

# Informatique Embarquée & Objets Connectés

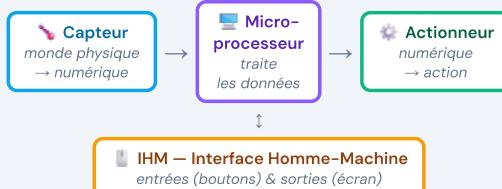
SNT · SECONDE

Fiche de synthèse — SNT Seconde | À connaître pour l'évaluation

## ■ Définitions essentielles

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

## ■ Fonctionnement d'un système embarqué



**Temps réel** : le système doit répondre dans un délai garanti (ex : l'ABS freine en quelques millisecondes).

**Boucle infinie** : while True → le programme tourne en permanence pour surveiller capteurs et IHM.

## ■ La carte micro:bit – Composants

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