



| | |
|-------------------------------------|---|
| CT 1.3 – CT 2.5 – CT 2.7 DIC 1.5 | Imaginer des solutions pour produire des objets et des éléments de programmes informatiques en réponse au besoin. |
| CT 3.1 OTSCIS 2.1 | Exprimer sa pensée à l'aide d'outils de description adaptés : croquis, schémas, graphes, diagrammes, tableaux. |
| CT 4.2 – CT 5.5 IP 2.3 | Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs. |

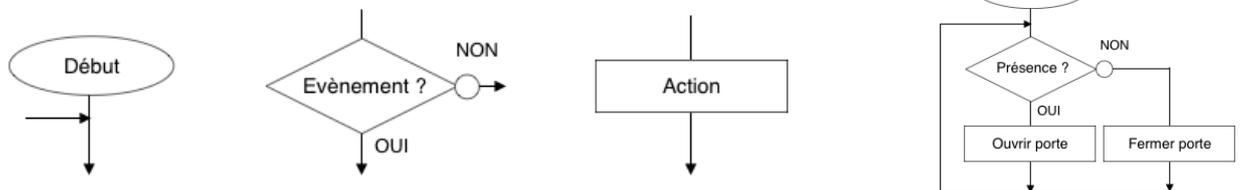
Symboles de base



Un algorithme est une suite d'instructions précises et structurées qui décrit la manière dont on résout un problème.

Cette description peut être textuelle (si, alors, sinon, tant que ...) ou graphique (appelé également organigramme ou logigramme).

Dans ce cas des normes d'écritures sont à respecter :



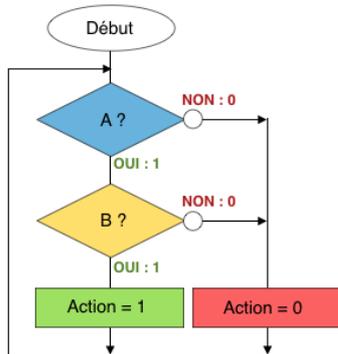
Fonctions ET et OU

L'utilisation des fonctions ET et OU sont essentielles pour présenter correctement une solution.



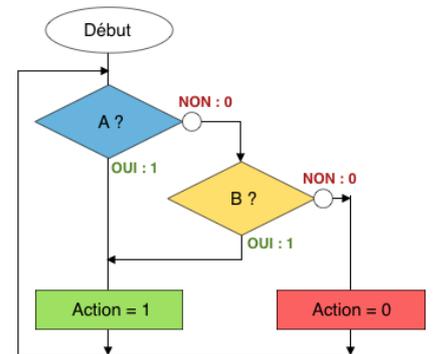
Fonction ET

| A ? | B ? | Sortie |
|-----|-----|--------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



Fonction OU

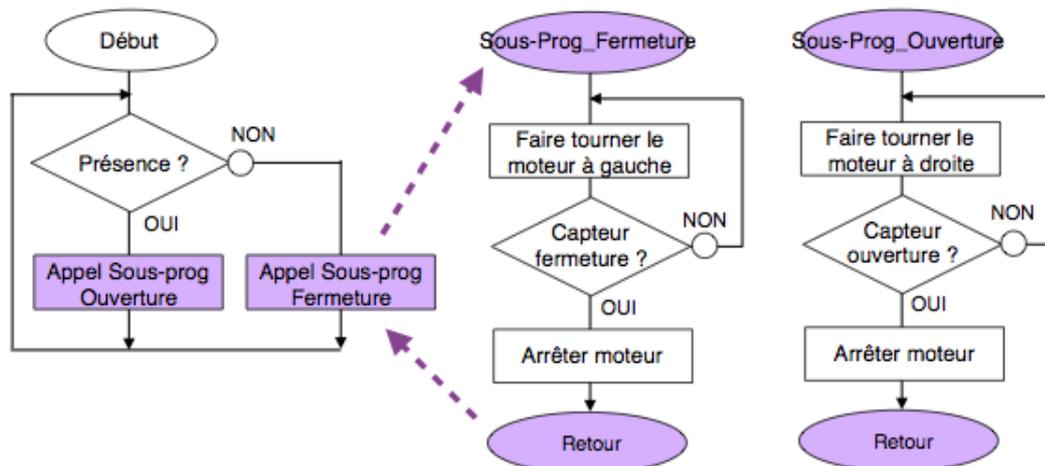
| A ? | B ? | Sortie |
|-----|-----|--------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



Algorithme et gestion des sous-problèmes



L'utilisation des sous-problèmes est idéale pour une meilleure lisibilité, pour alléger l'algorithme lors de succession d'actions identiques, pour faciliter le travail en collaboration, pour faciliter une recherche d'erreur (test individuel des sous-problèmes).





CT4.2, CT5.5
IP2.3

Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.

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).

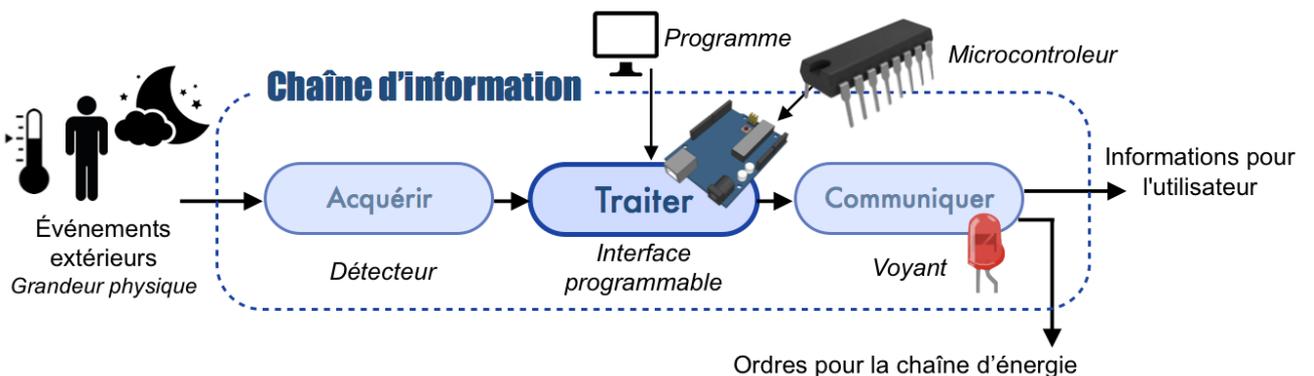
Le code sera ensuite traduit en langage compréhensible par le microprocesseur sous forme de « 0 » et « 1 » : le code **binaire**.

| Description du programme | | Programme | |
|--------------------------|--|-------------------|--|
| Algorithme | Algorithme | Langage graphique | Code |
| | Début : Allumer la DEL sortie 2 Attendre 1 seconde Eteindre la DEL sortie 2 Attendre 1 seconde Allumer la DEL sortie 2 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.



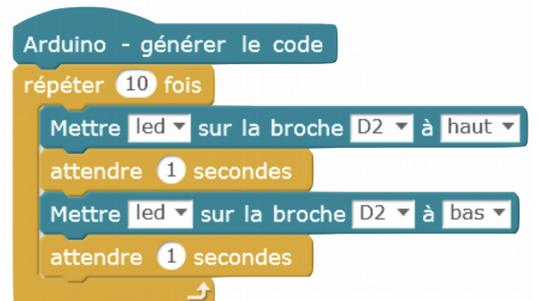
Boucles



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.

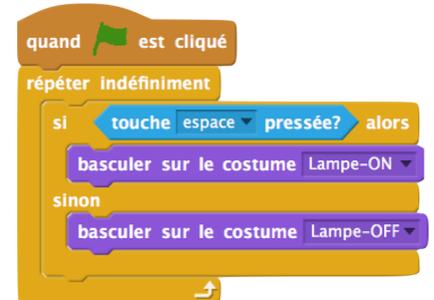
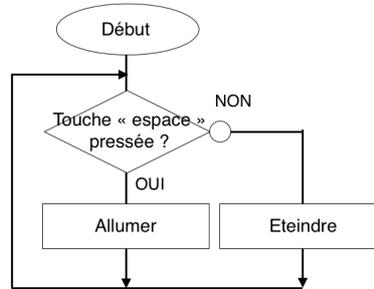
Exemple : variable « compteur »



Déclenchement d'une action par un événement, instructions conditionnelles



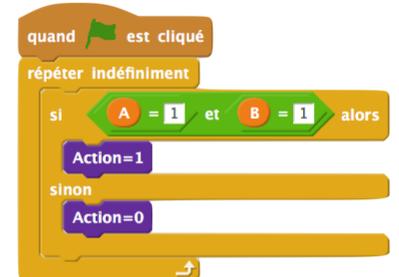
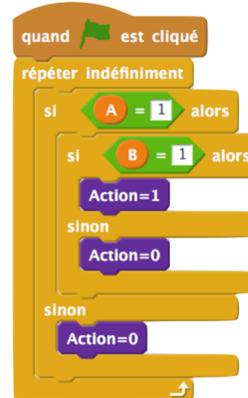
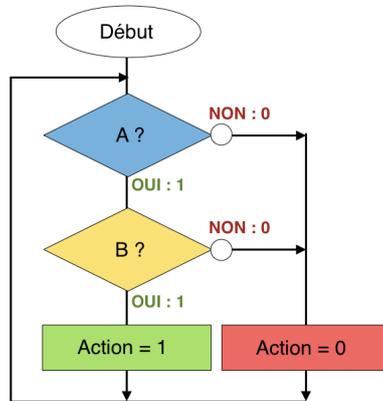
Début
Si touche « espace » pressée
Alors allumer
Sinon éteindre
Fin Si
Retour début



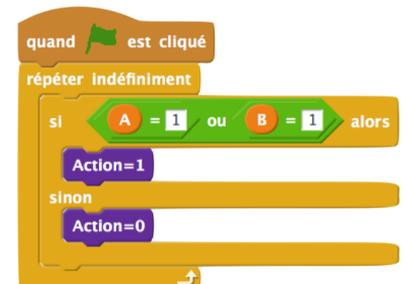
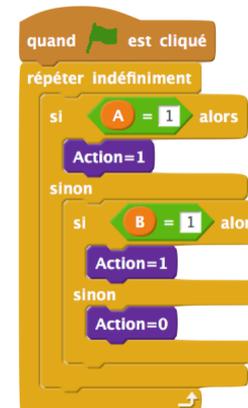
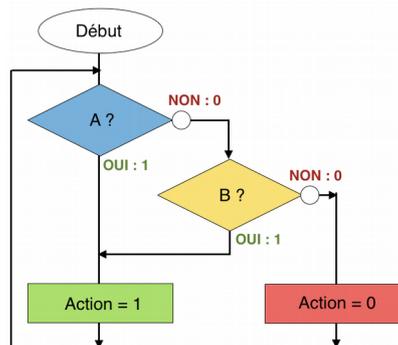
Déclenchement d'une action par une fonction ET, une fonction OU



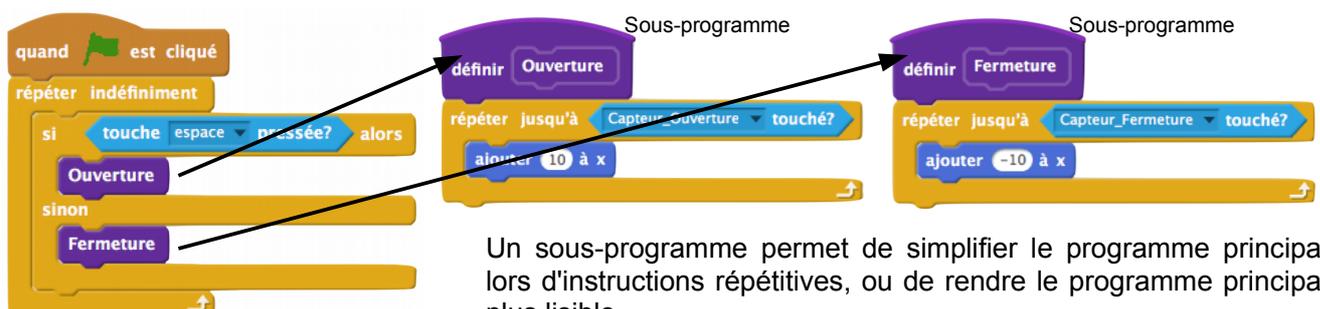
| Fonction ET | | |
|-------------|-----|--------|
| A ? | B ? | Action |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



| 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.

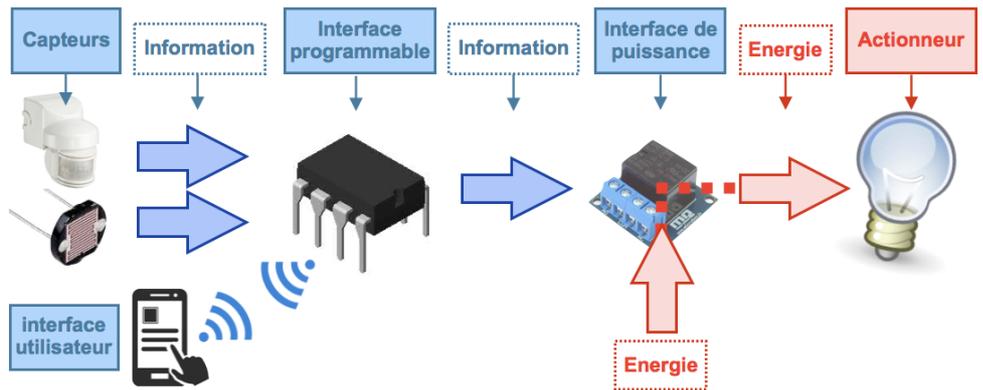
| | | | |
|--|---|---|---|
|  | <p align="center">TECHNOLOGIE</p> <p align="center"><i>Ce que je dois retenir</i></p> | <p align="center">SYSTÈMES EMBARQUÉS</p> <p align="center">CAPTEUR, ACTIONNEUR, INTERFACE</p> | <p align="center">CYCLE</p> <p align="center">4</p> |
| <p>CT 4.2 – CT 5.5 IP 2.3</p> | <p>Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs</p> | | |
| <p>CS 1.6 MSOST 1.4</p> | <p>Identifier les flux d'information sur un objet et décrire les transformations qui s'opèrent.</p> | | |

Capteur, actionneur, interface



Les capteurs permettent d'acquérir des informations qui sont traitées par une interface programmable pour piloter des actionneurs. Souvent, il faut utiliser une interface de puissance pour distribuer l'énergie vers l'actionneur.

Il est aussi possible d'envoyer des informations directement depuis des interfaces utilisateur (ordinateur, appareil nomade, ...) afin de modifier en temps réel le fonctionnement du système embarqué.



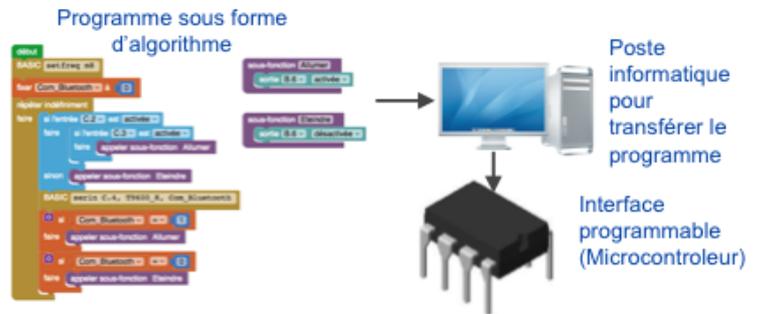
Système embarqué



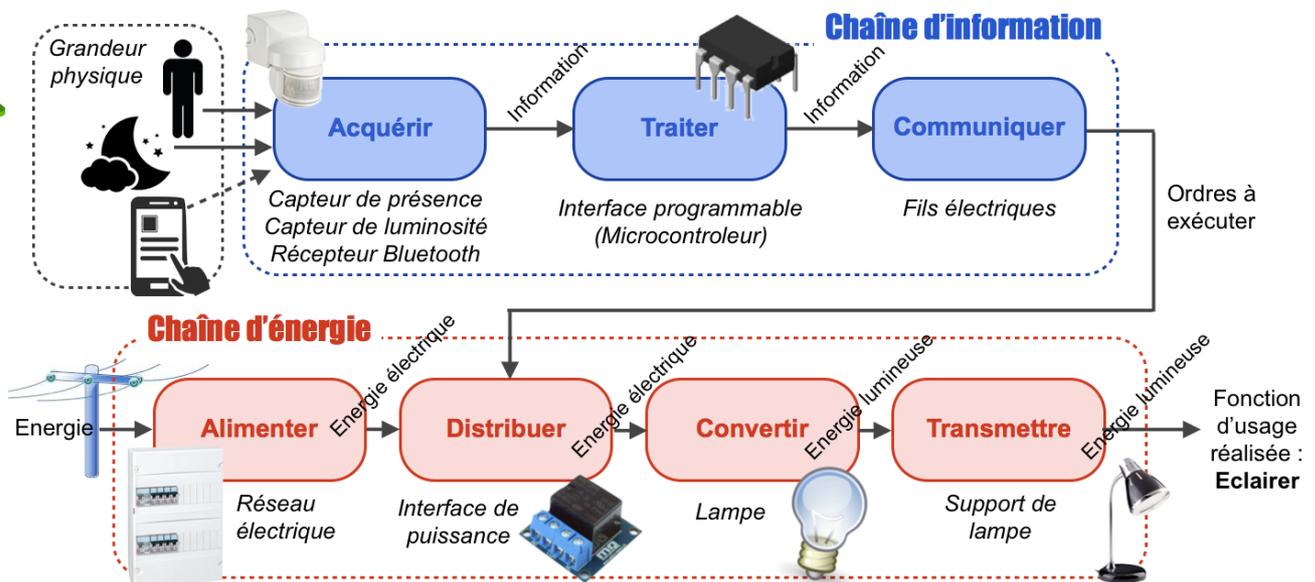
Le système embarqué réagit en fonction de la programmation qui lui est associée et de l'acquisition de grandeurs physiques qu'il reçoit de ses capteurs ou d'une interface utilisateur.

Ainsi le système est autonome dans son environnement et s'adapte correctement si :

- La programmation qui lui est associée prend en compte l'ensemble des scénarios possibles.
- Les capteurs qui lui sont associés lui permettent d'acquérir les informations souhaitées.



Chaîne d'information et chaîne d'énergie / Structure des systèmes





CT 5.5
IP 2.3

Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs

Forme d'un signal

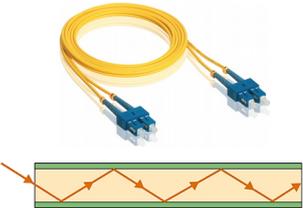
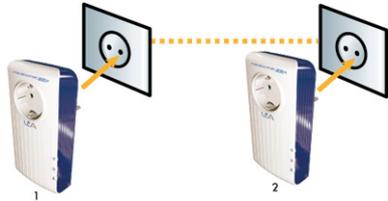


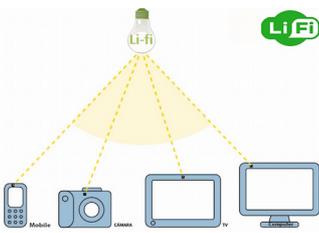
Un signal peut prendre différentes formes, un support de communication permet sa transmission.

| Impulsion électrique | Impulsion lumineuse | Vibration mécanique | Onde |
|----------------------|----------------------|---|---|
| <i>Fil de cuivre</i> | <i>Fibre optique</i> | <i>L'eau pour les dauphins, la peau pour le tambour, la membrane pour les hauts-parleurs, ...</i> | <i>L'air ou l'espace pour les ondes radio et les ondes des satellites</i> |

Transmission d'un signal



| Transmission du signal avec conducteur | | |
|--|---|---|
| Par fil électrique | Par fibre optique | Par courant porteur en ligne (CPL) |
|  <p><i>Transporte une impulsion électrique.</i> <i>Solution la moins coûteuse : souris informatique filaire, cordon d'écouteur, ...</i></p> |  <p><i>Transporte une impulsion lumineuse.</i> <i>Constituée de faisceaux de fibre de verre. Elle permet des communications à très longue distance à la vitesse de la lumière.</i></p> |  <p><i>Transporte une impulsion électrique.</i> <i>La communication se fait par les lignes électriques du réseau de l'habitation. Les boîtiers CPL permettent d'adapter le signal. Cette solution ne permet pas de longues distances car elle ne fonctionne que dans le réseau électrique où elle se trouve.</i></p> |

| Transmission du signal sans conducteur | | | |
|--|--|---|--|
| Par vibration | Par infra-rouge | Par radio (Satellite, 4G, Bluetooth, Wifi) | Par Li-Fi |
|  <p><i>Transporte une vibration mécanique.</i> <i>La vibration de la membrane du haut-parleur est générée électriquement ce qui provoque un son.</i></p> |  <p><i>Transporte une impulsion lumineuse.</i> <i>Solution peu onéreuse pour de courtes distances (10m env.) en l'absence d'obstacle.</i></p> |  <p><i>Transporte une onde.</i> <i>Solution sans fil ou pour traverser des obstacles. Plus l'émetteur est haut, plus le signal va loin : satellite, relais téléphonique 3G/4G, antenne radio FM, ...</i> <i>Le bluetooth et le WiFi sont des transmissions radios.</i> <i>Bluetooth : 10 mètres</i> <i>WiFi : 50 mètres</i> <i>Radio FM : 70 mètres</i></p> |  <p><i>Transporte une impulsion lumineuse.</i> <i>En cours de développement : Lampe qui intègre une communication infra-rouge continue (même lampe éteinte).</i></p> |



CT 1.2
MSOST 1.6

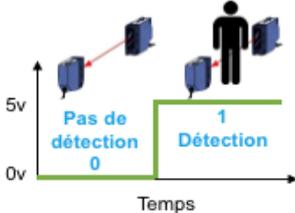
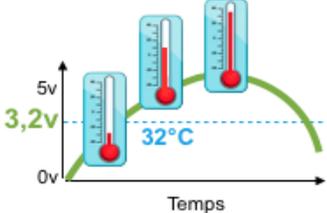
Mesurer des grandeurs de manière directe ou indirecte

Nature d'une information : logique ou analogique



Une **information** peut être **logique** ou **analogique**.

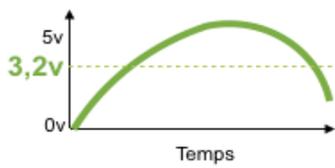
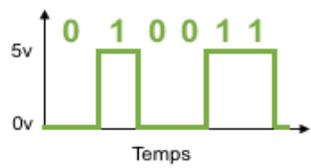
Le choix du capteur sera déterminant pour interpréter l'information souhaitée.

| Exemple de capteur | Signal fournie par le capteur | Information interprétée |
|---|--|---|
|  <p>Barrière infrarouge</p> |  | <p>Détection ou pas de passage</p> <p>Information type LOGIQUE 2 valeurs possibles (tout ou rien)</p> |
|  <p>Capteur de température</p> |  | <p>Température en degrés</p> <p>Information type ANALOGIQUE Plusieurs valeurs possibles</p> |

Nature d'un signal : Analogique ou numérique



Un capteur fournit un **signal** de type **Analogique** ou **numérique**.

| Signal Analogique | Signal numérique |
|---|---|
|  |  |
| <p>Souvent un signal analogique évolue en tension (volt) Exemple : 3,2 volts – Capteur de température</p> | <p>Un signal numérique est une suite de 0 et de 1 Exemple : 010011 - « Capteur » Ultrason</p> |

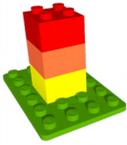
Un signal analogique doit être convertie en numérique pour pouvoir être traiter par le microcontrôleur. C'est la numérisation du signal.

Principe de fonctionnement d'un détecteur, capteur, codeur



| Type de capteur | Exemple | Information | Exemple | Signal |
|-----------------|-----------|-------------------|---------------------------------|-------------------|
| Détecteur | 1 ou 0 | Logique | Détection ou pas (tout ou rien) | Numérique |
| Capteur | 3,2 volts | Analogique | Degrés, Lux, ... : 32°C | Analogique |
| Codeur | 010011 | Analogique | Position, ... : 45° | Numérique |

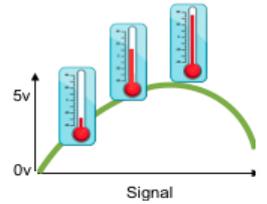
Principe de fonctionnement d'un capteur : numérisation



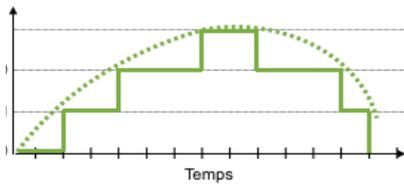
Un signal analogique doit être converti en numérique pour pouvoir être traité par le microcontrôleur (interface programmable) : C'est la numérisation du signal.

Plus la numérisation utilise de bits, meilleure est la précision.

Exemple avec un capteur de température :



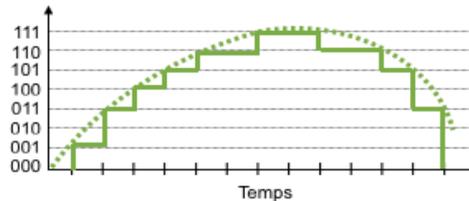
Numérisation sur 2 bits



Soit 4 valeurs possibles : de 0 à 3

| Puissance de 2 | 2 ¹ | 2 ⁰ |
|----------------|----------------|----------------|
| Décimal | 2 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |

Numérisation sur 3 bits

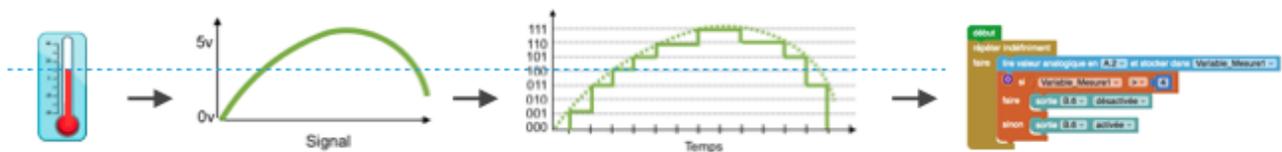


Soit 8 valeurs possibles : de 0 à 7

Exemple :

100 en binaire correspond à 4 en décimal.

| Puissance de 2 | 2 ² | 2 ¹ | 2 ⁰ |
|----------------|----------------|----------------|----------------|
| Décimal | 4 | 2 | 1 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |



Température extérieure

32°C

Acquisition en analogique

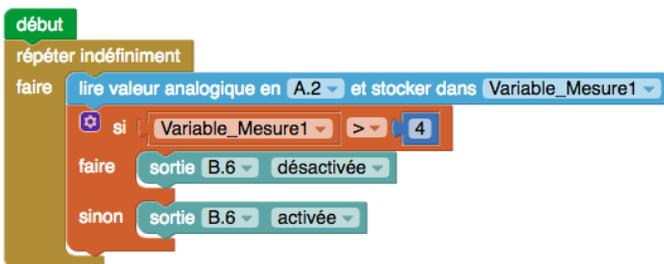
3,2 Volts

Numérisation

Binaire : 100

Traitement

Décimal : 4



Exemple avec le capteur de température qui communique sur l'entrée A2 du microcontrôleur.

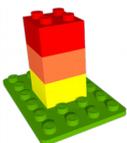
La valeur analogique est enregistrée dans la variable : Variable_Mesure1.

Si la variable > 4 (soit ici par ex 100 en binaire).

La sortie B6 se désactive (arrêt du chauffage)

Sinon la sortie B6 s'active (chauffage)

Principe de fonctionnement d'un codeur



L'avantage d'utiliser un codeur, est qu'il fournit un signal directement numérique, il peut donc être directement traité par le microcontrôleur.

Exemple ici avec un codeur angulaire de position :

32 positions possibles soit une précision de $360^\circ / 32 = 11,25^\circ$ position codée sur 5 bits.

