Périphériques et IHM

Fiche de synthèse

1 Périphériques

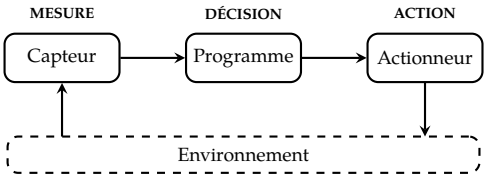
Un **périphérique d'entrée** fait entrer une information dans l'ordinateur (clavier, souris, micro), un **périphérique de sortie** la fait sortir (écran, imprimante, haut-parleur). Un écran tactile fait les deux.

2 Capteurs et actionneurs

Définition 1 **(Capteur)** Un **capteur** mesure une **grandeur physique** (température, lumière, distance, présence, appui) et la transmet au programme sous forme de **nombre**. C'est un périphérique d'entrée.

Définition 2 **(Actionneur)** Un **actionneur** transforme une **commande** du programme en **action physique** : mouvement (moteur), lumière (lampe), son (haut-parleur), chaleur (résistance), passage d'eau (électrovanne). C'est un périphérique de sortie.

3 Mesure, décision, action



L'action modifie la grandeur mesurée : la chaîne se répète **sans fin**. Elle tourne dans un **système embarqué** (ordinateur intégré à un appareil), un **objet connecté** (relié à un réseau) ou un **robot**.

Capteur	Décision	Actionneur
Sonde de température	$t <$ consigne ?	Radiateur
Luminosité, mouvement	sombre et présence ?	Lampes
Distance	obstacle proche ?	Moteurs

4 La chaîne en Python

On simule capteur et actionneur par des fonctions ; la décision est une fonction testable. Le programme principal répète mesure, décision, action dans une boucle.

```
import random

def lire_temperature():
    return random.randint(15, 25)

def decider(temperature, consigne):
    return temperature < consigne

assert decider(17, 19) == True
assert decider(19, 19) == False
```

5 Interface homme-machine

Définition 3 **(IHM)** L'**IHM** est l'ensemble des moyens par lesquels un utilisateur et une machine échangent des informations : périphériques et programme. On la conçoit à partir d'un **cahier des charges** : ce qu'elle affiche, les actions possibles, les réponses du programme.

En **programmation événementielle**, le programme crée ses composants, associe une fonction à chaque action, puis attend dans la **boucle d'événements** : un clic est un **événement**, qui appelle la fonction associée.

```
import tkinter as tk

def basculer():
    if etat.cget("text") == "Éteinte":
        etat.configure(text="Allumée")
    else:
        etat.configure(text="Éteinte")

fenetre = tk.Tk()
etat = tk.Label(fenetre, text="Éteinte")
etat.pack()
bouton = tk.Button(fenetre, text="Lampe",
                   command=basculer)
bouton.pack()
fenetre.mainloop()
```

`Label` affiche (sortie), `Button` et `Scale` (curseur) font entrer une action ; `configure` modifie un texte, `get` lit un curseur. On écrit `command=basculer` **sans parenthèses**.

À retenir.

- Capteur : grandeur physique vers nombre ; actionneur : commande vers action physique.
- Un système réactif répète sans fin la chaîne mesure, décision, action.
- La décision s'écrit dans une fonction, testée avec `assert`.
- Une IHM répond à un cahier des charges : composants d'entrée et de sortie, fonctions appelées lors des événements.