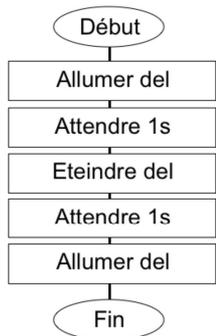
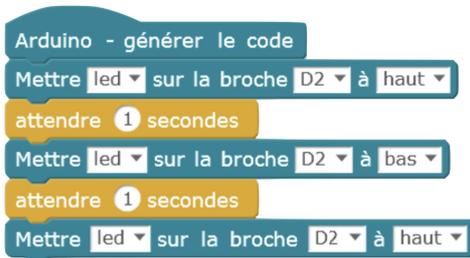


### Algorithme/algorithme et Programme : séquences d'instructions



Un **programme** informatique est une suite d'instructions déterminées par l'Informaticien pour répondre à un problème (jeux, application, système réel, ...). Il est mis au point, testé puis corrigé avant d'être mémorisé puis traité par un **microcontrôleur** (ou un microprocesseur).

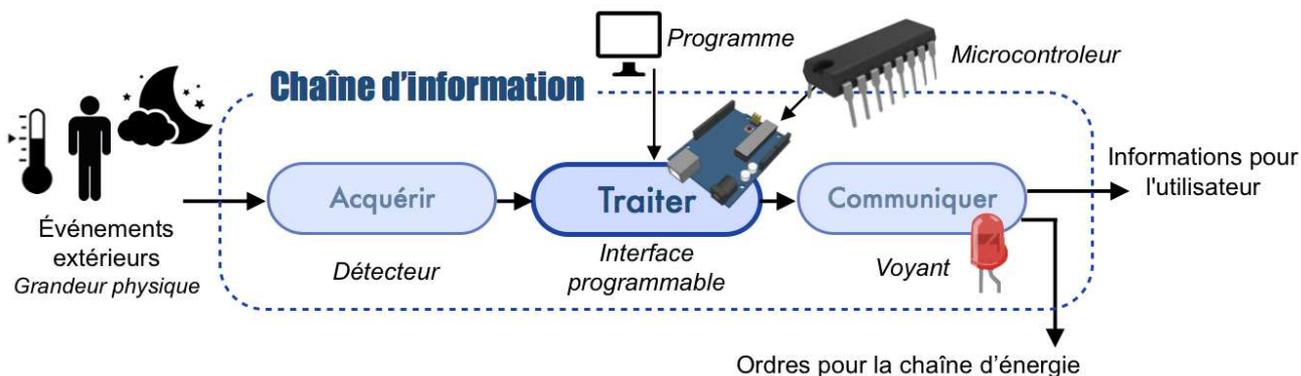
Le code sera ensuite traduit en langage compréhensible par le microprocesseur sous forme de « 0 » et « 1 » : le code **binaire**.

| Description du programme   |  | Programme  |  |
|--|--|--|--|
| Algorithme   | Algorithme   | Langage graphique  | Code   |
|  | Début :<br>Allumer la DEL sortie 2<br>Attendre 1 seconde<br>Eteindre la DEL sortie 2<br>Attendre 1 seconde<br>Allumer la DEL sortie 2<br>Fin |  | <pre>void setup() {   pinMode(2,OUTPUT);   digitalWrite(2,1);   delay(1000*1);   pinMode(2,OUTPUT);   digitalWrite(2,0);   delay(1000*1);   pinMode(2,OUTPUT);   digitalWrite(2,1); };</pre> |

### Chaîne d'information



C'est dans le bloc **Traiter** de la **chaîne d'information** que les informations sont traitées en fonction des **instructions du programme**. Le programme étant enregistré dans le microcontrôleur.



### Boucles



Lorsque des instructions sont répétées, on utilise des **boucles** pour optimiser le programme.

**Exemples de boucles :**

Répéter indéfiniment, Répéter x fois, Répéter jusqu'à ...



## Variable informatique

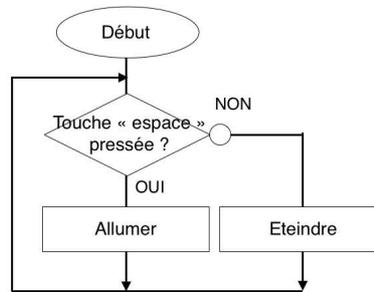


Une **variable** est une donnée (une information) associée à un nom. Elle est mémorisée/stockée et elle peut changer de valeur en fonction des instructions du programme.

Exemple : variable « compteur »



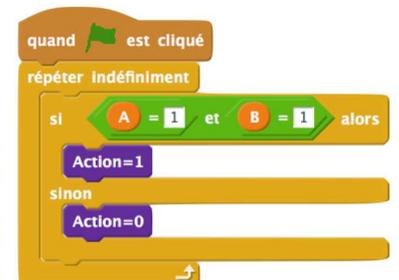
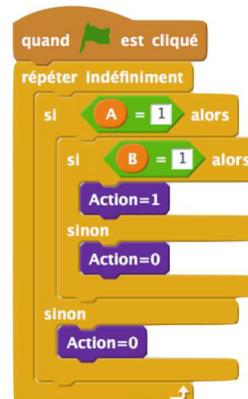
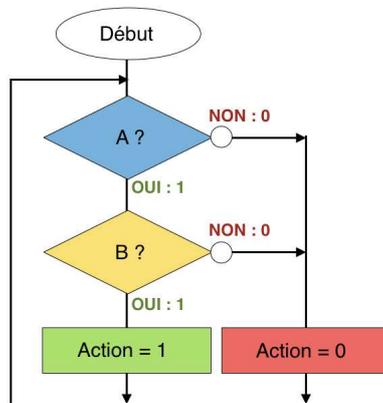
## Déclenchement d'une action par un événement, instructions conditionnelles



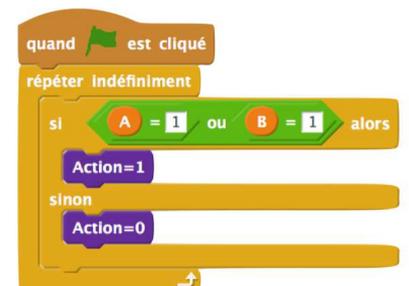
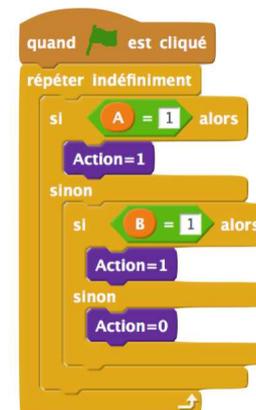
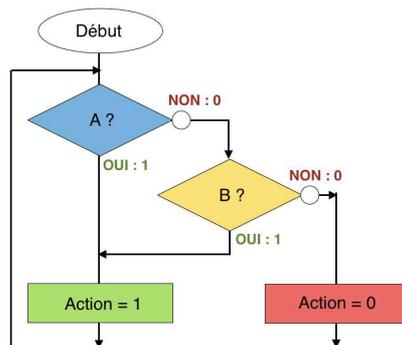
## Déclenchement d'une action par une fonction ET, une fonction OU



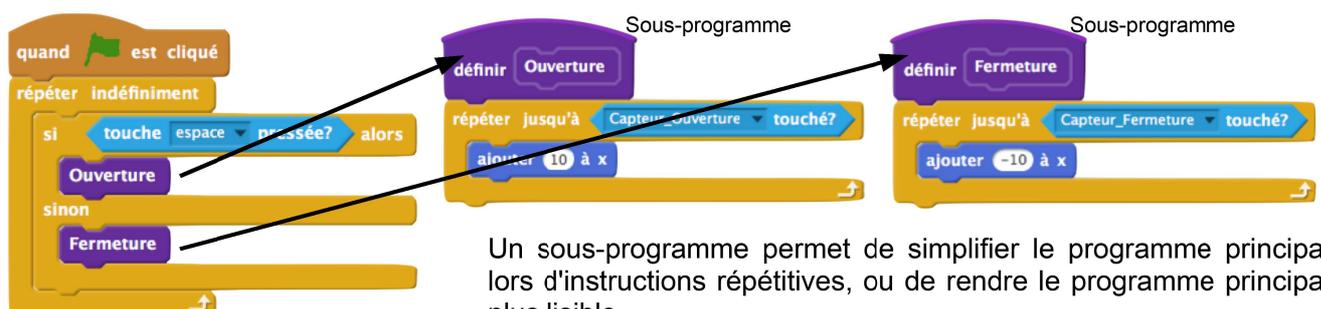
| Fonction ET |     |        |
|-------------|-----|--------|
| A ?         | B ? | Action |
| 0           | 0   | 0      |
| 0           | 1   | 0      |
| 1           | 0   | 0      |
| 1           | 1   | 1      |



| Fonction OU |     |        |
|-------------|-----|--------|
| A ?         | B ? | Action |
| 0           | 0   | 0      |
| 0           | 1   | 1      |
| 1           | 0   | 1      |
| 1           | 1   | 1      |



## Structure d'un programme avec des sous-programmes



Un sous-programme permet de simplifier le programme principal lors d'instructions répétitives, ou de rendre le programme principal plus lisible.