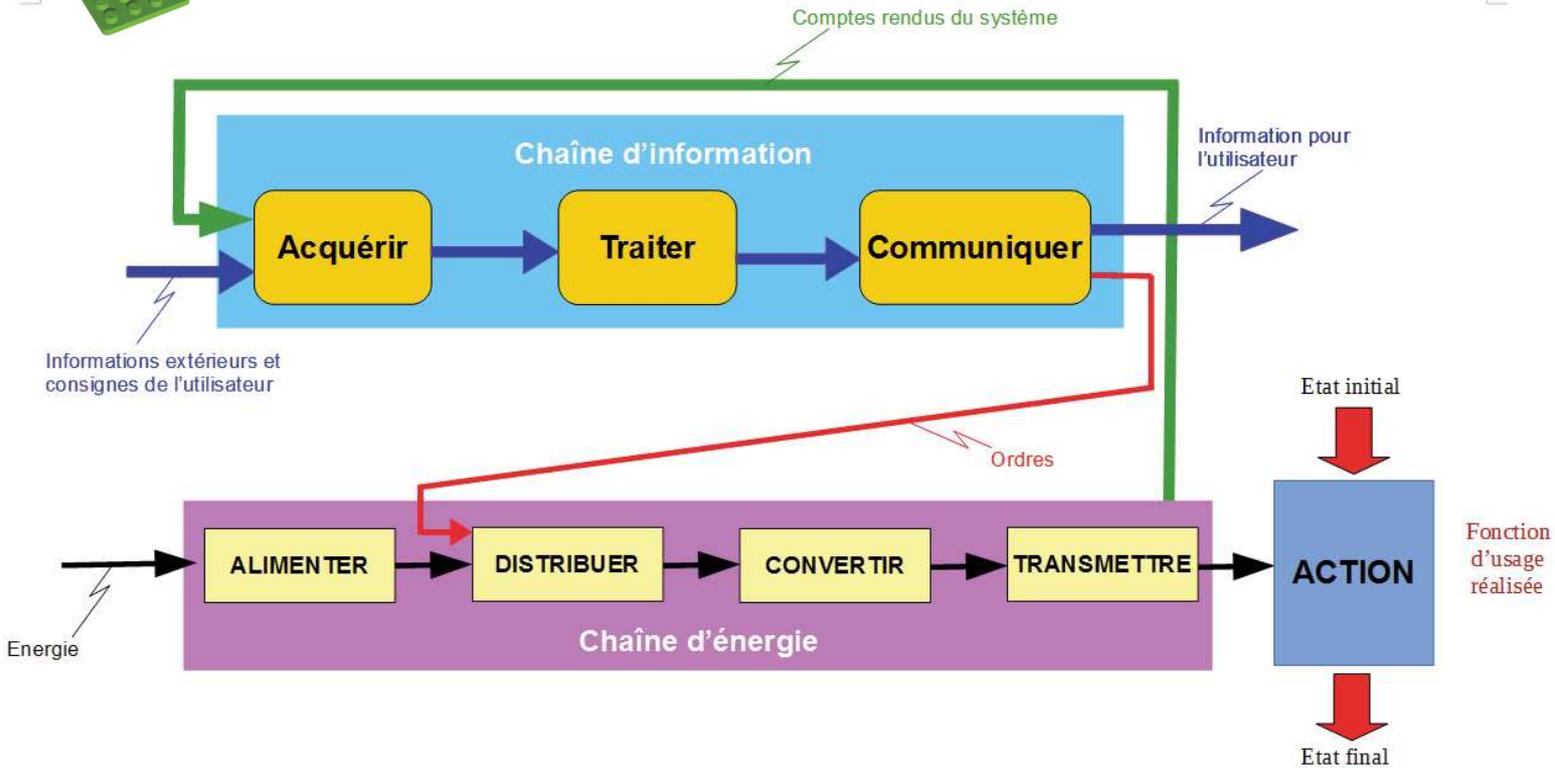




Représentation fonctionnelle des systèmes

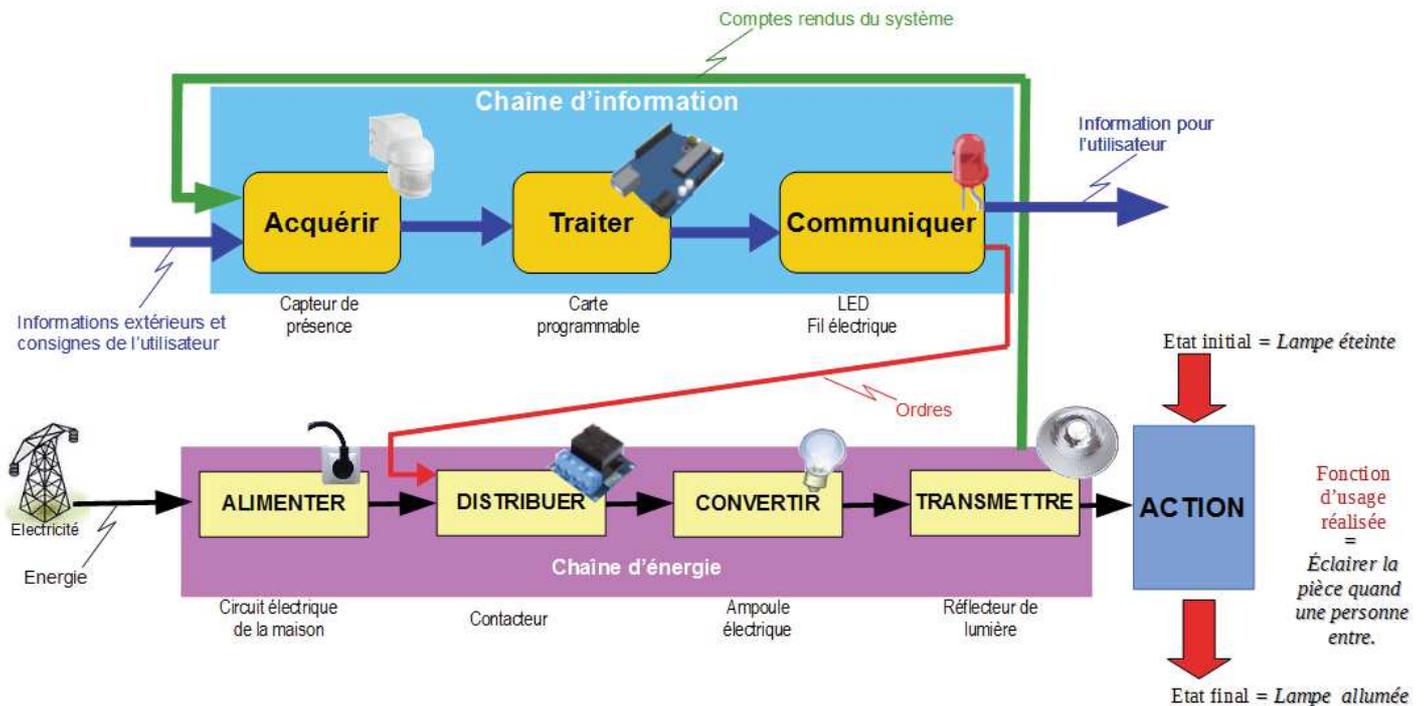


Un système permet de répondre à un besoin.
Il est composé d'éléments ayant chacun leurs fonctions.



Pour plus de précision : <http://tkcollege.fr/4-chaine-energie-et-information.html>

Exemple : Système d'éclairage avec détection de présence





Besoin



Le besoin est une nécessité ou un désir ressenti par une personne. Il évolue en fonction du progrès technique, des inventions et des innovations. Si l'objet technique ne répond pas à un besoin alors il n'est d'aucune utilité ! Un individu ressent différents besoins hiérarchisés.

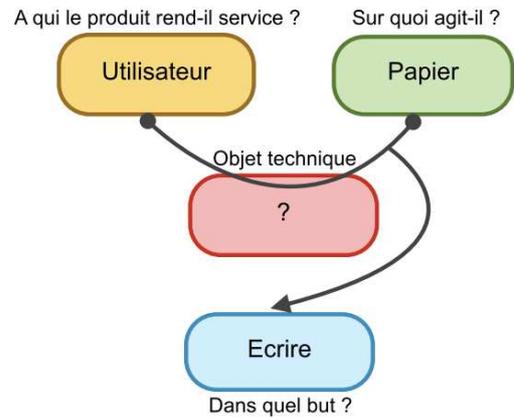


Pyramide des besoins

Non prioritaire



Prioritaire



Outil graphique d'identification d'un besoin :

Exemple ici avec le stylo

Contraintes



Pour satisfaire notre besoin, un objet technique doit prendre en compte des contraintes qui limitent la liberté du concepteur.

Exemple avec un casque audio :



Lors d'une démarche de projet, l'ensemble des contraintes sont indiquées dans un document nommé « Cahier des charges ». Le cahier des charges est le contrat à remplir par le concepteur.

Normalisation



En plus des contraintes personnelles, l'objet technique doit respecter des normes, qui sont des contraintes supplémentaires pour nous protéger ou simplifier son utilisation.

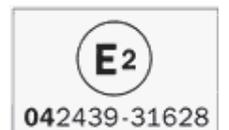
Exemple avec la prise audio du casque : Le format de la prise est une norme qui permet d'utiliser l'objet avec ensemble des appareils existants qui réalisent la même fonction.



La normalisation est primordiale, des organismes sont donc en charge de la faire respecter : AFNOR, CE, ISO



Exemple avec un casque : Pour qu'un casque soit homologué en France et donc reconnu officiellement « protecteur », il doit comporter une étiquette verte NF ou Blanche E+n°.





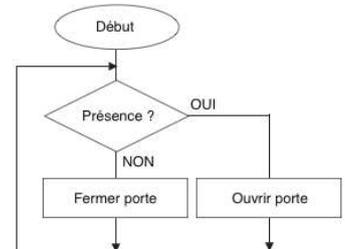
CT 4.2-CT 5.5-IP 2.3	Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.
CT1.3-CT2.5-CT2.7-DIC 1.5	Imaginer des solutions pour produire des objets et des éléments de programmes informatiques en réponse au besoin.
CT 3.1-OTSCIS 2.1	Exprimer sa pensée à l'aide d'outils de description adaptés : croquis, schémas, graphes, diagrammes, tableaux.

Qu'est ce qu'un organigramme ?



Un organigramme (aussi appelé logigramme) est une suite d'instructions précises et structurées qui décrit la manière dont on résout un problème.

Un organigramme peut être utilisé pour programmer un système.

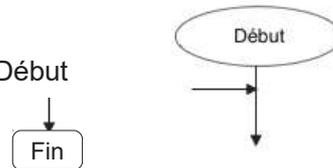


Symboles de base

Des normes d'écritures sont à respecter :

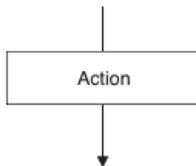


- Un organigramme s'écrit toujours de haut en bas
- Un organigramme commence toujours par la case Début
- Un organigramme à souvent une case fin



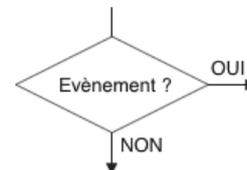
Les cases actions :

- Elles ont une seule entrée (toujours au-dessus).
- Elles ont une seule sortie (toujours en-dessous).



Les cases de test :

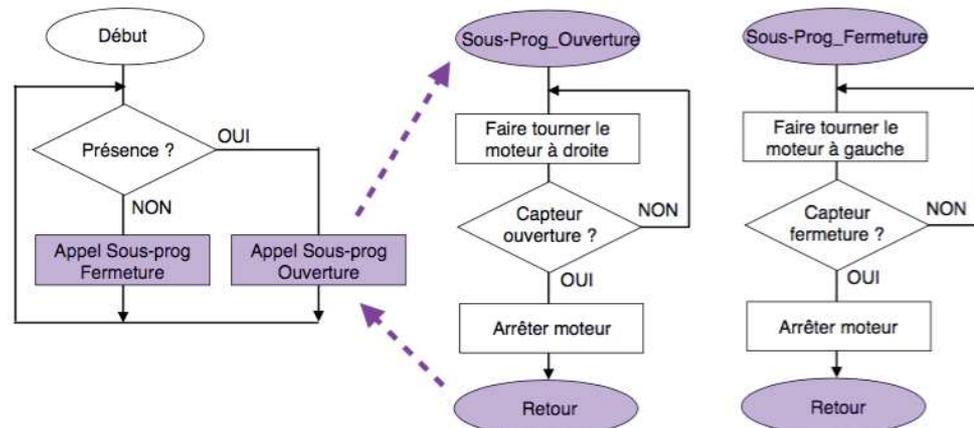
- Elles ont une seule entrée (toujours au-dessus).
- Elles ont toujours 2 sorties
- A l'intérieur de la case on indique une question (ex : Mur en face ?)



Organigramme et gestion des sous-problèmes



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





CT 4.2-CT 5.5-IP 2.3	Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.
CT1.3-CT2.5-CT2.7-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.

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 **microprocesseur** ou un **microcontrôleur**.

Un programme s'exprime successivement sous différentes formes :

1 ^{ère} Étape	2 ^{ème} Étape (collège)	2 ^{ème} Étape (professionnel)
Langage naturel = Algorithme	Langage graphique Logigramme ou programmation par Blocs	Code
Allumer la DEL sortie 2 Attendre 1 seconde Eteindre la DEL sortie 2 Attendre 1 seconde Allumer la DEL sortie 2 Attendre 1 seconde Eteindre la DEL sortie 2 Attendre 1 seconde ...		<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); delay(1000*1); pinMode(2,OUTPUT); digitalWrite(2,0); delay(1000*1); }</pre>

Ces différentes formes de programmes facilitent le travail du programmeur. Elles seront ensuite traduites en langage compréhensible par le microprocesseur ou le microcontrôleur, « 0 » et « 1 » : le code **binaire**.

Boucles

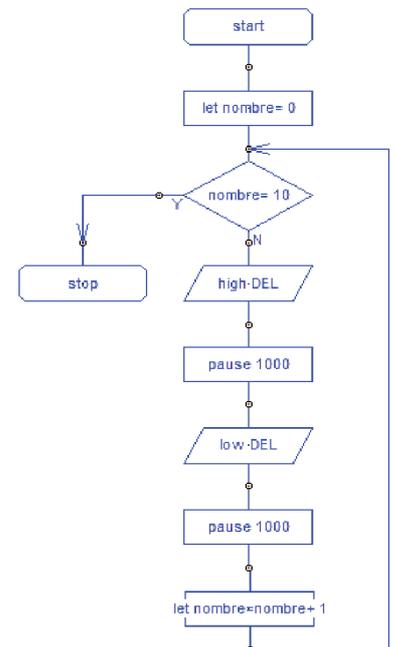


Un programme doit être le plus court possible. Lorsque des instructions sont répétées, on utilise des **boucles** pour optimiser le programme.

Exemple de boucles : TANT QUE, JUSQU'À, REPETER ...



Exemple Diode clignote 10 fois



Il est possible d'imbriquer plusieurs boucles les unes dans les autres pour répondre au problème.



Déclenchement d'une action par un événement, instructions conditionnelles



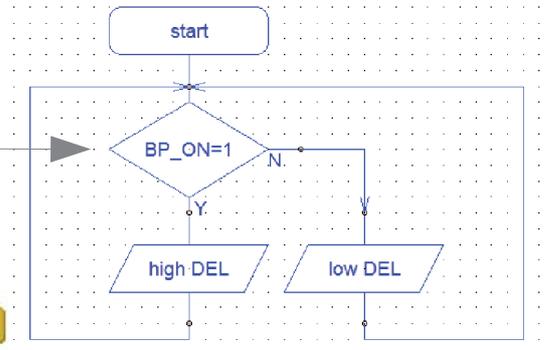
L'enchaînement des opérations et le **déclenchement d'actions** se fait toujours par un **événement** :

- interne au programme (début programme, variable, ...)
- externe au programme (capteur, touche du clavier, ...)

Condition dans un Algorithme

SI ...
ALORS ...
SINON ...

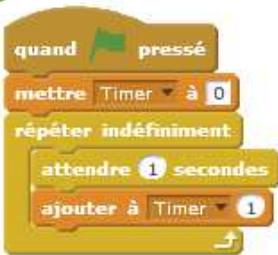
Condition en langage graphique



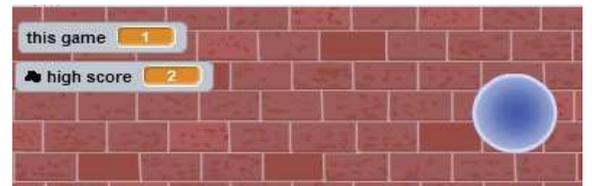
Variable informatique



Une **variable** est une donnée (information) associée à un nom. Elle est mémorisée et elle peut changer dans le temps, lors de l'exécution du programme.



Exemple : timer



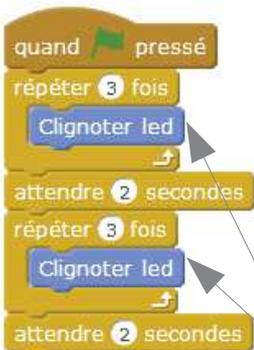
Exemple : score et meilleur score pour un jeu

Sous-Programme

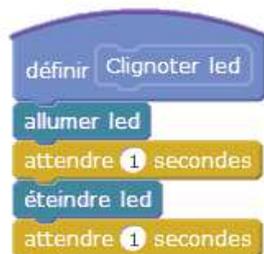


Les **sous-programmes** sont des modules de programmation indépendants répondant à des **sous-problèmes** du programme principal.

Programme principal



Sous-programme : clignoter led

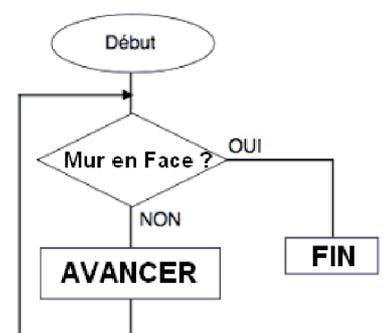


Appel du sous-programme : Clignoter led

Programme principal



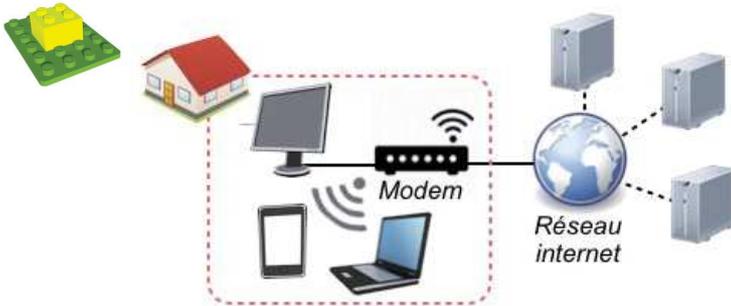
Sous-programme : Alleraumur



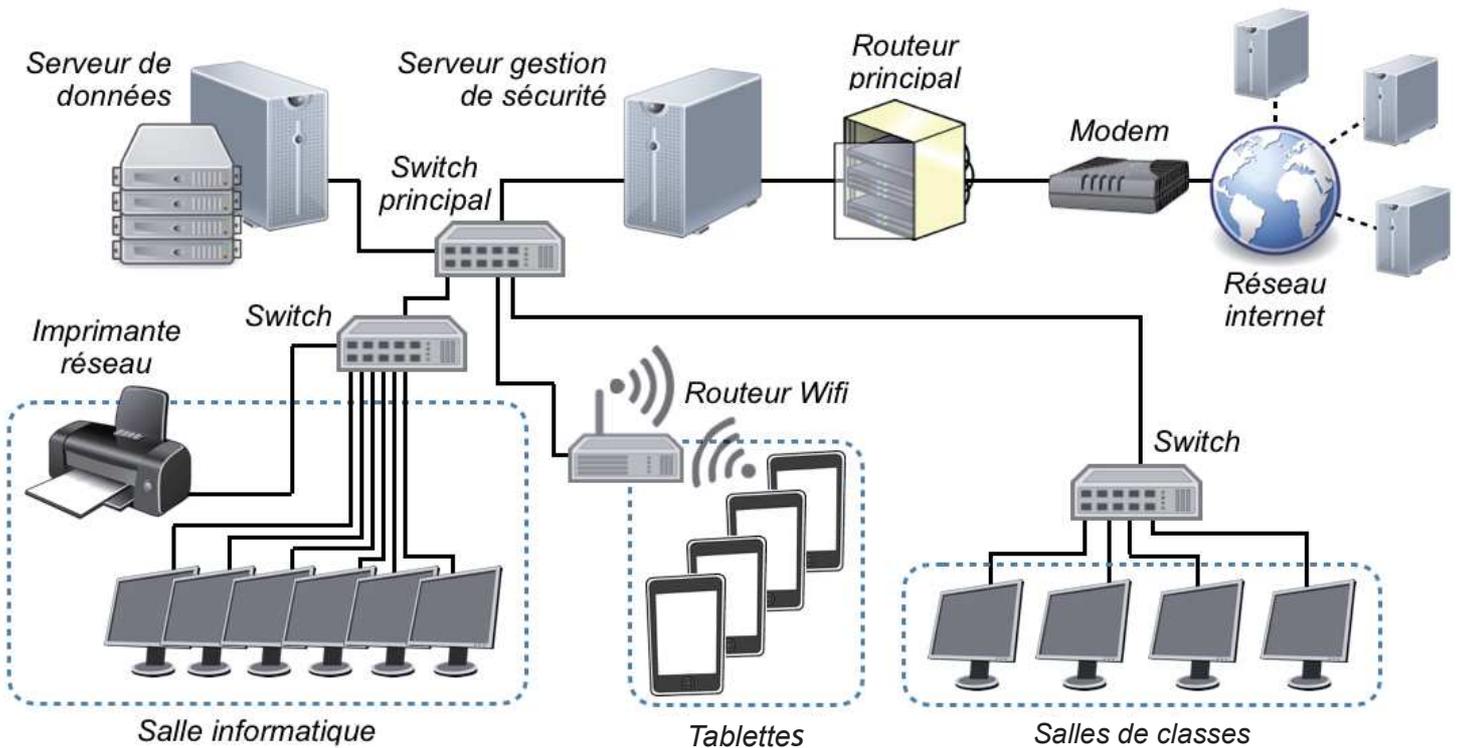
Appel du sous-programme : Alleraumur



Architecture d'un réseau



Contrairement à l'installation simple que nous pouvons retrouver à la maison, l'architecture d'un réseau local s'impose au collège comme dans toutes entreprises qui utilisent des moyens numériques.



Composants principaux d'un réseau

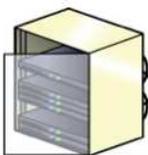


Le modem permet une connexion à internet. C'est une interface entre le réseau et l'extérieur (câble téléphonique ou fibre optique).



Un serveur permet de :

- Gérer les autorisations des utilisateurs
- Stocker les données des utilisateurs
- Gérer la sécurité des données qui transitent entre internet et le réseau ainsi qu'au sein du réseau lui même (firewall).



Le routeur permet de relier plusieurs réseaux locaux ensemble. Il est présent dans **une baie de brassage** : armoire technique qui centralise les connexions du réseau local.



Le switch (commutateur) permet de relier plusieurs équipements (poste informatique, imprimante, ...) au sein du réseau local.



Le routeur Wifi permet tout comme le switch de relier plusieurs équipements mais avec une connexion sans fil en Wifi. Pour cela, il génère un sous-réseau local qui lui est propre (d'où le mot routeur)

Moyens de connexion à un réseau



Actuellement il existe différents moyens de communication soit autant de connexion à un réseau.

Le choix de la solution de connexion se fera en fonction de la nature mobile de l'équipement (appareil fixe ou mobile) et en fonction de la portée et de la rapidité souhaitée.

Moyen de connexion	Transmission du signal	Portée de la communication	Rapidité de communication	Nature du signal
Câble ethernet	Filaire	😊😊😊	😊😊	Electrique
Courant porteur en ligne (CPL)	Filaire	😊	😊😊	Electrique
Fibre optique	Filaire	😊😊😊	😊😊😊	Impulsion lumineuse
WiFi	Sans fil	😊	😊	Onde radio
Bluetooth	Sans fil	😊	😊	Onde radio
Li-Fi	Sans fil	😊	😊😊😊	Impulsion lumineuse infra-rouge
Satellite	Sans fil	😊😊😊	😊	Onde radio

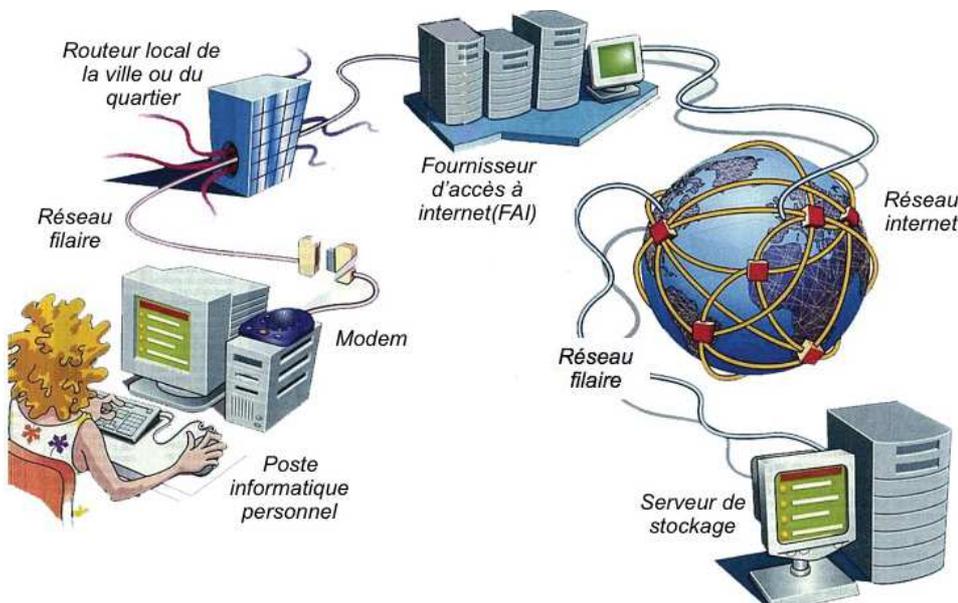
Un réseau mondial : Internet



Internet est un réseau de millions d'ordinateurs et d'objets interconnectés pour communiquer et échanger des informations. L'utilisateur se connecte à internet par son fournisseur d'accès à internet (FAI) qui lui fournit une adresse IP unique le temps de la connexion.

Chaque ordinateur ou équipement (tablette, smartphone, console,...) connecté à internet possède donc une adresse IP propre.

Des serveurs spécifiques font le lien entre une URL et une adresse IP.



Ainsi il est facile de se connecter avec son navigateur (firefox, chrome, internet explorer, ...) à un serveur (qui stocke un site internet par exemple) avec uniquement l'adresse URL.

Exemple :
Laposte.fr = 94.124.132.36/

↑ URL ↑ IP



CT4.2-CT5.5-IP2.3
CS1.6-MSOST1.4

Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs
Identifier les flux d'information sur un objet et décrire les transformations qui s'opèrent.

Systemes automatisés



Dans les systèmes automatisés, on trouve toujours une Interface programmable (ou plusieurs Interfaces) qui fait le lien entre les capteurs et les actionneurs.

Capteur



Le capteur réalise l'acquisition d'une grandeur physique (température, luminosité, présence, distance,...) qu'il transforme en un signal analogique ou numérique afin qu'il puisse être traité par l'interface programmable.



Capteur de mouvement



Capteur d'humidité dans le sol



Capteur de contact

Actionneur



L'actionneur transforme l'énergie en une action.



Del ou Led



Moteur électrique



Buzzer

Interface programmable



Les capteurs permettent d'acquérir des informations qui sont traitées par une interface programmable pour piloter des actionneurs.

Interface programmable :
Arduino



Interface programmable :
Picaxe



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

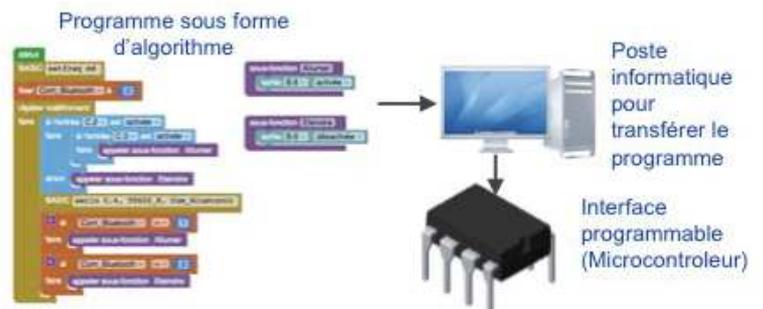
Système embarqué



Le système embarqué réagit en fonction de l'acquisition de ses capteurs, des informations qu'il reçoit de l'extérieur (capteurs externes ou communiqués depuis un appareil nomade) et de la programmation qui lui est associée.

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.





CT 4.2-CT 5.5-IP 2.3	Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.
CT1.3-CT2.5-CT2.7-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.

Qu'est ce qu'un algorithme ?



Un algorithme est une suite d'instructions écrite à la main qui décrit le programme que l'on doit faire.

L'algorithme est la première étape de la réalisation d'un programme informatique.

Il est généralement écrit en langage naturel. Il comporte les mots **Si, alors, sinon, tant que, et, ou, non**

*Si le capteur de mouvement détecte un mouvement ou si le capteur d'ouverture détecte une ouverture alors la sirène se déclenche et le gyrophare clignote.
Si le code d'annulation est tapée la sirène et le gyrophare s'éteignent.*

Méthode



4 étapes sont à respecter pour faire un algorithme :

- Surligner les mots importants dans le texte qui décrit le programme
- Avec les mots surligner faire des phrases complètes puis continuant par **alors**
- Aller à la ligne à chaque nouvelle phrase
- Relire l'algorithme et vérifier que les mots surlignés précédemment sont présent.

A finir

Exemple