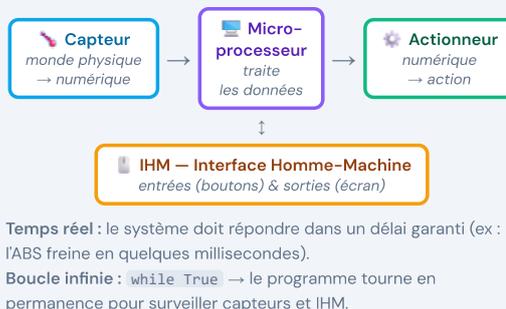


## Définitions essentielles

- Système embarqué** : Système informatique intégré dans un objet, dédié à une tâche précise, miniaturisé, souvent soumis à des contraintes de **temps réel**.
- Capteur** : Mesure une grandeur physique et la convertit en signal numérique. Ex : *thermomètre, GPS, caméra, accéléromètre*
- Actionneur** : Reçoit un signal numérique et produit une action physique. Ex : *moteur, LED, haut-parleur, écran*
- IHM** : **Interface Homme-Machine** : permet à un humain d'interagir avec le système (entrées : boutons, écran tactile ; sorties : affichage, son).
- Objet connecté** : Objet embarqué qui **communique** via un réseau (Wi-Fi, Bluetooth, Internet). L'ensemble forme l'IIoT (Internet of Things).

## Fonctionnement d'un système embarqué



## La carte micro:bit – Composants

Composant	Type	Rôle
Thermomètre	Capteur	Mesure la température
Accéléromètre	Capteur	Mouvements, secousse, inclinaison
Boussole	Capteur	Orientation magnétique
Boutons A et B	IHM entrée	Interaction utilisateur
Matrice 5x5 LEDs	Actionneur IHM sortie	Affiche texte, chiffres, images
Haut-parleur (v2)	Actionneur	Émet des sons
Bluetooth / Radio	Communication	Échange avec d'autres appareils

💡 La matrice LED est à la fois *actionneur* (produit de la lumière) et *IHM sortie* (informe l'utilisateur). Un même composant peut jouer plusieurs rôles !

## Code MicroPython – À connaître

```

from microbit import *
import random

# Structure de base – boucle infinie
while True:
    # Lecture d'un capteur
    t = temperature() # en °C

    # Boutons (IHM entrée)
    if button_a.was_pressed():
        display.scroll(str(t) + "°C")
    elif button_b.was_pressed():
        display.show(Image.HAPPY)

    # Conditionner selon le capteur
    if t > 25:
        display.show(Image.ANGRY)
    else:
        display.show(Image.HAPPY)

    sleep(200) # pause en millisecondes
  
```

⚠ Écrire `t = temperature()` et non `temperature = temperature()` (conflit de nom !)

⚠ Préférer `was_pressed()` à `is_pressed()` pour un appui unique.

## Sûreté & Sécurité

- Sûreté** : *Safety* : le système fonctionne **sans erreur**. Un bug peut être dangereux (avion, métro, voiture).
  - Sécurité** : *Security* : le système résiste aux **attaques informatiques**.
  - Botnet** : Réseau d'objets connectés piratés utilisés pour lancer des attaques (ex : Mirai 2016).
- ⚠ Utiliser **mots de passe par défaut** = risque de piratage
- ⚠ Pas de mise à jour = failles non corrigées
- ✓ Changer le mot de passe dès l'installation
  - ✓ Mettre à jour régulièrement le firmware

## Objets connectés – Pour & Contre

✓ Avantages	⚠ Risques
Confort & contrôle à distance	Piratage & vol de données
Économies d'énergie	Dépendance au réseau
Sécurité améliorée (alarmes)	Collecte de données personnelles
Automatisation des tâches	Consommation électrique accrue
Suivi médical à distance	Obsolescence rapide

## Repères

Année	Événement
1967	Mission Apollo (guidage)
1971	1er microprocesseur Intel
1984	Airbus A320 (commandes électriques)
1998	Météor — métro sans conducteur
1999	Expression « IoT » inventée
2007	Arrivée du smartphone
2020	~50 milliards d'objets connectés