperçu général sur les capteurs

Exercice d'application

N°	Proposition	V	F
1	Un capteur analogique fournit un signal binaire (0 ou 1).		
2	Un capteur passif fonctionne sans alimentation externe et réagit directement à la grandeur mesurée.		
3	Un capteur de température DS18B20 est un capteur analogique.		
4	Un détecteur de présence PIR est un capteur logique car il ne donne que deux états (présence ou absence).		
5	Un capteur actif peut générer un signal de sortie sans alimentation externe.		

Q1/ Quelle est la différence entre un capteur analogique et un capteur numérique?
Q2/ Quelle est la différence entre un capteur actif et un capteur passif?
Q3/ Classifier les capteurs suivant en capteurs analogiques, logiques ou numériques:
Capteur de lumière, effort, pression, vitesse de rotation, température, détecteur de présence, effet Hall;

Aperçu général sur les actionneurs

Dans les **systèmes embarqués**, les **actionneurs** sont des composants qui convertissent un signal de commande (électrique, électronique ou numérique) en une action physique (mouvement, force, pression, etc.). Ils sont essentiels pour interagir avec l'environnement, notamment dans **systèmes industriels à savoir: les robots, voitures intelligentes, drones, ...**

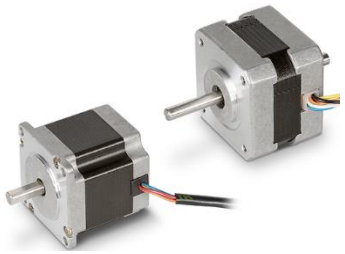
Catégories des actionneurs

❑ Actionneurs électriques (Moteurs)

Servo-moteurs

↓

Robotique



Moteurs pas à pas

↓

Imprimantes 3D

❑ Actionneurs Électrostatiques

Micro-actionneurs MEMS: très miniaturisés
Et utilisés dans les smartphones

Actionneurs capacitifs:
Utilisés dans
les écrans tactiles,

❑ Actionneurs hydrauliques/pneumatiques

Automobile : Direction assistée hydraulique, freins.

Médical : Outils chirurgicaux, respirateurs.

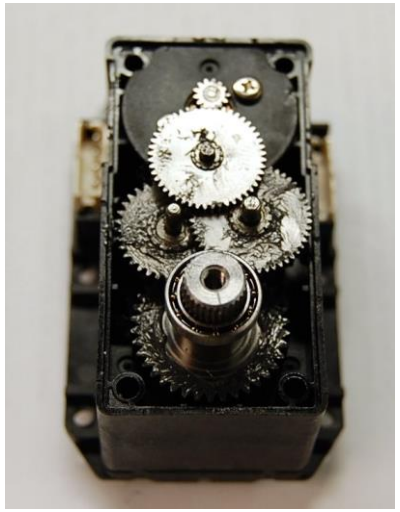
Aperçu général sur les actionneurs

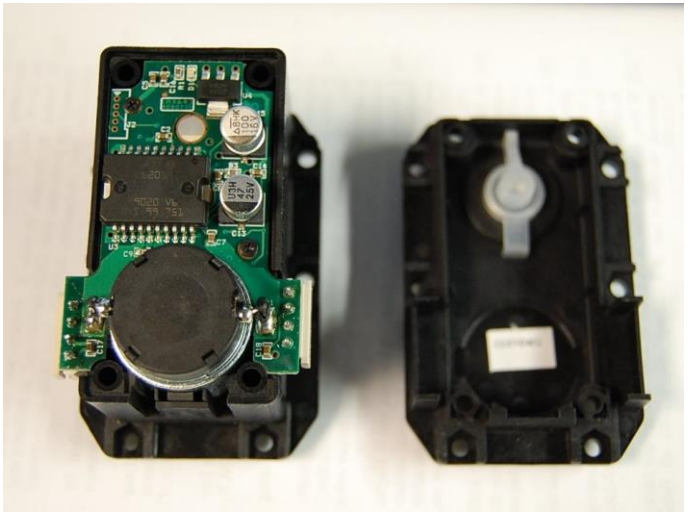
❖ Servo-moteurs

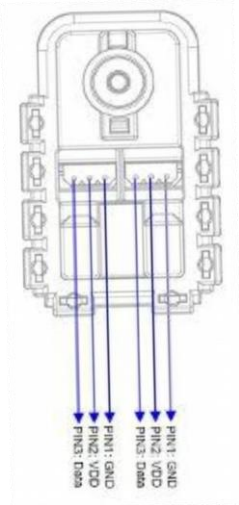
Gamme servomoteurs **programmables hautes performances tout en un** de la société Robotis (Corée du Sud):

Servo dynamixel

Moto-réducteur + Distributeur + Asservissement pos/vitesse + Réseau







Hautes performances : robustesse, précision, **fonctionnalités**

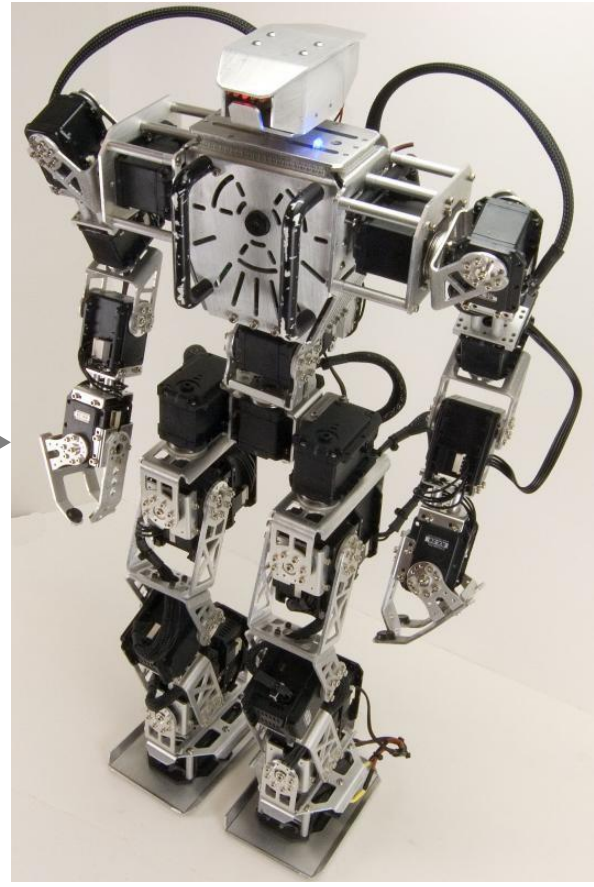
Aperçu général sur les actionneurs

❖ **Servo-moteurs**

Exemples d'applications

Caméra suiveuse

Robot humanoïde Giger



Robot Darwin

Servo dynamixel

Bras robotisé Spectrosun

Aperçu général sur les actionneurs

❖ Servo-moteurs

Gammes disponible

Ces actionneurs sont caractérisés par:

✓ **Contrôle précis** : Position, vitesse et couple réglables.

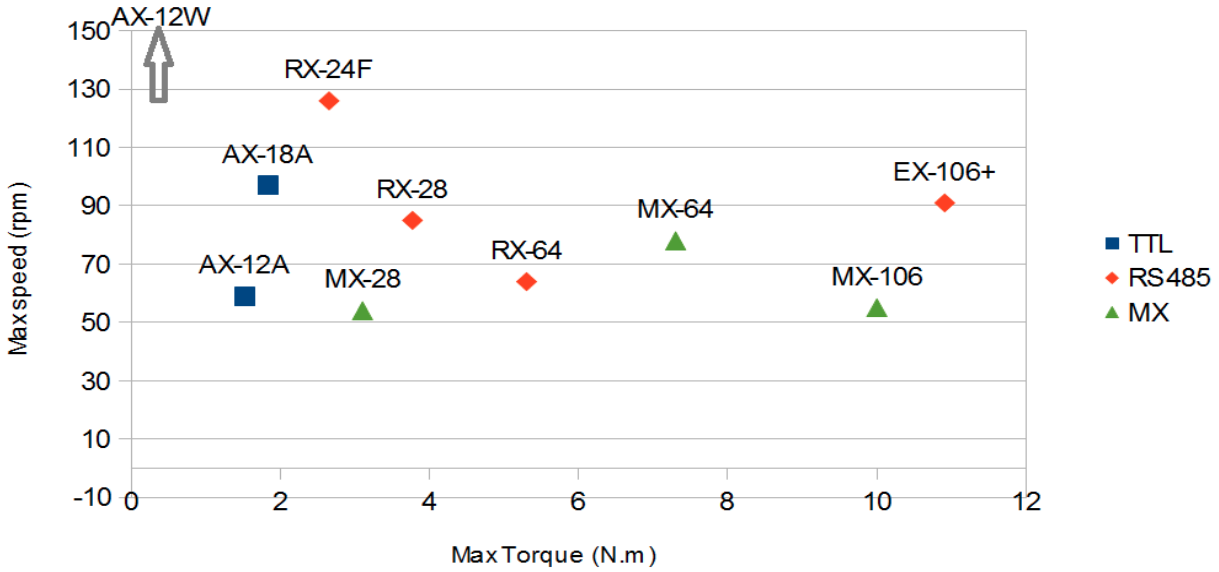
✓ **Communication numérique** : Protocoles **TTL ou RS-485** (selon modèle).

✓ **Capteurs intégrés** : Retour d'information sur température, charge, tension, etc.

✓ **Chaînage facile** : Plusieurs servos connectés en **daisy-chain** via un seul bus.

✓ **Boîtier robuste** : Matériaux en métal ou plastique selon la gamme.

✓ **compatibilité/contrôlabilité** : Compatible avec **Arduino, Raspberry Pi, ROS, C++, Python**.



Aperçu général sur les actionneurs

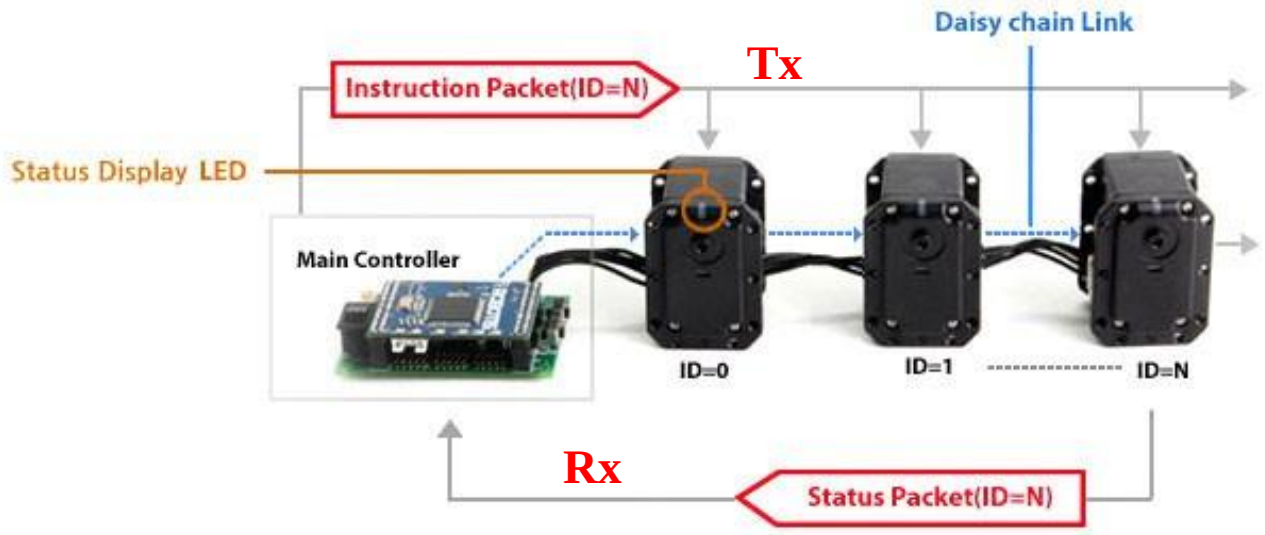
❖ Servo-moteurs

Possibilités de contrôle :

Position, vitesse, couple, correcteur PID (suivant modèle), précision du positionnement, rigidité du maintien en position, Amplitude max. en position, température limite de fonctionnement Tensions limites de fonctionnement

Débit de communication, identifiant,

...



Il est donc possible de connecter un servo en TTL directement sur un port série (Rx,Tx) de la carte Arduino. Les bibliothèques utilisées gèrent les synchronisations Rx,Tx.

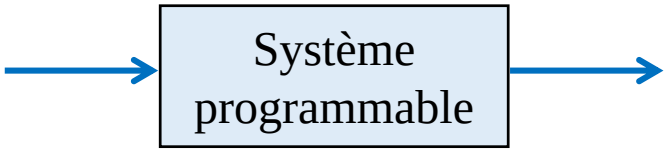
Retours :

Position, vitesse, courant, température ...

Jusqu'à 254 servos sur une seule ligne de données!!!

Aperçu général sur les systèmes programmables

Un **système programmable** dans un **système embarqué** est une plateforme matérielle (microcontrôleur, FPGA, DSP, etc.) associée à un logiciel permettant d'exécuter des tâches spécifiques de manière autonome et optimisée.



❖ Il assure le traitement des données (informations)

✓ Interprète les **signaux des capteurs** (température, pression, mouvement).
✓ Prend des **décisions en temps réel** pour contrôler le système.
✓ Exécute des **algorithmes** pour le traitement d'image, la communication ou l'IA.

❖ Il gère Interfaces de communication

✓ Échange des données avec d'autres systèmes via **UART, SPI, I2C, CAN, Ethernet**.
✓ Connecte le système embarqué aux **réseaux IoT** (WiFi, Bluetooth, LoRa).

❖ Il permet le pilotage des actionneurs

✓ Gère les **moteurs, servos, relais, vannes hydrauliques**.
✓ Réagit aux conditions en temps réel (détection d'obstacle, correction de trajectoire).

UART : *Universal Asynchronous Receiver-Transmitter*
SPI : *Serial Peripheral Interface*
I²C : *Inter-Integrated Circuit*
CAN : *Controller Area Network*

Aperçu général sur les systèmes programmables

Moins chère
Simple
Moins performante

Plus chère
Complex
Plus performante

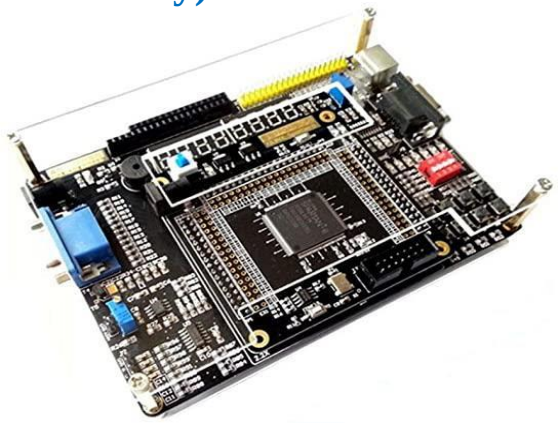
❑ Cartes Arduino

Programmation C/C++

❑ Cartes Raspberry

Programmation Python

❑ Cartes FPGA (Field-Programmable Gate Array)



ALTERA

QUARTUS® II

ModelSim

Langages VHDL/Verilog/VHDL-AMS

❑ Cartes DSP (Digital Signal Processor)

Langages de programmation C/C++/Python

Aperçu général sur les systèmes programmables

❑ **Différentes caractéristiques**

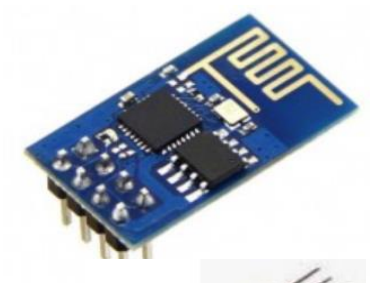
Critères	Arduino	Raspberry Pi	FPGA	DSP
Type de système	Microcontrôleur	Microprocesseur	Matériel reconfigurable	Processeur spécialisé
Architecture	AVR / ARM Cortex-M	ARM Cortex-A	Matrice de portes logiques (LUTs)	Architecture Harvard ou Von Neumann
Rapidité	Faible (16-100 MHz)	Moyenne (1-2 GHz)	Très élevée (MHz-GHz, parallèle)	Élevée (MHz-GHz, optimisé pour signal)
Complexité	Très simple (code en C++)	Moyenne (OS Linux requis)	Très complexe (VHDL/Verilog)	Moyenne (C, ASM, bibliothèques spécifiques)
Mémoire	Faible (2-512 KB RAM)	Moyenne (512 MB - 8 GB RAM)	Dépend du modèle (Mo - Go SRAM/DDR)	Variable (KB - MB de RAM optimisée)
Temps réel	Basique (interruptions)	Non adapté (OS non déterministe)	Excellent (traitement 100% matériel)	Très bon (optimisé pour temps réel)
Applications	**Projets embarqués, robotique simple**	**IoT, serveurs, multimédia**	**Traitement parallèle, IA, cryptographie**	**Traitement du signal (audio, vidéo, télécoms)**
Flexibilité	Facile à programmer, mais limité	Polyvalent, mais dépend de l'OS	Très flexible mais difficile à coder	Spécialisé pour le signal numérique

Interfaces de communication dans les systèmes embarqués

Les **interfaces de communication** jouent un rôle essentiel dans les **systèmes embarqués.** Elles permettent l'échange des informations entre les différentes composantes à savoir **microcontrôleurs, capteurs, actionneurs, mémoires et autres périphériques**, où entre différents systèmes. Elles assurent **l'interconnexion, la transmission rapide et fiable des données** pour le bon fonctionnement du système ou ensemble des systèmes.

❑ Communication sans fil

Wi-Fi



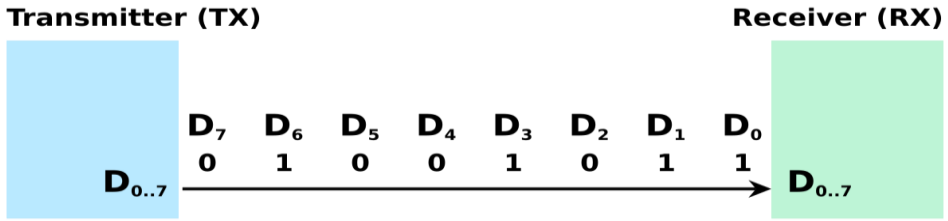
Bluetooth

ZigBee

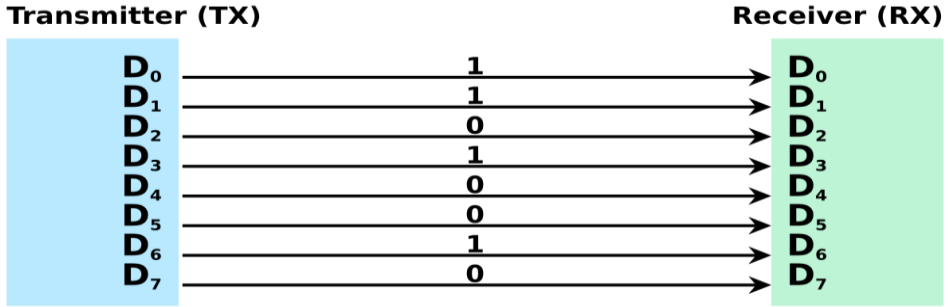
GSM/GPRS

❑ Communication filaire (fibre optique, fils de cuivre)

❑ *Communication série (USB, I2C, …): exemple le protocole RS232*

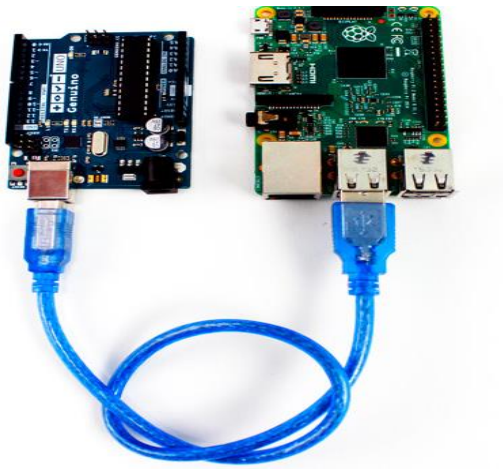


❑ *Communication parallèle (RAM, LCD …)*

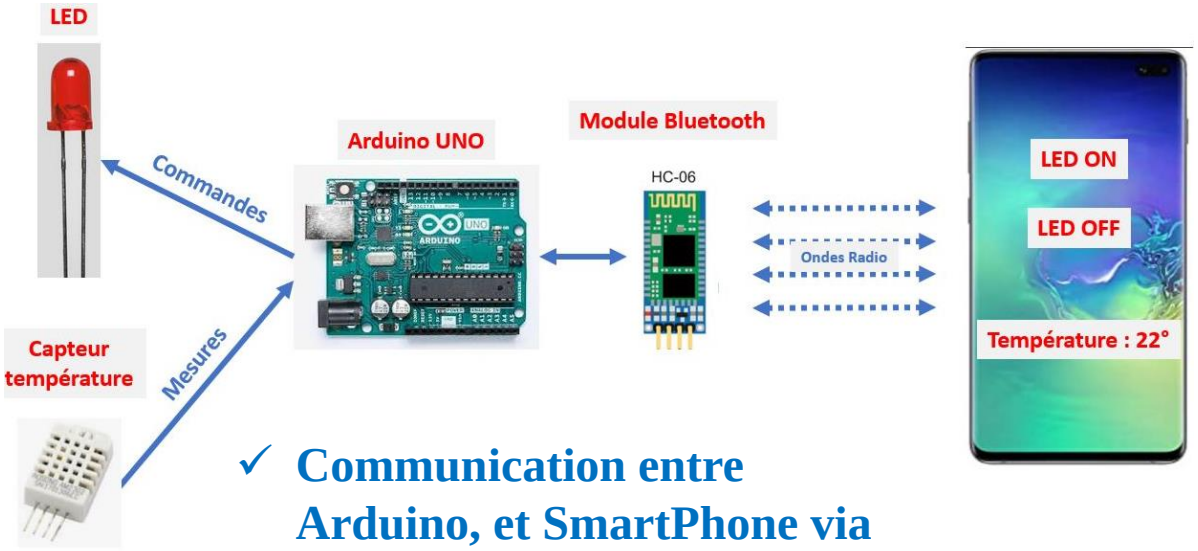


Interfaces de communication dans les systèmes embarqués

Exemples

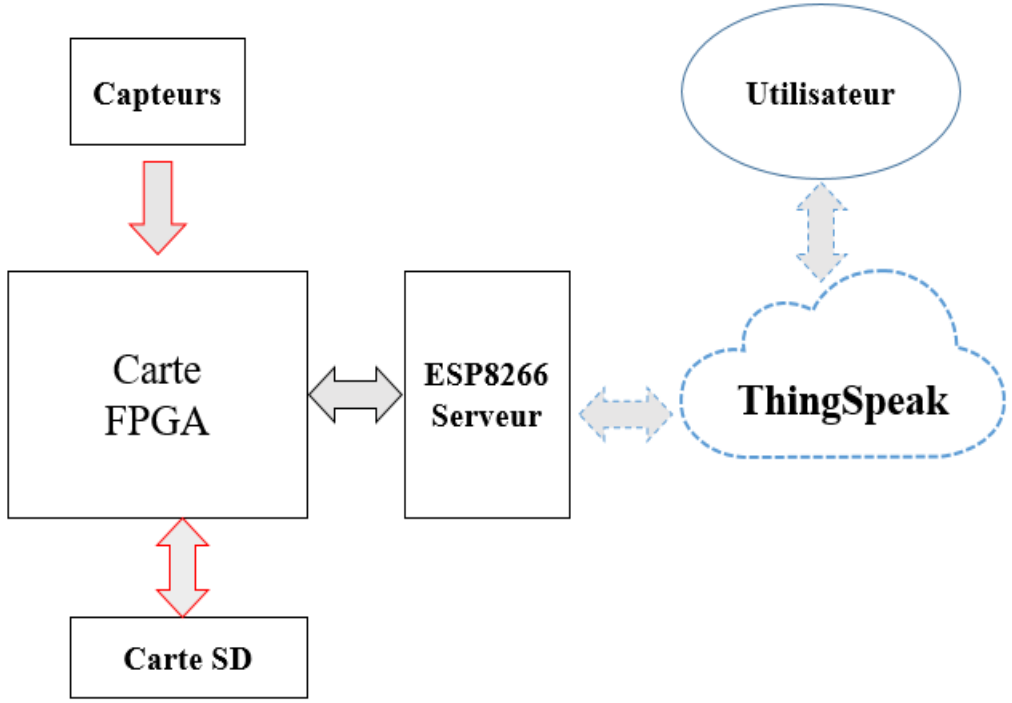


✓ **Communication entre Arduino Raspberry via une liaison série**

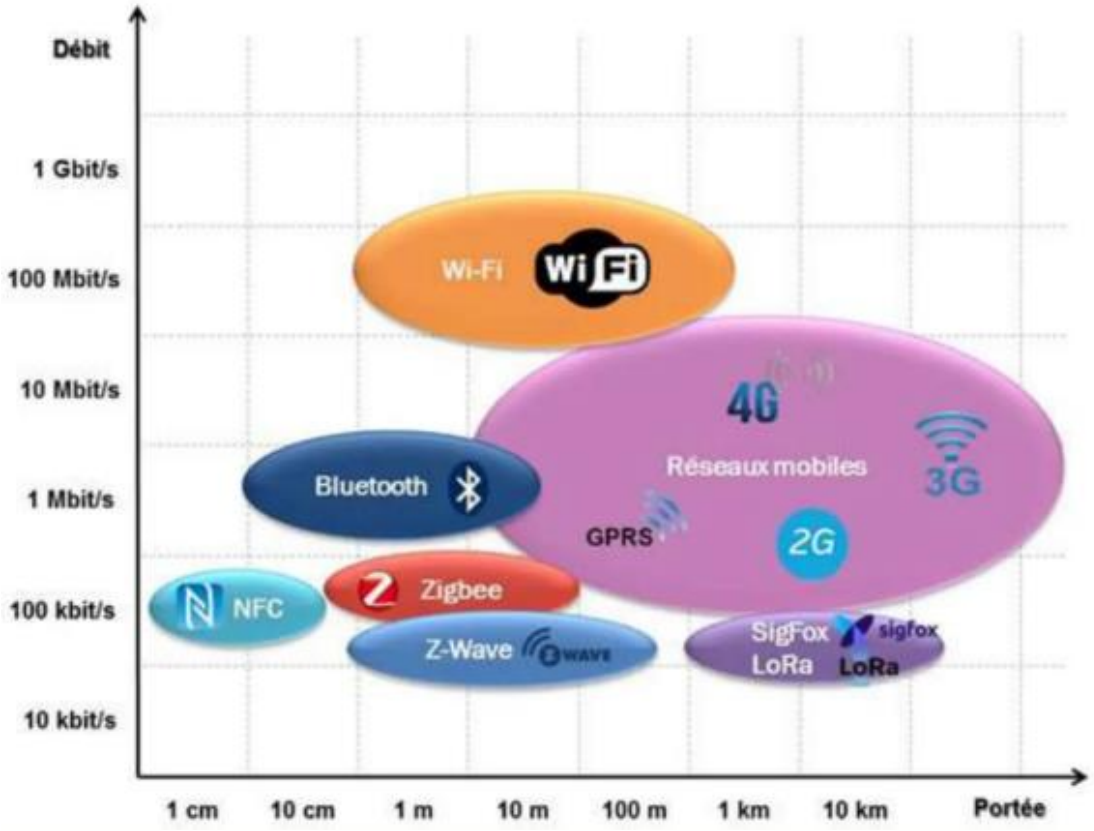


✓ **Communication entre Arduino, et SmartPhone via une liaison Bluetooth**

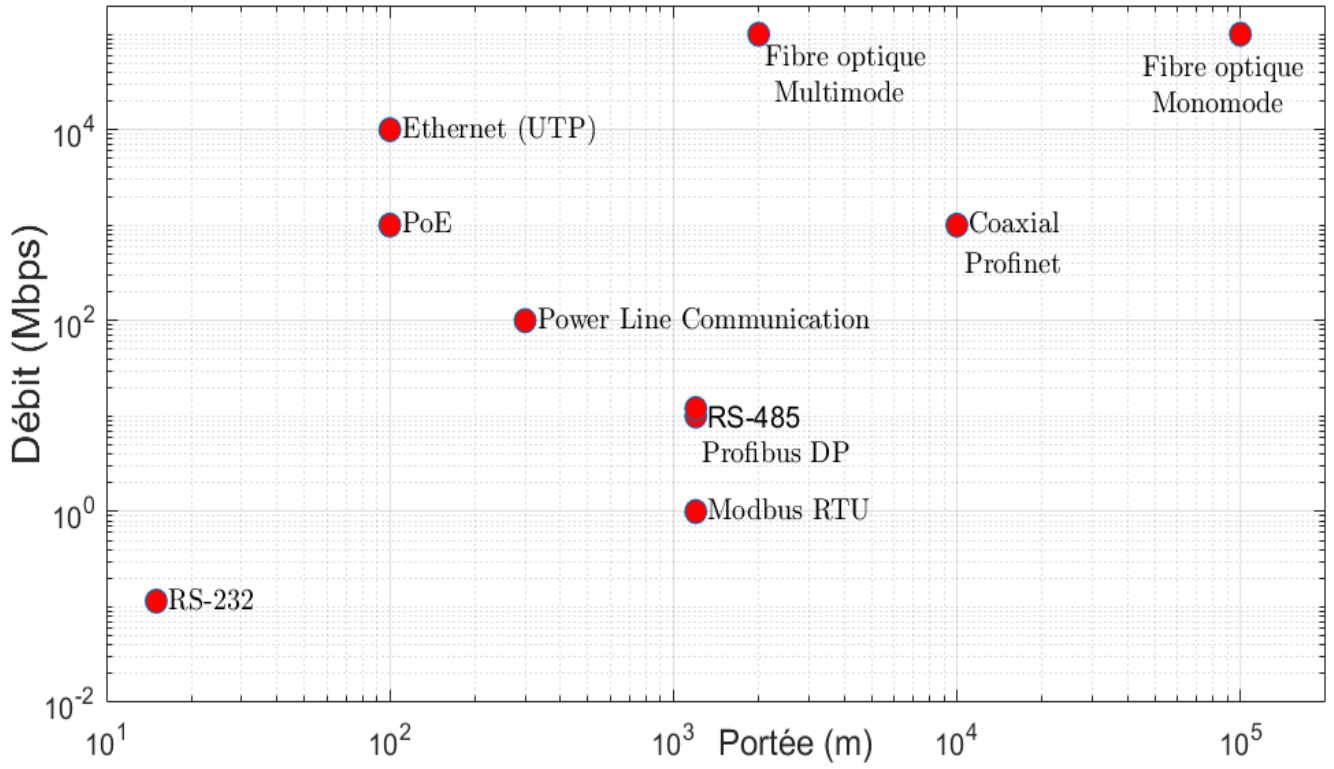
✓ **Objet connecté à l'internet à base d'une carte FPGA et un module Wi-Fi ESP8266**



Interfaces de communication dans les systèmes embarqués



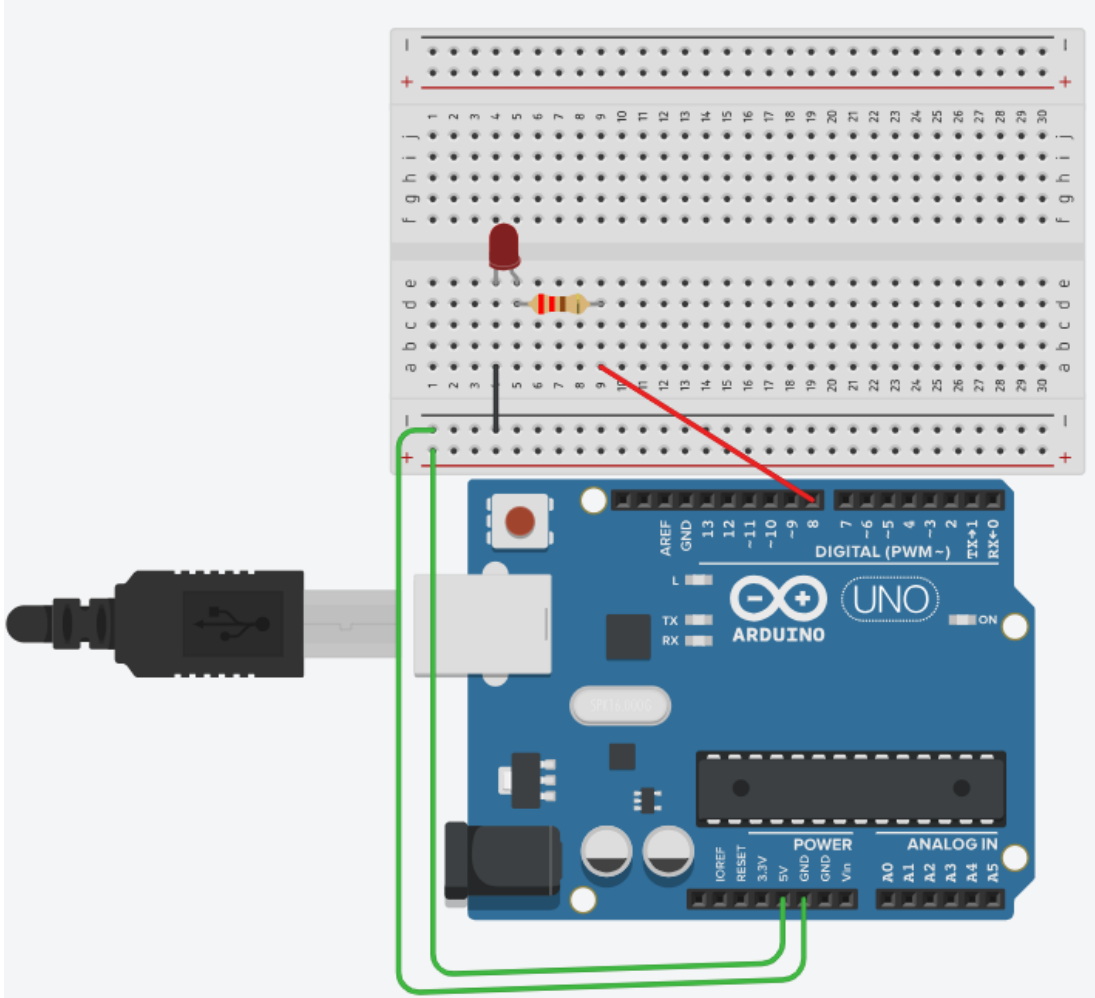
- ✓ **Courte portée et faible débit** : NFC, Zigbee et Z-Wave…
- ✓ **Portée moyenne et débit modéré** : Bluetooth et Wi-Fi
- ✓ **Longue portée et haut débit** : Les réseaux mobiles (2G, 3G, 4G) et technologies IoT (LoRa et SigFox)



- ✓ **Haut débit et courte portée** : Technologies comme **Ethernet et Fibre Optique**
- ✓ **Portée myenne et débit modéré** : Protocoles industriels comme **Profinet et Modbus RTU**.
- ✓ **Longue portée et faible débit** : Standards comme **RS-232 et RS-485**

Exemple 1: Clignotement d'une Led en utilisant Arduino

Partie matérielle: Circuit électronique

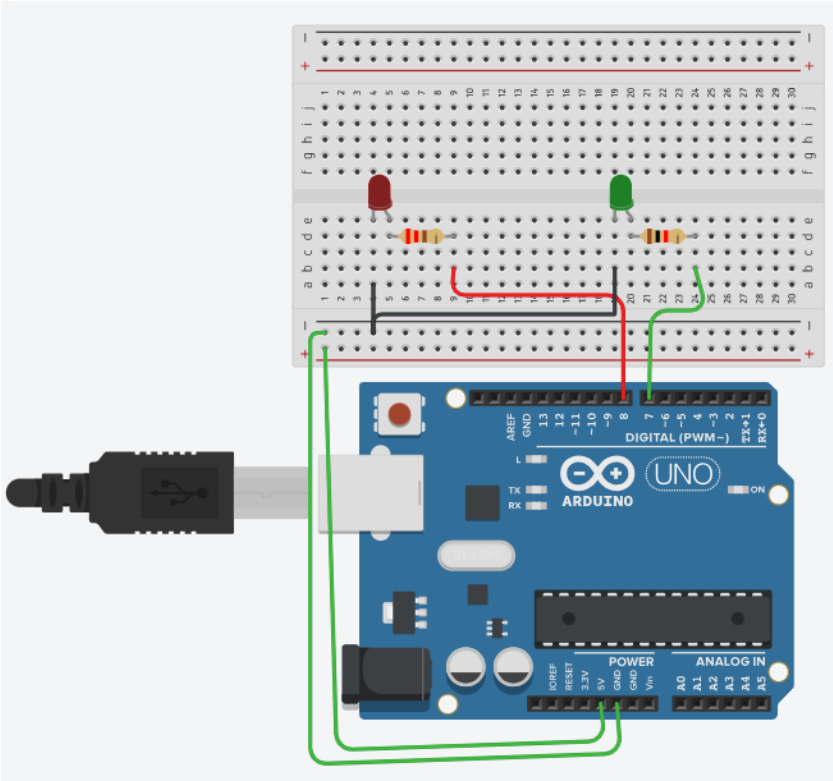


Partie logicielle: code Arduino

```
int ledPin=8; //definition digital 8 pins as pin to control the LED
void setup()
{
   pinMode(ledPin,OUTPUT);    //Set the digital 8 port mode,
OUTPUT: Output mode
}
void loop()
{
   digitalWrite(ledPin,HIGH); //HIGH is set to about 5V PIN8
   delay(1000);               //Set the delay time, 1000 = 1S
   digitalWrite(ledPin,LOW);  //LOW is set to about 5V PIN8
   delay(1000);               //Set the delay time, 1000 = 1S
}
```

Exemple 1: Clignotement d'une Led en utilisant Arduino

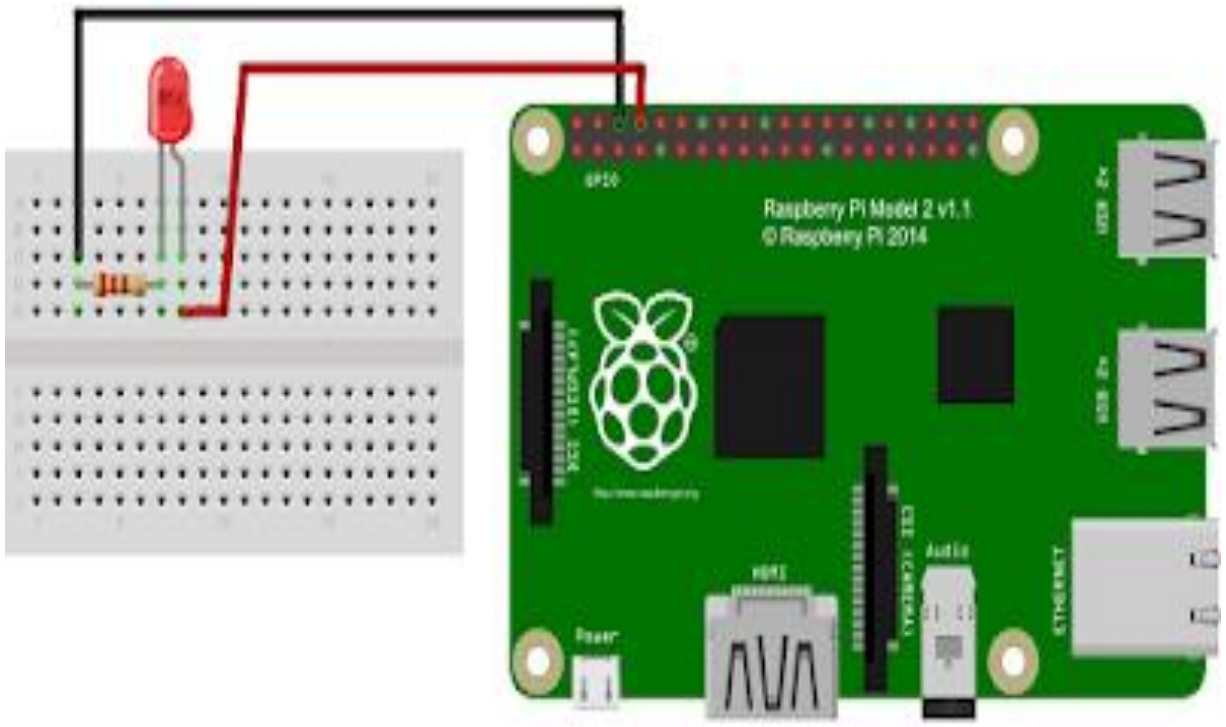
Modifier le programme précédent de manière à ce qu'il correspond au montage suivant




```
int ledPin=8; //definition digital 8 pins as pin to control the LED
void setup()
{
   pinMode(ledPin,OUTPUT);    //Set the digital 8 port mode,
OUTPUT: Output mode
}
void loop()
{
   digitalWrite(ledPin,HIGH); //HIGH is set to about 5V PIN8
   delay(1000);               //Set the delay time, 1000 = 1S
   digitalWrite(ledPin,LOW);  //LOW is set to about 5V PIN8
   delay(1000);               //Set the delay time, 1000 = 1S
}
```

Exemple 2: Clignotement d'une Led en utilisant Raspberry Pi

Partie matérielle: Circuit électronique



Partie logicielle: code Python

```python
import RPi.GPIO as GPIO
import time

# Définition de la broche GPIO utilisée
LED_PIN = 14  # Correspond à la pin BCM 14

# Configuration de la bibliothèque GPIO
GPIO.setmode(GPIO.BCM)  # Utilisation du mode BCM
GPIO.setup(LED_PIN, GPIO.OUT)  # Configuration de la pin en sortie

try:
    while True:
        GPIO.output(LED_PIN, GPIO.HIGH)  # Allumer la LED
        time.sleep(1)  # Pause de 1 seconde
        GPIO.output(LED_PIN, GPIO.LOW)  # Éteindre la LED
        time.sleep(1)  # Pause de 1 seconde

except KeyboardInterrupt:
    print("\nArrêt du programme")
    GPIO.cleanup()  # Nettoyer les GPIO à la fin
```

Exemple 2: Clignotement d'une Led en utilisant Raspberry Pi

Exercice:

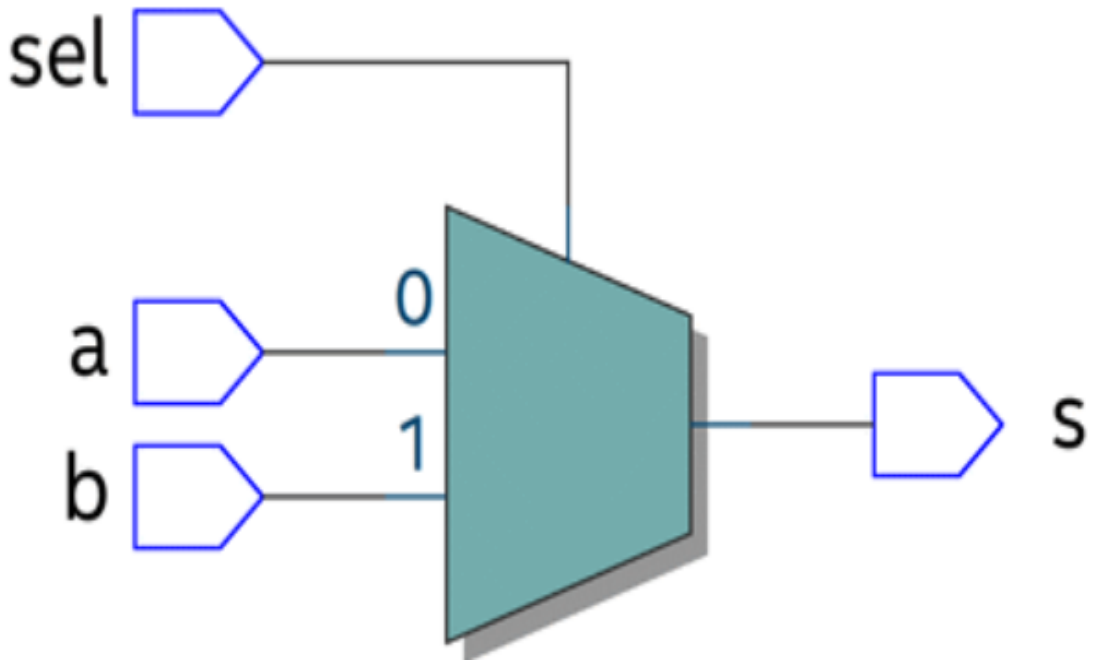
Modifier le programme ci-contre de manière à réaliser le Clignotement successif de trois leds. Ces Leds seront notées LedRouge, LedVerte, LedJaune.

```python
import RPi.GPIO as GPIO
import time

# Définition de la broche GPIO utilisée
LED_PIN = 14  # Correspond à la pin BCM 14

# Configuration de la bibliothèque GPIO
GPIO.setmode(GPIO.BCM)  # Utilisation du mode BCM
GPIO.setup(LED_PIN, GPIO.OUT)  # Configuration de la pin en sortie

try:
    while True:
        GPIO.output(LED_PIN, GPIO.HIGH)  # Allumer la LED
        time.sleep(1)  # Pause de 1 seconde
        GPIO.output(LED_PIN, GPIO.LOW)  # Éteindre la LED
        time.sleep(1)  # Pause de 1 seconde

except KeyboardInterrupt:
    print("\nArrêt du programme")
    GPIO.cleanup()  # Nettoyer les GPIO à la fin
```

Exemple 3: Description d'un Mux par VHDL

Niveau structurel

Niveau fonctionnel ➔




```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Mux2to1 is
    Port ( a    : in  STD_LOGIC;
           b    : in  STD_LOGIC;
           Sel  : in  STD_LOGIC;
           s    : out STD_LOGIC);
end Mux2to1;

architecture Behavioral of Mux2to1 is
begin
    process(a, b, Sel)
    begin
        if (Sel = '0') then
            s <= a;  -- Si Sel est 0, la sortie est a
        else
            s <= b;  -- Si Sel est 1, la sortie est b
        end if;
    end process;
end Behavioral;
```