

<b>Séquence S3</b> <i>Comment programmer un éclairage automatique ?</i>	<b>SYNTHESE</b>	Cycle 4
	<b>Assurer le confort dans une habitation</b>	<b>5ème</b>

Compétences développées en activités		Connaissances associées
<b>CT 1.1</b>	Imaginer, synthétiser, formaliser et respecter une procédure, un protocole.	Outils numériques de présentation. Charte graphique. Procédures, protocoles. Ergonomie
<b>CT 4.2</b>	Appliquer les principes élémentaires de l'algorithmique et du codage à la résolution d'un problème simple.	Chaîne d'énergie. Chaîne d'information. Déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles. Systèmes embarqués. Capteur, actionneur, interface.
<b>CS 1.6</b>	Analyser le fonctionnement et la structure d'un objet, identifier les entrées et sorties.	Représentation fonctionnelle des systèmes. Structure des systèmes. Chaîne d'énergie. Chaîne d'information

## I. Représentation fonctionnelle d'un système

Un **système** est un ensemble structuré de composants qui interagissent entre eux pour répondre à un besoin de l'homme. Il agit sur une **matière d'œuvre** et lui apporte une **valeur ajoutée** ; en faisant cela, il réalise une fonction.

La figure ci-dessous illustre un mode de représentation fonctionnelle simplifié d'un lampadaire.



Dans l'exemple ci-dessus, la matière d'œuvre est l'espace à éclairer ; la valeur ajoutée est l'éclairage de cet espace. La fonction du lampadaire est donc d'éclairer un espace.

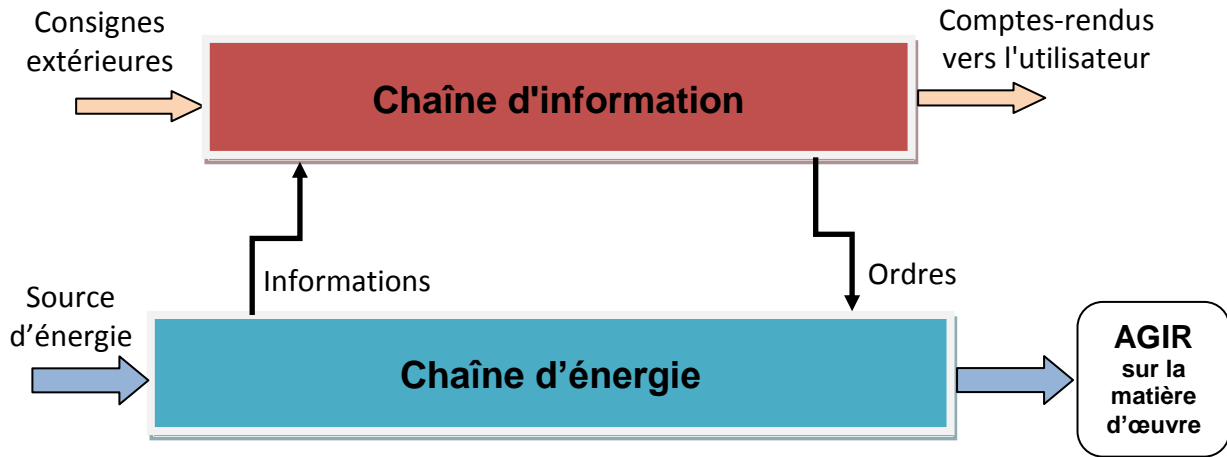
Les systèmes sont plus ou moins complexes et peuvent fonctionner avec ou sans intervention extérieure. Un véhicule automobile est un système complexe qui nécessite un conducteur. Un robot est également un système complexe mais qui est **programmé** pour réaliser en toute autonomie un certain nombre de tâches. Un lampadaire d'éclairage urbain peut être considéré comme étant de complexité moindre, mais qui est programmé pour éclairer à certaines heures de la journée.

## II. Organisation fonctionnelle et structurelle d'un système

Pour fonctionner, un système a besoin d'énergie. La chaîne d'**énergie** est l'ensemble des composants qui permettent de stocker l'énergie, de la distribuer puis de la convertir pour obtenir l'action voulue.

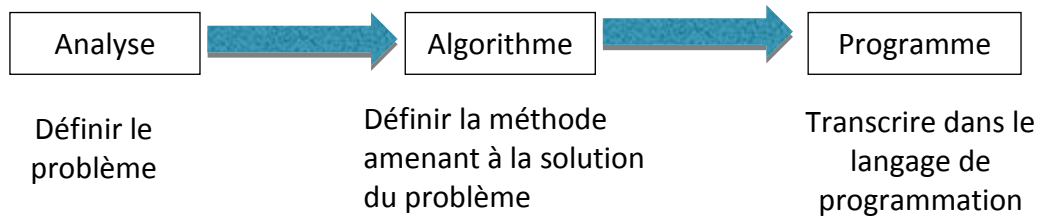
De plus, un système échange des **informations** avec son milieu extérieur. Il reçoit des **consignes** qui peuvent être fournies par un utilisateur externe ou bien transmises à distance. Il **communique** des **comptes rendus** sur son état de fonctionnement. Certains systèmes sont **programmés** pour pouvoir effectuer en toute autonomie un certain nombre de tâches. Pour cela, ils sont munis de **capteurs** qui **acquièrent** les **informations** correspondant à l'état de fonctionnement du système. Celles-ci sont ensuite **traitées** afin de pouvoir délivrer les **ordres** qui vont permettre d'enclencher les actions à réaliser pour **agir** sur la matière d'œuvre. L'ensemble des composants qui permettent d'**acquérir** ou de détecter les informations, de les traiter et de communiquer appartient à la **chaîne d'information** du système.

La figure ci-dessous illustre une représentation simplifiée de l'organisation fonctionnelle d'un système.



### III. Programmer un système

Pour réaliser un programme, il est nécessaire de suivre la démarche suivante :



L'analyse décrit le problème à résoudre. L'algorithme décrit la solution au problème en langage naturel (langage humain). L'algorithme est une représentation graphique de l'algorithme. Le programme est la traduction de l'algorithme (ou algorithme) en langage compréhensible par la machine.

Exemple : Comment automatiser l'éclairage d'une pièce ?

Analyse	Algorithme	Algorithme
On souhaite que la pièce ne soit allumée uniquement qu'en présence d'une personne.	<b>Répéter (</b> <b>Si</b> une personne est présente <b>alors</b> allumer la lampe <b>Sinon</b> garder la lampe éteinte)	<pre> graph TD     Start([Début]) --&gt; Decision{présence personne}     Decision -- Oui --&gt; Action1[allumer lampe]     Decision -- Non --&gt; Action2[éteindre lampe]     Action1 --&gt; Start     Action2 --&gt; Start         </pre>