



Structurer les connaissances

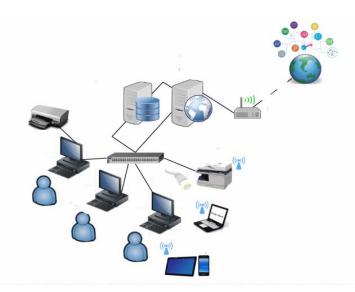
Composants d'un réseau, architecture d'un réseau local





Pour pouvoir se servir du matériel informatique du réseau local du collège, il faut en comprendre son architecture.

• L'utilisation d'un schéma informatique permet de représenter l'architecture du réseau composé principalement de plusieurs postes informatiques appelés postes clients, d'un serveur, d'un commutateur (switch) - des liaisons par câble, fibre optique ou sans fil, d'un routeur Internet qui permet de se connecter au réseau Internet.



Ce schéma représente un **réseau informatique** typique d'un établissement scolaire.

C'est un ensemble d'ordinateurs et de périphériques qui sont reliés entre eux pour partager des informations et accéder à des services.

Il est relié au **réseau Internet, qui** est un gigantesque **ensemble de réseaux informatiques** à l'échelle mondiale.

Environnement informatique de l'établissement

• Pour pouvoir se servir du matériel informatique du réseau local du collège, il faut en comprendre son fonctionnement.



L'usager devra saisir ses **identifiants** : **nom d'utilisateur** et **mot de passe**, depuis le poste client.

Cela lui permettra de sauvegarder ou consulter des fichiers sur le serveur de données mais également d'imprimer son travail en choisissant une imprimante réseau "visible" dans tout le réseau du collège.

L'usager peut saisir une URL, une adresse de site dans son navigateur web. Le serveur Internet « proxy » examine la requête et accepte la connexion au site si celui-ci est autorisé. L'ensemble routeur, serveur proxy, pare feu permet l'accès au serveur web.

Un réseau informatique est un ensemble d'équipements informatiques (ordinateurs, imprimantes, tablettes, smartphones, etc.) reliés entre eux par des câbles ou avec des technologies sans fil (wifi, Bluetooth). Il permet de communiquer des informations et de partager des périphériques (imprimante, scanner, etc.).







Moyens de connexion d'un moyen informatique





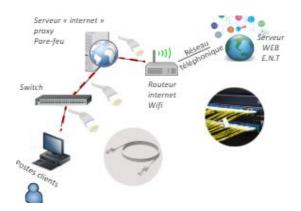
Pour pouvoir communiquer entre eux, les objets connectés ainsi que les composants d'un réseau doivent établir une connexion.

- Il existe deux moyens pour communiquer :
 - soit des connexions filaires.
 - soit des communications sans fil.

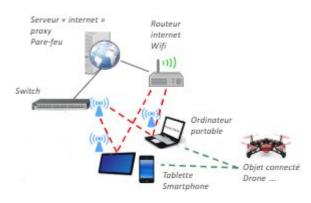
Connexions filaires		Communications sans fil	
Fibres optiques par impulsions lumineuses		Le wifi	WÎfi
Câble réseau par impulsions électriques		Le bluetooth	₿ Bluetooth
Câble mini Jack par impulsions électriques		Emetteur /récepteur ondes lumineuses : infrarouge d'une télécommande.	

• Pour réaliser ces connexions, on utilise :

des moyens matériels



des moyens sans fil



Les composants d'un réseau informatique nécessitent d'être connectés pour communiquer.

Pour réaliser ces connexions, on utilise des moyens matériels : câbles, fibre optique... et des moyens sans fil : ondes radio (WiFi, Bluetooth), infrarouge.







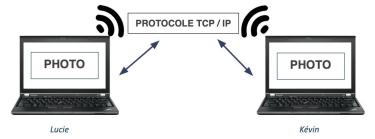
Notion de protocole, d'organisation de protocole en couche





Pour **échanger** des informations, on utilise un **réseau de communication**. Pour communiquer dans ce réseau, un **mode de communication** commun est utilisé, c'est le **protocole**.

 Internet est le plus grand réseau de communication. Pour se comprendre, les appareils (ordinateurs, téléphones, tablettes...) qui y sont connectés ont besoin d'un langage commun : c'est le protocole TCP/IP.



Exemple : un élève envoie une photo à un autre élève.

- Il existe différents types de protocoles suivant les informations à échanger :
- Protocole HTTP : permet d'échanger des pages web entre le client et le serveur.
- Protocole DNS : affecte une adresse IP à un nom de domaine.
- Protocole SMTP : permet d'envoyer un email.
- Protocole POP3 ou IMAP : permet de recevoir un email.
- Protocole FTP : permet d'échanger des fichiers.



Dans un réseau, l'information circule d'une adresse IP (expéditeur) vers une autre adresse IP (destinataire). Chaque ordinateur sur le réseau est identifié par une adresse IP unique.





Composition d'une adresse IP

Un protocole informatique est un ensemble de règles qui définit le mode de communication entre deux machines. Ces protocoles vont permettre d'organiser les transferts de données entre les appareils identifiés par leurs adresses IP.







Notion d'algorithme de routage

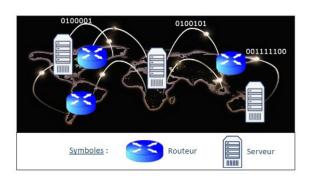




Pour **assurer les échanges d'informations** dans un **réseau de communication**, on utilise des **algorithmes de routage**.

Pour circuler sur internet, l'information est découpée en paquets de 0 et 1 (binaire).
 Ces paquets sont ensuite aiguillés par les routeurs vers les réseaux et serveurs auxquels ils sont destinés.

Les paquets d'informations binaires sont relayés de routeurs en routeurs pour établir la communication entre expéditeur et destinataire.



Chaque routeur utilise un algorithme de routage pour déterminer le meilleur aiguillage des informations.

 L'algorithme de routage étudie et compare toutes les routes accessibles par chaque routeur ou poste du réseau pour faire circuler l'information d'un point à un autre du réseau.

Ces routes sont inscrites dans une liste appelée : table de routage.

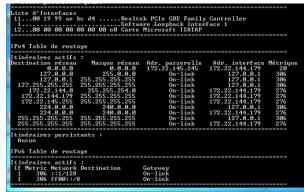


Si Lucie envoie une photo à Kévin

L'algorithme de routage compare toutes les routes de la table de routage afin de rendre la communication plus efficace.

Pour connaître la table de routage d'un poste, on peut utiliser la commande route print dans l'invite de commande (cmd).

On découvre ainsi toutes les routes que l'algorithme va comparer afin de trouver la meilleure.



Un algorithme de routage est une méthode de calcul qui permet de déterminer le meilleur chemin entre un expéditeur et un destinataire lors de l'échange d'informations sur un réseau.

Pour ce faire, l'algorithme compare toutes les routes inscrites dans les tables de routage et indique aux routeurs sur quelles voies renvoyer l'information reçue.







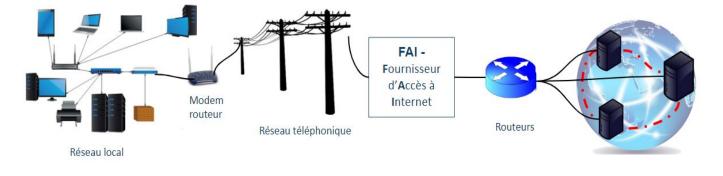
Internet





Pour assurer les échanges d'informations entre tous les ordinateurs sur Terre, on utilise le réseau Internet.

• Ce réseau utilise du matériel et des Applications reliés entre eux et connectant ainsi des millions d'ordinateurs. Tout ordinateur domestique, d'un réseau d'établissement scolaire ou d'une entreprise, voulant se connecter à Internet, utilise un modem-routeur qui, via le réseau téléphonique, se connecte au FAI (Fournisseur d'Accès à Internet). Celui-ci va ensuite établir une connexion avec les différents routeurs qui vont aiguiller l'information vers le bon serveur hébergeant le site web ou le service demandé.



Réseau local Modem Serveur hébergeur

 Pour se connecter à Internet depuis le collège, il faut ouvrir sa session puis utiliser un navigateur capable d'afficher les pages hébergées sur un serveur web,et pour faire des recherches un moteur de recherche.



Ouverture d'un session

Ouverture du navigateur

le serveur web renvoie le document que le navigateur affichera

Internet relie des millions d'ordinateurs dans le monde qui communiquent grâce au protocole TCP/IP. Internet est donc le réseau des réseaux. Du point de vue matériel, un modem-routeur permet d'assurer la connexion avec un fournisseur d'accès Internet (FAI) puis aux différents serveurs. Du point de vue logiciel, un navigateur permet d'afficher et de consulter les pages que le moteur de recherche propose grâce aux requêtes rédigées avec des mots-clés par l'utilisateur.







Analyser le comportement d'un système réel et décomposer le problème posé en sous-problèmes





Pour analyser le comportement attendu d'un système réel et décomposer le problème posé en sous-problèmes afin de structurer un programme de commande, il faut :

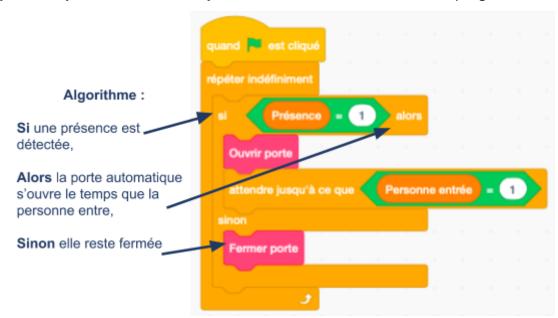
 Observer le système automatisé en fonctionnement.



 Définir ses actions et rédiger l'algorithme correspondant à la résolution de son problème initial :

Exemple d'un algorithme pour le fonctionnement de la porte de supermarché **avec 3** sous-problèmes :

- Si quelqu'un est détecté devant la porte automatisée, la porte doit d'ouvrir.
- S'il n'y a personne de détecté devant la porte automatisée, la porte doit se fermer.
- Ces 2 actions doivent être vérifiées en permanence indéfiniment.
- Décomposer le problème en sous problèmes et traduire en blocs de programmation :



Le fonctionnement d'un système automatisé répond à un ou des problèmes posés.

L'observation de ce fonctionnement permet de définir et de décomposer le problème en sous-problèmes qui correspondent aux étapes de l'algorithme. Ces étapes sont ensuite traduites en blocs dans le logiciel de programmation.







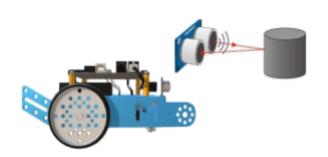
Analyser le comportement d'un système réel et décomposer le problème posé en sous-problèmes





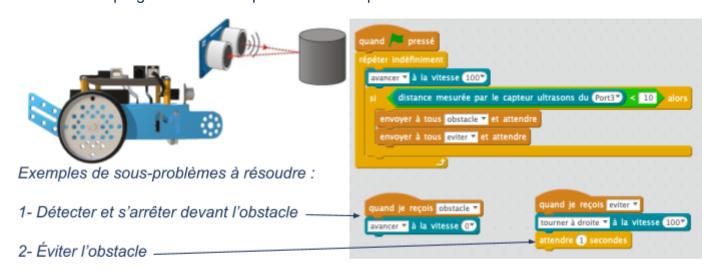
Pour analyser le comportement attendu d'un système réel et décomposer le problème posé en sous problèmes afin de structurer un programme de commande, il faut :

• Observer le système automatisé en fonctionnement afin de décrire son comportement et ldentifier les sous-problèmes à résoudre lors de sa programmation :



Le comportement attendu de ce robot est :

- Il se déplace en avant jusqu'à ce qu'il rencontre un obstacle.
- S'il rencontre un obstacle, il doit s'arrêter.
- Ensuite, il doit tourner à droite pendant une seconde.
- Enfin, il doit recommencer à se déplacer en avant jusqu'au prochain obstacle.
- Créer ou modifier des scripts qui permettent d'adapter le programme général et d'appeler des sous-programmes correspondant au comportement attendu.



Le fonctionnement d'un système automatisé répond à un ou des problèmes posés. L'observation de ce fonctionnement permet de structurer un programme correspondant au comportement attendu.

Ce programme est constitué de **sous-programmes** qui **répondent chacun à un sous-problème**. (Ces sous-programmes permettent d'adapter le programme général).







Ecrire, mettre au point (tester, corriger) et exécuter un programme





Lors de la conception d'un système programmable, le concepteur va devoir écrire un programme pour commander le système en fonction du comportement attendu.

Le concepteur écrit un algorithme à partir du cahier des charges...
 Puis, il créé des programmes à l'aide d'un logiciel de programmation par blocs pour commander le système réel.



Exemple : un robot évitant un obstacle





Si la distance détectée par le robot est inférieure à 10 cm, alors le robot devra s'arrêter, sinon il continue d'avancer.

Ces **logiciels** possèdent des **menus**, où sont stockés des **instructions**, qui permettront de **concevoir un programme** et enfin **vérifier le comportement attendu** en l'exécutant.

```
mBot - générer le code
avancer ▼ à la vitesse 100♥
répéter indéfiniment
si distance mesurée par le capteur ultrasons V < 10 alors
avancer ▼ à la vitesse 0♥
```

Comment écrire, mettre au point un programme pour vérifier le comportement attendu ?

Pour écrire un programme, plusieurs étapes sont nécessaires :

- 1. vérifier les attentes du cahier des charges : tenir compte des fonctions de service.
- 2. **rédiger un algorithme** : **des phrases** qui permettent de respecter **les fonctions de service demandées**.
- 3. repérer les capteurs et les actionneurs à utiliser
- 4. repérer les instructions sur le logiciel de programmation
- 5. mettre au point le programme : assembler les instructions.
- 6. exécuter le programme
- 7. **vérifier le comportement attendu** : Si cela ne correspond pas au comportement attendu. Il faut corriger le programme.
- 8. adapter le programme pour obtenir le comportement attendu

Pour écrire, mettre au point, exécuter un programme commandant un système réel programmable et vérifier le comportement attendu, il faut donner des instructions à ce système, en utilisant des langages compréhensibles à la fois par le concepteur et par le système programmé. Plusieurs étapes sont nécessaires : on rédige un algorithme, que l'on met au point avec un logiciel de programmation par blocs et on vérifie le comportement attendu du système réel.

Fiche connaissance – 1/1 IP-2-2-C1





Structurer les connaissances

Notions d'algorithme et de programme



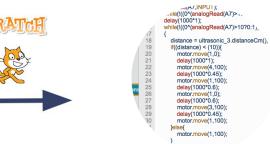


Les **objets connectés** sont souvent **programmés** pour fonctionner **automatiquement**. Chaque fonction de l'objet connecté peut être assimilée à un **problème à résoudre** par un **algorithme**.

• En informatique, un **algorithme** est une **suite logique d'opérations** ou **d'instructions** aboutissant à la résolution d'un problème.

L'algorithme d'une voiture sans conducteur va devoir piloter :

- la mise en marche, la direction et l'arrêt
- la détection des lignes délimitant la route
- la détection et l'évitement des obstacles...
- Cet algorithme est traduit, grâce à un langage de programmation, en un programme exécutable par un système informatique (ordinateur, carte microprocesseur, objet connecté...).
 - Le robot avance de 5m
 - Le robot tourne à gauche de 30°
 - Le robot avance de 3m
 - Le robot tourne à gauche de 60°
 - Le robot avance de 2m



• Quelles sont les étapes de l'élaboration de l'algorithme et du programme ?

Etape 1 : Ecrire un algorithme en langage naturel : suite logique d'opérations ou d'instructions, souvent rédigées sur feuille de papier en utilisant des mots clés : **si, alors, tant que, jusqu'à...**



Etape 2 : Construire une représentation graphique de l'algorithme à l'aide d'un logiciel.



Etape 3: A partir de la représentation graphique, **le logiciel traduit l'algorithme** en **langage de programmation** pour que l'objet puisse exécuter le programme.

```
1 finclude <Arduino.h>
2 finclude <Wire.h>
3 finclude <SoftwareSerial.h>
4
5 double angle_rad = PI/180.0;
6 double angle_deg = 180.0/PI;
7
```

Un algorithme décrit une suite finie d'opérations à appliquer dans un ordre déterminé pour résoudre un problème. Un algorithme peut être traduit, grâce à un langage de programmation, en un programme exécutable par un système informatique (ordinateur, carte microprocesseur, objet connecté).







Notions d'algorithme et de programme









Structurer les connaissances Notions d'algorithme et de programme





Les **objets connectés** sont souvent programmés pour fonctionner **automatiquement**. Chaque fonction de l'objet connecté peut être assimilée à un problème à résoudre par un algorithme.

• Un algorithme est la méthode permettant de résoudre ces problèmes. En informatique il peut s'apparenter à une suite logique d'opérations ou d'instructions aboutissant à la

résolution d'un problème.

L'algorithme d'une voiture sans conducteur va devoir piloter:

- la mise en marche, la direction et l'arrêt
- la détection des lignes délimitant la route
- la détection et l'évitement des obstacles...
- le repérage d'une place pour s'y garer



Quelles sont les étapes de l'élaboration de l'algorithme et du programme ?

Etape 1 : Ecriture d'un algorithme en langage naturel : Suite logique d'opérations ou d'instructions, souvent rédigées sur feuille de papier en utilisant le langage naturel et des mots clés : si, alors, tant que, jusqu'à ...

- Si le robot détecte un obstacle avec son capteur de pare-choc alors tourner à gauche de 90°
- Sinon avancer indéfiniment

Etape 2 : Construction à l'aide d'un logiciel d'une représentation graphique de l'algorithme.

Répéter les instructions, jusqu'à un événement précis.



On construit l'algorithme à l'aide d'un logiciel comme Scratch ou Blockly.

Etape 3 : à partir de la représentation graphique, on réalise la génération de l'algorithme en langage de programmation par le logiciel, pour que l'objet puisse exécuter le programme après traduction.

Il existe différents langages de programmation comme le C, le C++ le Python, pouvant 11001010010000001112011 être abordés au lycée. Après une succession de traductions, l'algorithme devient un 001110100011011110001 langage machine binaire (0 ou 1) interprété par le microprocesseur de l'ordinateur.



Un algorithme décrit une suite finie d'opérations ou de règles à appliquer dans un ordre déterminé pour résoudre un problème. Un algorithme peut être traduit, grâce à un langage de programmation : C / C++ / Python ..., en un programme binaire 0 et 1, interprété et exécuté par le microprocesseur du système programmable.







Notion de variable informatique





La programmation des **objets connectés** nécessite la gestion de **situations complexes** (déplacements, trajectoires, mesures des capteurs...). Pour résoudre ces problèmes, les programmeurs introduisent des **variables informatiques** dans leurs **algorithmes**.

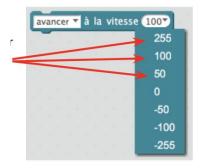
 Les variables statiques correspondent à des valeurs constantes et fixes que l'on utilise pour fixer des valeurs numériques comme pour définir des vitesses, des couleurs, des distances ...



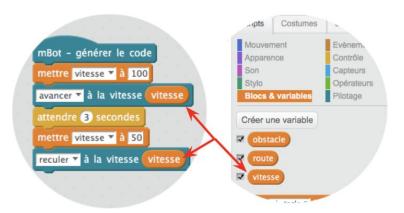
Comment utiliser des variables statiques pour régler la vitesse d'un robot ?

Cas 1 - La plupart du temps, les variables sont prédéfinies dans le logiciel de programmation.

Le programmeur a le choix entre plusieurs valeurs pour régler la vitesse de son robot.



Cas 2 - Les variables peuvent aussi être créées par le programmeur. Elles porteront un nom précis en fonction du langage de programmation (exemple : var, B0, B1, vitesse...)



Les variables sont des éléments qui associent un nom (l'identifiant) à une valeur qui sera implantée dans la mémoire du système programmé. Cette valeur peut être une vitesse, une distance, un temps ... Une variable contient une valeur qui peut varier au cours de l'exécution du programme, comme la couleur des habits d'un personnage, le nombre d'activation d'un capteur, etc.







Notion de variable informatique





La programmation des **objets connectés** nécessite la gestion de **situations complexes** : déplacements, trajectoires, mesures des capteurs... Pour résoudre ces problèmes plus **évolués**, les programmeurs utilisent deux types de **variables informatiques** dans leurs **algorithmes** : **les variables dites « statiques » et « dynamiques »**.

 <u>Les variables statiques</u> sont des valeurs constantes. Elles sont stockées dans la mémoire de l'objet connecté.



Variable statique permettant de régler la vitesse d'un robot.

 <u>Les variables affectées à des capteurs</u> sont dynamiques car leurs valeurs changent dans le temps en fonction de la variation de la mesure du capteur. Ces variables sont stockées dans la mémoire.

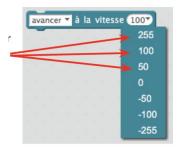


Variable dynamique permettant de stocker la valeur mesurée par le capteur à ultrasons.

• Comment utiliser des variables statiques pour régler la vitesse d'un robot ?

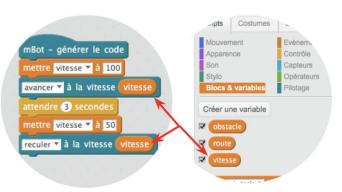
Cas 1 - La plupart du temps, les variables sont prédéfinies dans le logiciel de programmation.

Le programmeur a le choix entre plusieurs valeurs pour régler la vitesse de son robot.



Cas 2 - Les variables peuvent aussi être créées par le programmeur.

Elles **porteront un nom précis** en fonction du langage de programmation (exemple : var, B0, B1, vitesse...)









Notion de variable informatique





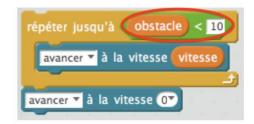
 Comment utiliser des variables dynamiques pour gérer l'arrêt d'un robot devant un obstacle?

Un capteur à ultrasons va mesurer le changement de distance entre le capteur et l'obstacle.

Dans le programme, on stocke la valeur mesurée par le capteur à ultrasons dans la variable «obstacle » qui va varier quand le robot se déplacera.



Quand il sera à moins de 10 cm de l'obstacle, il s'arrêtera.



• Comment utiliser des variables dynamiques pour réaliser le comptage des intersections d'une route ?



- **1 Le robot suit la route** et passe devant les intersections (ruptures de la route).
- 2 Une variable nommée « intersection » sera incrémentée quand le capteur de ligne du robot détectera la rupture de la route.

```
mettre intersection à 0

répéter indéfiniment

si suiveur de ligne Port 2 Côté gauche est blanc et suiveur de ligne Port 2 Côté droit est blanc alor ajouter à intersection 1

avancer à la vitesse 100 attendre 0.3 secondes
```

3 - On compare la variable « intersection » à un nombre précis de carrefour (ici 3) pour donner l'ordre de tourner.



Les variables sont des éléments qui associent un nom (l'identifiant) à une valeur qui sera implantée dans la mémoire du système programmé. Une variable contient une valeur qui peut varier au cours de l'exécution du programme. La variable est statique (temps, vitesse ...) ou dynamique (détection de distances, de couleurs ...) et peut servir au comptage.







Notion de variable informatique





La programmation des **objets connectés** nécessite la gestion de **situations complexes** : déplacements, trajectoires, mesures des capteurs... Pour résoudre ces **problèmes plus évolués**, les programmeurs utilisent **deux types** de **variables informatiques** dans leurs **algorithmes** : les **variables statiques** et **dynamiques**.

 <u>Les variables statiques</u> sont des valeurs constantes. Elles sont stockées dans la mémoire de l'objet connecté.

Variable statique permettant de **régler la vitesse** d'un robot.

avancer ▼ à la vitesse (10▼

 <u>Les variables affectées à des capteurs</u> sont <u>dynamiques</u> car leurs <u>valeurs changent dans</u> <u>le temps</u> en fonction de la <u>variation de la mesure du capteur</u>. Ces variables sont <u>stockées</u> dans la <u>mémoire</u>.

mettre obstacle à distance mesurée par le capteur ultrasons du Port 3

Variable dynamique permettant de stocker la valeur mesurée par le capteur à ultrasons.

• Comment utiliser des variables statiques pour régler la vitesse d'un robot ?

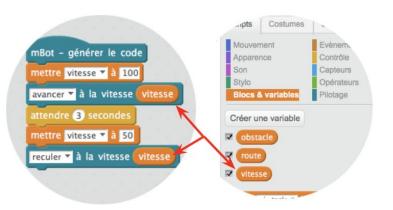
Cas 1 - La plupart du temps, les variables sont prédéfinies dans le logiciel de programmation.

Le programmeur a le choix entre plusieurs valeurs pour régler la vitesse de son robot.



Cas 2 - Les **variables** peuvent aussi **être créées** par le programmeur. Elles **porteront un nom précis** en fonction du langage de

fonction du langage de programmation (exemple : var, B0, B1, vitesse...)









Notion de variable informatique





 Comment utiliser des variables dynamiques pour gérer l'arrêt d'un robot devant un obstacle ?

Un capteur à ultrasons va mesurer le changement de distance entre le capteur et l'obstacle.

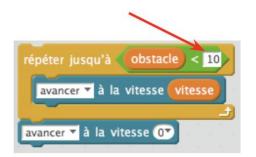
On commence par créer la variable « obstacle » qui permettra de stocker la valeur correspondant à la distance des obstacles mesurée par le capteur à ultrasons.



Puis, dans le programme, on stocke la valeur mesurée par le capteur à ultrasons dans la variable.



On compare ensuite la valeur de la variable avec un seuil correspondant à la distance de l'obstacle à laquelle le robot doit s'arrêter.



La valeur de la variable « obstacle » va varier quand le robot se déplacera.

Quand celui-ci sera à moins de 10 cm de l'obstacle, le robot s'arrêtera.

Les variables sont des éléments qui associent un nom (l'identifiant) à une valeur qui sera implantée dans la mémoire du système. Une variable contient une valeur statique ou dynamique. Cette dernière varie au cours de l'exécution du programme, comme la couleur, la distance, la vitesse...

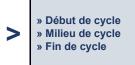






Déclenchement d'une action par un évènement, instructions conditionnelles





Pour résoudre différents problèmes et rendre les systèmes toujours plus autonomes, on utilise un algorithme avec des instructions qui peuvent être exécutées si des événements sont détectés par des capteurs.

 Dans un algorithme ou un programme, les conditions qui déclenchent des actions à partir d'événements sont des instructions conditionnelles. Elles permettent d'exécuter les instructions si les conditions sont vraies ou pas.

Différents événements peuvent être détectés comme :

- la variation d'une grandeur physique (changement de luminosité, de chaleur, de couleur...)
- le déplacement d'un objet mesuré par un capteur du système
- la présence ou non d'une route
- ou simplement un **événement initial**, permettant de lancer le programme.



- Les instructions peuvent être conditionnées par l'apparition d'événements initiaux ou détectés par des capteurs.
 - Déclenchement d'une action par un événement dit initial : "Quand évènement"

Exemple: « quand touche espace est pressée »

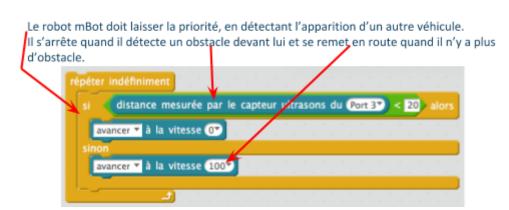
```
quand la touche espace est pressée
régler la DEL de la carte tout en rouge 0 vert 0 bleu 150 
régler la DEL de la carte tout en rouge 150 vert 150 bleu 150
```

Déclenchement d'une action par un événement conditionnel : Si... Alors...Sinon

Exemple : le robot mBot doit détecter un obstacle.



Si obstacle Alors arrêt Sinon avancer Si obstacle Alors arrêt...



Dans un algorithme ou un programme, l'exécution des instructions peut être conditionnée par l'apparition d'un événement. Celles-ci sont des instructions conditionnelles. Dans ce cas, l'instruction s'exécute SI l'événement a lieu, SINON une instruction différente pourra aussi se réaliser.







Séquences d'instructions, boucles





Pour résoudre différents problèmes et rendre les systèmes toujours plus **autonomes**, le programmeur va devoir réaliser **des algorithmes** avec des **séquences d'instructions** et des **boucles**.

Pour écrire un programme, on va utiliser des instructions qui peuvent être exécutées une seule fois ou répétées plusieurs fois.

• Comment réaliser des séquences d'instructions ?

Les actions d'un système peuvent être déclenchées 1 fois en séquences d'instructions sans condition préalable : avancer, tourner à gauche, à droite, reculer...

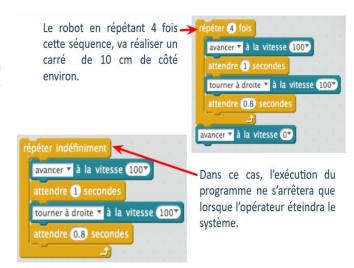




Comment réaliser des boucles ?

Les instructions peuvent être répétées en boucles un certain nombre de fois avant de passer à une autre action ou répétées indéfiniment.





Les instructions d'un algorithme peuvent être déclenchées :

- en séquences (séquences d'instructions) : les ordres sont enchaînés les uns à la suite des autres sans condition préalable (avancer, tourner...)
- et/ou répétées en boucle un nombre précis de fois ou indéfiniment.







Systèmes embarqués



Un système embarqué est un système électronique et informatique autonome, qui est capable de réagir souvent en temps réel et de réaliser des tâches précises.

• Exemples de systèmes embarqués :

Aspirateur robot



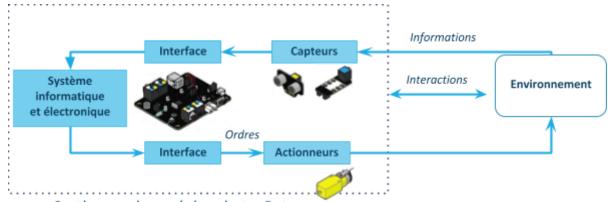
Robot mBot



Voiture autonome



 Le système embarqué du mBot comprend des capteurs, des actionneurs, une interface et un programme stocké dans sa mémoire. Il assure un fonctionnement autonome qui réagit avec son environnement.



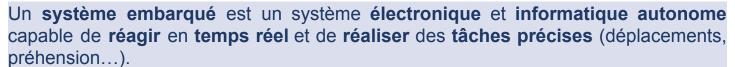
Système embarqué du robot mBot

• Pour que le robot mBot soit **autonome** dans ses déplacements en **temps réel** et donc éviter les obstacles, on va utiliser ses **capteurs ultrasons** grâce à son **programme téléchargé**.

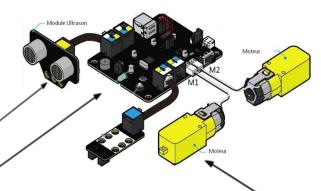
Exemple de fonctionnement du système embarqué du robot mBot :

- 1. L'utilisateur **télécharge** un **programme** dans la mémoire de la carte électronique avec le câble USB.
- Pour éviter l'obstacle, le robot utilise les informations venant de son capteur à ultrasons.
- 3. La carte électronique (avec son programme) à traite ces informations.





Il est intégré dans un objet et permet, à partir de son **ordinateur** (microprocesseur, mémoires, carte mère, alimentation électrique autonome...), de ses **capteurs**, de ses **actionneurs** et d'un **programme stocké** dans sa **mémoire**, d'assurer un **fonctionnement autonome**.









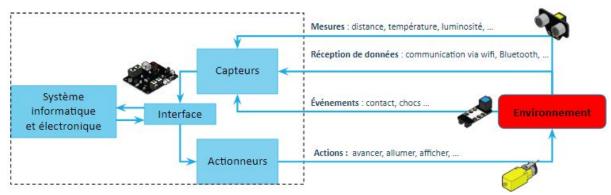
Systèmes embarqués





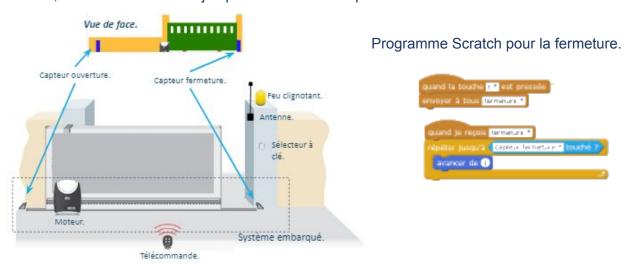
Un système embarqué est un système électronique et informatique autonome, qui est capable de réagir souvent en temps réel et de réaliser des tâches précises.

 Pour que l'objet réagisse en temps réel, le système embarqué comprend des capteurs, des actionneurs, une interface et un système électronique et informatique.



 Pour automatiser un portail, par exemple, le système embarqué comprend un capteur (de contact pour l'ouverture et la fermeture), une interface, un actionneur (moteur), un programme stocké dans la mémoire d'un ordinateur embarqué (exemple ici de programmation sur scratch).

Exemple de système embarqué sur un portail automatique : Les informations proviennent du capteur de contact, le moteur sera activé jusqu'à la fermeture du portail.



Le système embarqué permet aux objets de réaliser des tâches prédéfinies à l'avance (intelligence artificielle faible) ou de rendre l'objet plus autonome, capable « d'apprendre » et de modifier son programme interne (intelligence artificielle forte) comme un robot aspirateur.

Il se compose d'un programme stocké dans la mémoire d'un ordinateur embarqué. En ajoutant des capteurs ou des actionneurs et/ou en modifiant son programme, le système peut modifier son comportement afin de s'adapter à ce qui est attendu de lui. Un système embarqué peut être associé à d'autres, pour permettre à un système plus global d'assurer des fonctions plus complexes.







Forme et transmission du signal





Pour assurer la communication et le fonctionnement des objets connectés, on utilise différentes formes de signaux transmis par différents moyens de transmission.

 Un signal peut prendre différentes formes : 		 Un signal est transmis par différents moyens : 	
Signal électrique		Transmission par fil	Transmission sans fil
Signal lumineux		Fil de cuivre	Ondes infrarouges
Signal sonore		Câble réseau	Bluetooth Wifi
Signal radio	*	Fibre optique Câble USB	Clé Wifi Vibrations mécaniques

- Pour reconnaître la forme et la transmission du signal :
 - 1 On repère l'émetteur, le récepteur et les composants utilisés



2 - Entre la clé WIFI et la carte WIFI :

Forme du signal : signal radio

Transmission du signal : par ondes électromagnétiques

3 - Dans le cordon USB:

Forme du signal : signal électrique (impulsions électriques)

Transmission du signal : par câble



Un **signal** peut prendre **différentes formes** : signal **électrique**, signal **lumineux**, signal **sonore** ou signal **radio**.

Pour transmettre un signal, nous avons deux possibilités :

- soit par fil: fils de cuivre ou fibre optique,
- soit sans fil : ondes électromagnétiques, ondes infrarouges ou vibrations mécaniques.







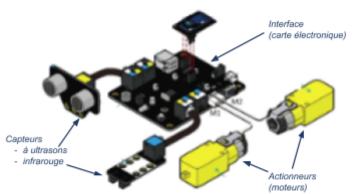
Capteur, actionneur, interface





Pour qu'un **système automatisé** réalise une action, il faut une **interface** qui fait le lien entre les **capteurs** et les **actionneurs**.

- Un capteur réalise l'acquisition d'une grandeur physique (température, luminosité, présence, distance, ...) qu'il transforme en un signal électrique.
- L'interface reçoit les informations des capteurs, les traite et envoie des ordres aux actionneurs.
- L'actionneur transforme l'énergie d'entrée pour réaliser une action.





Pour capter des informations, on peut utiliser différents capteurs :









Capteur à ultrasons

Détecte un obstacle à distance

Capteur infrarouge

Détecte un marquage

Microrupteur

Détecte une ouverture de porte

Détecteur de mouvement

Détecte une présence

• Pour réaliser différentes actions, on peut utiliser différents actionneurs :





Moteur

Transforme l'énergie électrique en mouvement

Buzzer

Transforme l'énergie électrique en son

DEL

Transforme l'énergie électrique en lumière

Un ou des **capteurs** permettent d'acquérir des informations qui sont ensuite traitées par une interface programmable pour piloter un ou des actionneurs qui réalisent l'action (à partir de l'énergie qu'il reçoit).







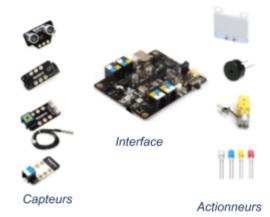
Capteur, actionneur, interface



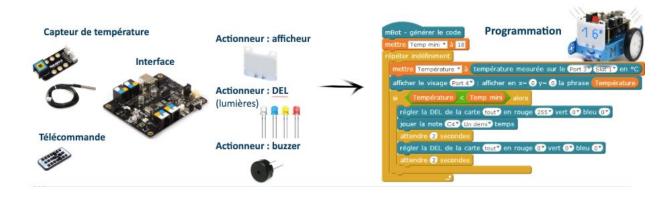


Pour la **programmation d'objets connectés**, nous pouvons utiliser des **systèmes techniques** comportant des **capteurs**, **actionneurs** et **interface**.

- Un capteur réalise l'acquisition d'une grandeur physique (température, luminosité, présence, distance...) pour la transformer en un signal électrique.
- Un actionneur convertit l'énergie qu'il reçoit pour réaliser une action.
- L'interface programmable traite les informations issues des capteurs avec un programme inclus, pour ensuite envoyer des ordres afin de piloter les actionneurs.



 Pour que l'interface traite les données et envoie des ordres, il faut la programmer, ainsi elle transmettra les ordres aux actionneurs en fonction des informations reçues des capteurs.



Un capteur est le composant qui réalise l'acquisition d'une grandeur physique (température, luminosité, présence, distance, ...). L'actionneur est le composant qui réalise une action à partir de l'énergie qu'il reçoit.

Une interface établit la communication entre tous les composants du système programmable. Elle reçoit les informations des capteurs ou de l'homme, effectue des traitements et envoie des ordres aux actionneurs.