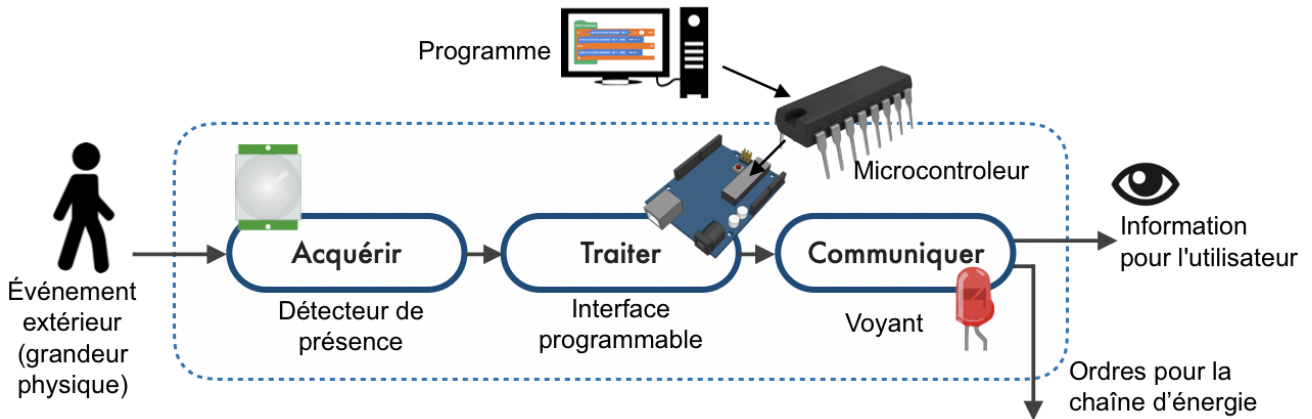




SFC 3.1	Entrées ou sorties d'un programme (données issues par exemple de capteurs IHM et sorties pouvant être en lien avec un actionneur, fichiers) ;
CCRI 3.1	La modularité : sous-programme, fonction La structuration d'un programme (organisation, modularité, commentaires)

### Entrées et sorties d'un programme

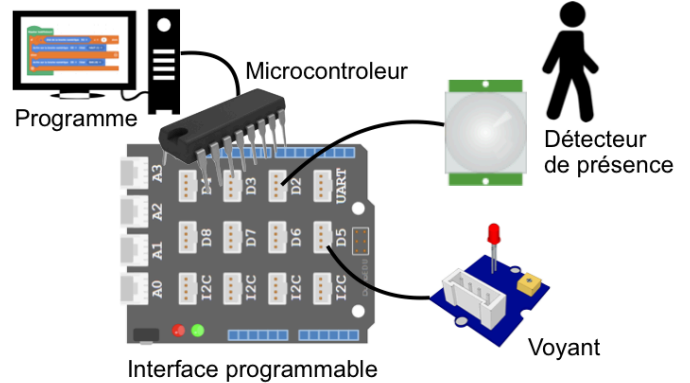
Dans un système technique, le programme est enregistré dans une interface programmable (par l'intermédiaire du microcontrôleur). L'interface permet d'être connectée à des capteurs (en entrée) ou des actionneurs (en sortie) par l'intermédiaire de broches numérotées.



**Exemple :** Le système permet d'allumer un voyant (actionneur) dès la détection d'une présence (capteur). Ici le détecteur de présence est branché sur la broche D2, le voyant sur la broche D5

```

Répéter indéfiniment
si état de la broche numérique D2 = HAUT (1) alors
  écrire sur la broche numérique D5 l'état HAUT (1)
sinon
  écrire sur la broche numérique D5 l'état BAS (0)
  
```



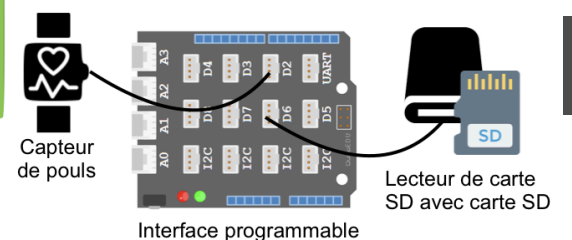
### Entrées ou sorties d'un programme sur un fichier

Il est possible via une interface programmable de traiter ou d'enregistrer des données stockées sur un fichier.

```

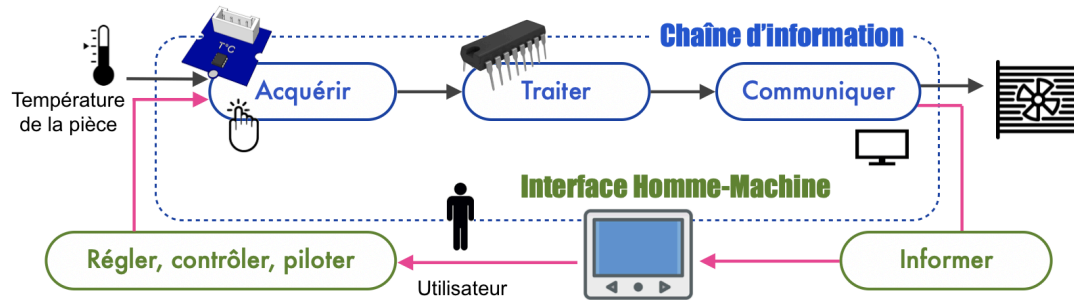
Répéter indéfiniment
affecter à Pulsation la valeur [Détecteur de pouls] valeur sur la broche A2
[Module SD SPI] écrire dans la carte SD
sur la broche CS D6
Données créer le texte valeur du chronomètre en (s) Pulsation
attendre 2 seconde(s)
Séparateur de données par ";"
  
```

**Exemple :** Enregistrement de la pulsation du pouls dans une carte SD toutes les 2 secondes.



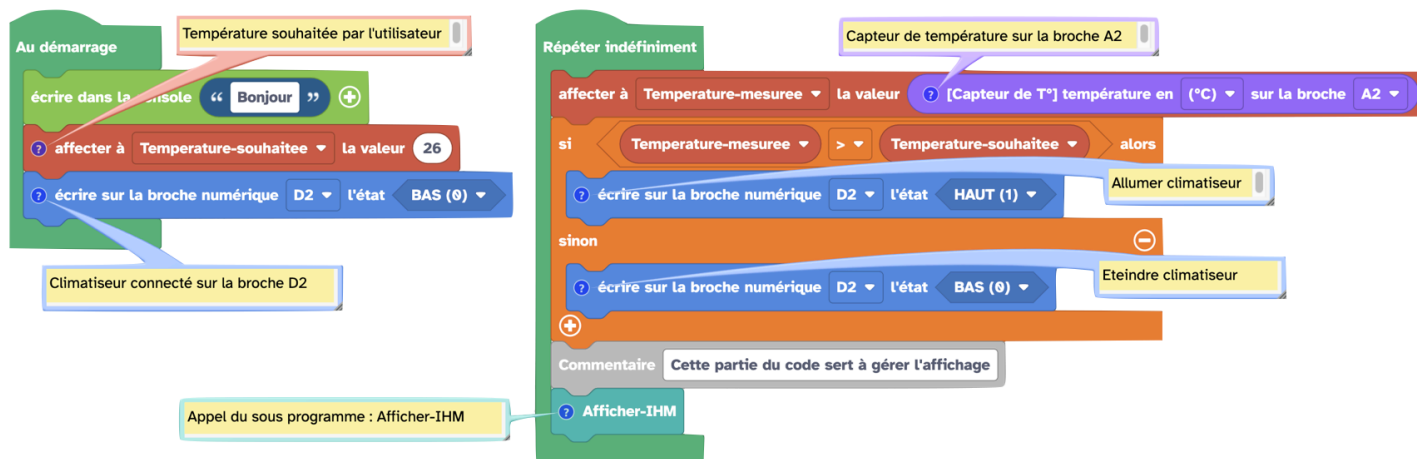
## IHM : Interface Homme Machine (exemple)

Sur cette page, le système de climatisation mesure la température de la pièce grâce à un capteur de température. La température souhaitée par l'utilisateur est de 26°C. Si la température mesurée est supérieure à 26°C alors la climatisation s'allume, sinon elle s'éteint avant de mesurer à nouveau la température. Le sous programme "Afficher-IHM" permet de faire un retour visuel à l'utilisateur.



## Structuration d'un programme : Initialisation - Programme principal - Commentaires

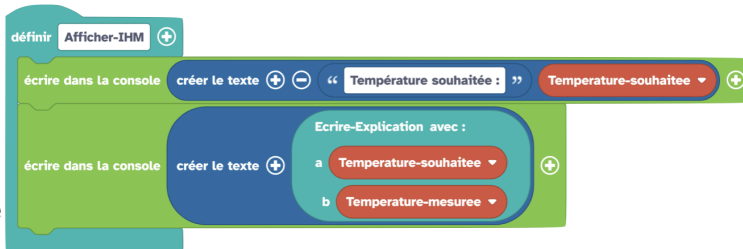
Il est important de structurer un programme afin de faciliter sa lecture, ou sa modification. Il faut différencier la partie dédiée à l'**initialisation** (bloc "Au démarrage", qui se déroule qu'une seule fois lors du lancement du programme) à la **partie du programme principal** (bloc "Répéter indéfiniment"). Il est important également d'ajouter des **commentaires** (en superposition du programme ou directement dans le programme) afin de rendre le programme plus facile à comprendre. Les commentaires sont ignorés par le logiciel.



## Modularité : Sous-programme - Fonction

L'utilisation des **sous-programmes** permet de limiter les instructions dans le programme principal et donc de le rendre plus lisible. Travailler avec des sous-programmes permet de dédier et valider les tâches d'un programme.

Exemple ici avec l'appel du sous-programme "Afficher-IHM" :



L'appel à des **fonctions** dans un programme permet de limiter les instructions répétitives. A la différence des sous-programmes, une fonction renvoie une valeur. Dans cet exemple, la fonction "Ecrire-Explication" retourne un texte en fonction de la température la plus élevée.

