

# FORMATION À L'ENSEIGNEMENT DE L'INFORMATIQUE AU COLLÈGE

DAVID VIOLEAU – XAVIER PESSOLES

**Chapitre 0 :** Introduction

**Chapitre 1 :** Architecture d'un système

**Chapitre 2 :** Nature des informations

**Chapitre 3 :** Traitement des informations

**Chapitre 4 :** Communication

**TP 1 :** Jeu Pong interactif

**TP 2 :** Commande d'un moteur CC

**TP 3 :** Jeu Simon

**TP 4 :** Interfacer scratch et Wiimote

**TP 5 :** Interfacer Smartphone et Arduino

**TP 6 :** Initiation réseau

**Fiche 1 :** Liste matériel

**Fiche 2 :** Scratch

**Fiche 3 :** Arduino

**Fiche 4 :** mBlock et Arduino

**Fiche 5 :** Appinventor

# FORMATION À L'ENSEIGNEMENT DE L'INFORMATIQUE AU COLLÈGE

CHAPITRE 0

CHAPITRE 0

## INTRODUCTION – RÉFÉRENCES AU PROGRAMME

### 1 AVANT-PROPOS

#### 1.1 À qui s'adresse cette formation ?

Cette formation a été réalisée pour aider les enseignants de technologie au collège à s'initier à l'informatique et à la programmation de systèmes embarqués. Cependant, elle peut être utilisée par tous les élèves ou tous les adultes désirant découvrir l'informatique.

Ces éléments de formations s'appuient (entre autre) sur le logiciel scratch (et des dérivés). Scratch est un logiciel libre développé par le MIT. Les algorithmes sont représentés sous forme graphique et permettent de réaliser des applications graphiques des plus simples aux plus évoluées.

Au niveau matériel, nous nous appuyons (entre autre) sur des cartes Arduino. Cette formation devrait vous permettre de transférer les compétences acquises sur des robots tels les Moway, shpéro, BB8, les robots proposés par mbot...

#### 1.2 Ce que vous trouverez dans cette formation

Dans cette formation vous trouverez :

- des cours de base sur l'informatique et la programmation ;
- des fiches pour prendre en main les logiciels ou le matériel ;
- des TP pour apprendre à utiliser le matériel seul ou avec les élèves ;
- les exécutables des applications quand ceux-ci existent.

## 2 RÉFÉRENCE AUX NOUVEAUX PROGRAMMES

### 2.1 Extraits des compétences et connaissances

Les programmes de collège font référence à l'informatique en technologie et en mathématiques.

#### Attendus de fin de cycle :

- Écrire, mettre au point et exécuter un programme simple.
- Comprendre le fonctionnement d'un réseau informatique.

Écrire, mettre au point et exécuter un programme	<b>Compétences</b>	
	Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous problèmes afin de structurer un programme de commande.	TP 2, 3, 4
	Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné. <sup>1</sup>	TP 2, 3, 4
	Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs. <sup>1</sup>	TP 1, 2, 3, 4
	Programmer des scripts de déroulant en parallèle. <sup>2</sup>	Fiche 2, TP 4
	<b>Connaissances</b>	
Comprendre le fonctionnement d'un réseau informatique	Notions d'algorithme et de programme. <sup>1</sup>	Ch. 3 TP 3
	Notion de variable informatique. <sup>1</sup>	Ch. 2 et 3 TP 3
	Déclenchement d'une action par un événement, séquences d'instructions, boucles instructions conditionnelles.	Ch. 3 TP 3 et 5
	Systèmes embarqués	Ch 1, 2 & 3 TP 2 et 5
	Forme et transmission du signal	Ch 2 TP 1
	Capteur, actionneur, interface.	Ch. 2 TP 1, 2, 4 et 5
<b>Connaissances</b>		
Comprendre le fonctionnement d'un réseau informatique	Composants d'un réseau, architecture d'un réseau local, moyens de connexion d'un moyen informatique.	Ch. 4 TP 6
	Notion de protocole, d'organisation de protocoles en couche, d'algorithme de routage Internet.	Ch. 4 TP 6
		Ch. 4 TP 6

<sup>1</sup>: Compétences communes avec le programme de mathématiques.

<sup>2</sup>: Compétences au programme de mathématiques.

### 2.2 Exemples d'activités proposées dans le programme

#### 2.2.1 Écrire, mettre au point, exécuter un programme

- Concevoir, paramétriser, programmer des applications informatiques pour des appareils nomades.
- Observer et décrire le comportement d'un robot ou d'un système embarqué. En décrire les éléments de sa programmation
- Agencer un robot (capteurs, actionneurs) pour répondre à une activité et un programme donnés.
- Écrire, à partir d'un cahier des charges de fonctionnement, un programme afin de commander un système ou un système programmable de la vie courante, identifier les variables d'entrée et de sortie.
- Modifier un programme existant dans un système technique, afin d'améliorer son comportement, ses performances pour mieux répondre à une problématique donnée.
- Les moyens utilisés sont des systèmes pluritechnologiques réels didactisés ou non, dont la programmation est pilotée par ordinateur ou une tablette numérique. Ils peuvent être complétés par l'usage de modélisation numérique permettant des simulations et des modifications du comportement.

### 2.2.2 Réseau informatique

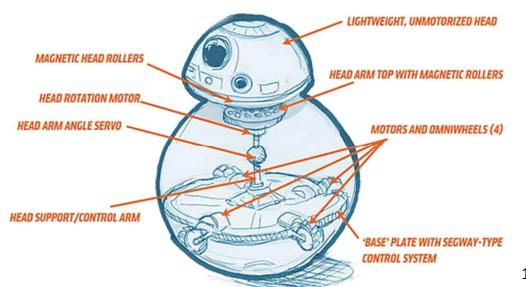
- Observer et décrire sommairement la structure du réseau informatique d'un collège, se repérer dans ce réseau.
- Exploiter un moyen informatique diversifié dans différents points du collège.
- Simuler un protocole de routage dans une activité déconnectée.

# FORMATION À L'ENSEIGNEMENT DE L'INFORMATIQUE AU COLLÈGE

INFORMATIQUE  
COLLÈGE

## CHAPITRE 1

### ARCHITECTURE D'UN SYSTÈME PLURITECHNOLOGIQUE



<b>1</b>	<b>Chaîne fonctionnelle d'un système pluritechnologique .....</b>	<b>2</b>
<b>2</b>	<b>Focus sur la chaîne d'information .....</b>	<b>3</b>
2.1	Nature des informations.....	3
2.2	Traitement des informations.....	3
2.3	Communication des informations .....	3
<b>3</b>	<b>Notion de systèmes embarqués .....</b>	<b>4</b>
<b>4</b>	<b>La carte Arduino .....</b>	<b>5</b>

<sup>1</sup> <http://jalopnik.com/the-new-star-wars-droid-is-not-cgi-so-how-does-it-work-1698461524>

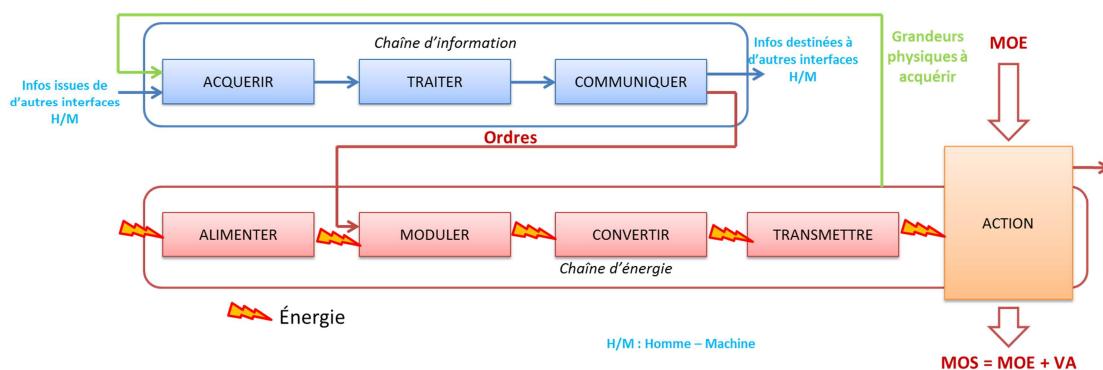
# 1 CHAÎNE FONCTIONNELLE D'UN SYSTÈME PLURITECHNOLOGIQUE

## Définition – Système pluritechnologique:

Un système pluritechnologique est composé de différents sous-ensembles issus de différentes technologies : électronique, informatique, automatique, mécanique, etc. Ces différents sous-ensembles doivent interagir pour assurer un service à l'utilisateur.

Il existe plusieurs moyens de décrire tel système pluritechnologique. Dans une première approche, il est possible d'utiliser la chaîne fonctionnelle, permettant de distinguer une chaîne d'information, une chaîne d'énergie et permettant de représenter l'action d'un système sur la matière d'œuvre.

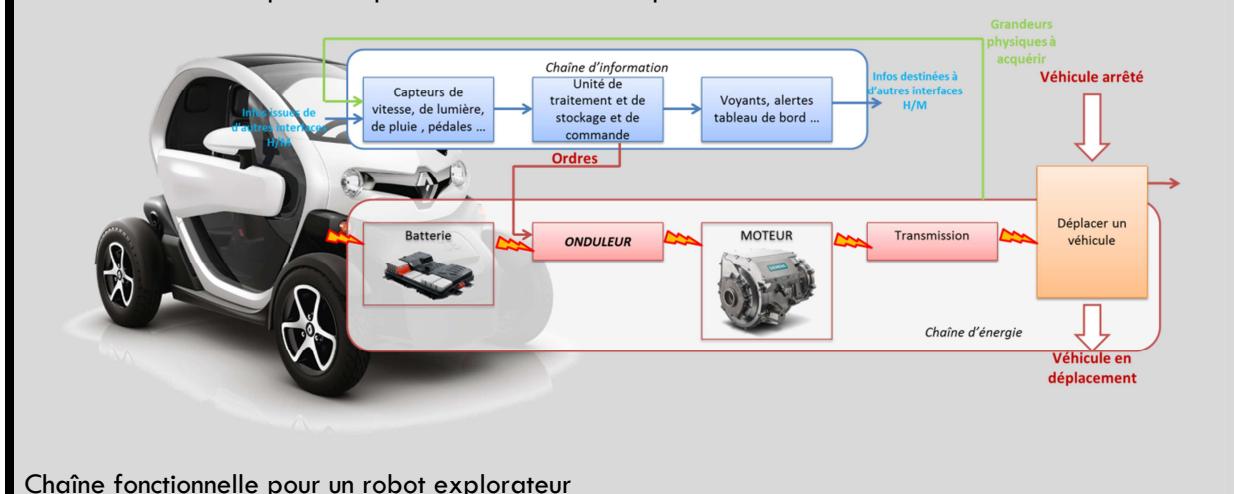
La chaîne d'énergie permet de mettre en évidence les sous-systèmes permettant de réaliser des transformations sur les transferts de matière et d'énergie d'un système. La chaîne d'information permet de mettre en évidence les sous-ensembles permettant de réaliser des transformations des informations.



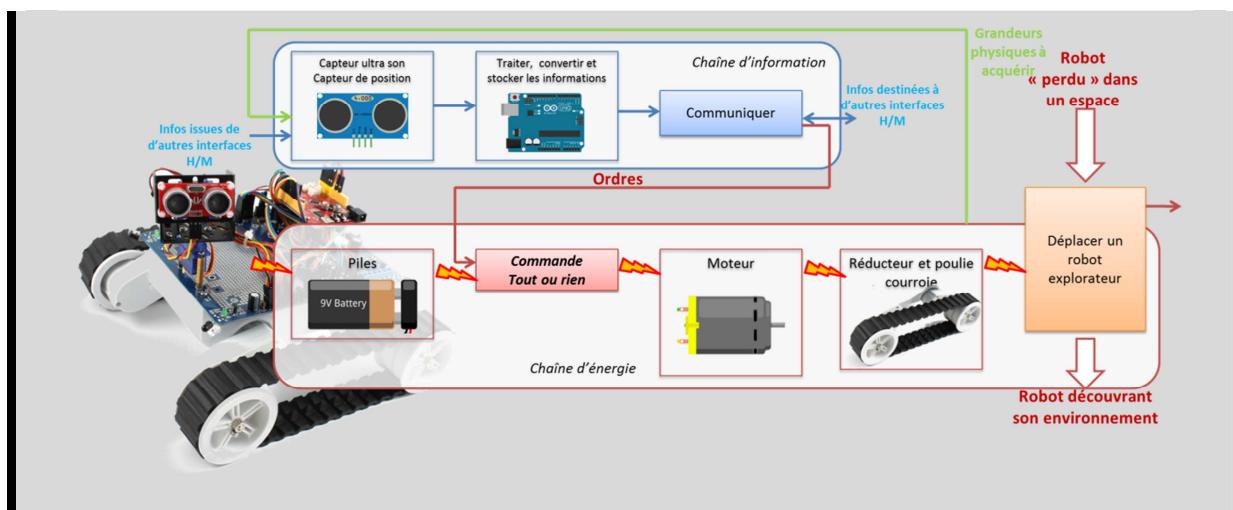
Cette représentation n'étant pas normalisée, il est possible de prendre des libertés pour représenter un système.

## Exemples

### Chaîne fonctionnelle partielle pour une voiture électrique

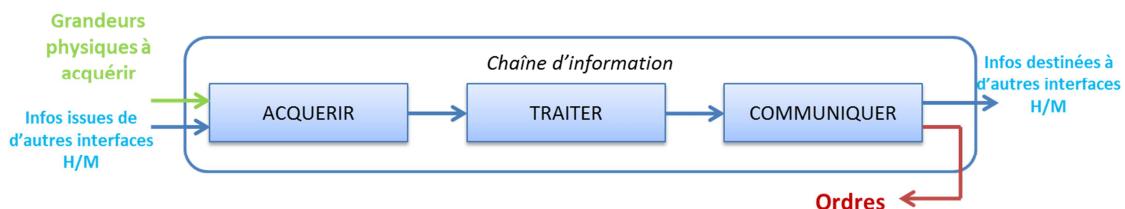


### Chaîne fonctionnelle pour un robot explorateur



## 2 FOCUS SUR LA CHAÎNE D'INFORMATION

Dans le cadre de l'informatique, on va se focaliser sur la chaîne d'information.



### 2.1 Nature des informations

On commencera par s'intéresser à la nature des informations transitant dans un système :

- Informations logiques (ou binaires) (0 ou 1);
- Informations numériques : il s'agit d'un type d'informations où une taille fixe de donnée peut être traitée. Les informations sont discrétisées;
- Informations analogique (température, tension) : les informations peuvent prendre des valeurs continues sur un intervalle donné.

Cela nous permettra d'abord définir quels types de données on peut acquérir avec un système (données d'entrées – input) et quels types de données on peut utiliser pour piloter un système (données de sorties – output).

On s'intéressera ensuite à la gestion de ces entrées et de ces sorties sous forme de variables par un logiciel.

### 2.2 Traitement des informations

Ce chapitre permettra de s'intéresser à la façon dont sont traitées les informations par un système. En effet, lors de cette phase il est possible de :

- Stocker l'information ;
- Traduire l'information ou adapter l'information pour qu'elle soit utilisable par l'utilisateur,
- Modifier et gérer l'information grâce à des algorithmes.

### 2.3 Communication des informations

Un système moderne est un système qui communique. Ce système peut communiquer avec un ordinateur, un téléphone, une tablette. La nature de la communication peut différer suivant les interfaces disponibles (éthernet, WiFi, Bluetooth).

Nous nous intéresserons donc à :

- la structure d'un réseau ;
- la transmission sans fil.

### 3 NOTION DE SYSTÈMES EMBARQUÉS

#### Définition – Système embarqué – Partie commande

Un système embarqué est un système électronique et informatique dont le but est de traiter en temps réel des informations provenant de capteurs ou de la mémoire dans le but de communiquer des informations ou d'envoyer des ordres à la chaîne d'énergie.

#### Définition – Partie opérative

On peut définir la partie opérative comme étant les solutions techniques permettant de réaliser les fonctions convertir et transmettre.

#### Exemples



Attrapeur de bonbons



Robotic Arm Edge



Lego Mindstorm

#### Définition – Partie logicielle

On peut définir la partie commande comme l'ensemble des programmes qui vont être mis en œuvre afin que le système réalise l'objectif fixé.

#### Exemples

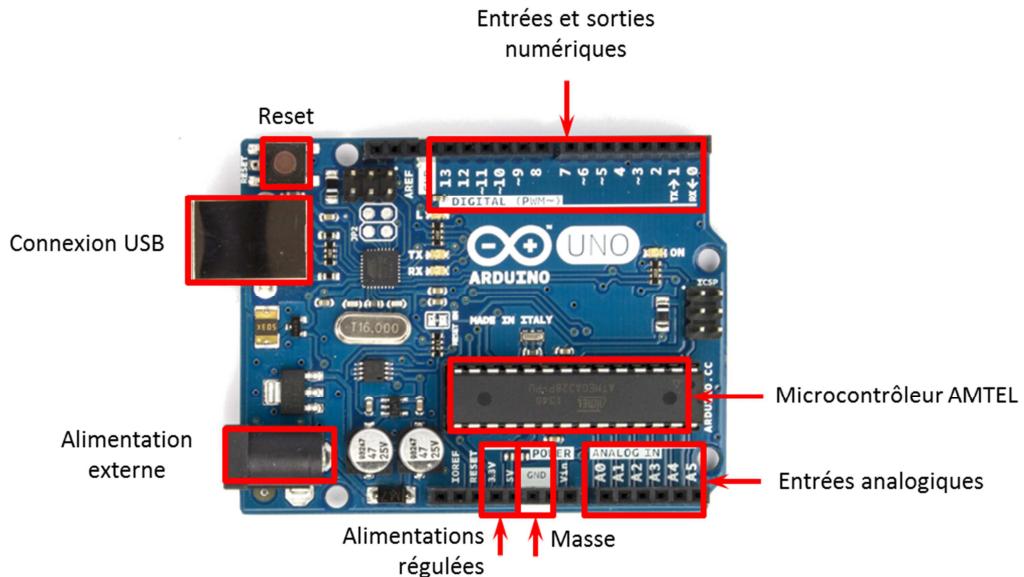


Programmation graphique d'un système embarqué  
avec Ardublock



Programmation graphique avec scratch

## 4 LA CARTE ARDUINO



Les caractéristiques de la carte Arduino UNO sont les suivantes :

- Mémoire et microcontrôleur :
  - Microcontrôleur ATmega328 cadencé à 16 MHz
  - Mémoire Flash 32ko (dont 0,5 ko pour le système d'amorçage)
  - SRAM : 2ko
  - EEPROM 1ko
- Entrées analogiques :
  - 5 entrées analogiques
- Entrées sorties numériques
  - 14 dont 6 en MLI (PWM) indiquées ~ (40mA).
  - Ports Tx et Rx : reprise du port série
- Alimentation :
  - Alimentation par le port USB : 5V, 500mA
  - Alimentation externe en 7 à 12 V (2,1 mm)
  - Reprise de l'alimentation externe
  - Alimentation externe régulée en 5V/500mA et 3,3 V/50mA.

# FORMATION À L'ENSEIGNEMENT DE L'INFORMATIQUE AU COLLÈGE

INFORMATIQUE  
COLLÈGE

## CHAPITRE 3

### TRAITEMENT DES INFORMATIONS

#### Connaissances

##### Algorithmique et programmation

Programme de  
mathématiques

Notion d'algorithme et de programme

Notion de variable informatique

Déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles

##### Écrire, mettre au point et exécuter un programme

Programme de  
technologie

Notion d'algorithme et de programme

Notion de variable informatique

Déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles

Systèmes embarqués



Programmation graphique  
avec scratch

<b>1</b>	<b>Traitements de l'information .....</b>	<b>2</b>
1.1	Présentation .....	2
1.2	Exécution d'un programme sur ordinateur .....	2
1.3	Exécution d'un programme sur une cible .....	2
<b>2</b>	<b>Structures algorithmiques .....</b>	<b>2</b>
2.1	Blocs d'instructions .....	3
2.2	Répétition – Boucle infinie .....	3
2.3	Gestion événementielle élémentaire .....	3
2.4	Structures conditionnelles – Si, Sinon, Alors .....	4
2.5	Structures itératives – Tant que, Faire – Répéter n fois ... .....	4
<b>3</b>	<b>Fonction-Procédure .....</b>	<b>4</b>
3.1	Procédure .....	4
3.2	Fonction .....	5

## 1 TRAITEMENT DE L'INFORMATION

### 1.1 Présentation

On s'intéresse ici à la fonction « Traiter ». Celle-ci est généralement réalisée par une carte électronique, ou par le processeur de l'ordinateur, d'une tablette ou d'un téléphone. Cette fonction est essentielle car c'est elle qui permet de traiter des entrées, de commander des sorties et de communiquer avec l'utilisateur ou l'environnement.



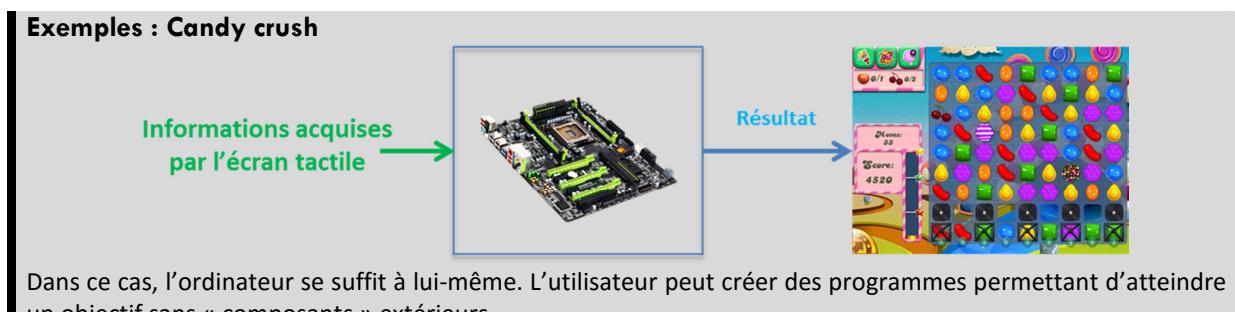
#### Exemples :



### 1.2 Exécution d'un programme sur ordinateur

Dans le cadre de l'utilisation d'un ordinateur, le processeur de la carte mère traite des informations qu'il a déjà en mémoire, en interaction avec des informations du clavier de la souris (ou de l'écran tactile). Les informations traitées peuvent être affichées sur l'écran ou diffusées par les enceintes.

#### Exemples : Candy crush

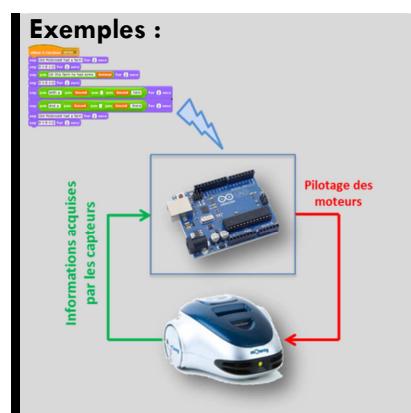


Dans ce cas, l'ordinateur se suffit à lui-même. L'utilisateur peut créer des programmes permettant d'atteindre un objectif sans « composants » extérieurs.

### 1.3 Exécution d'un programme sur une cible

Dans le cadre d'un système embarqué, on peut considérer :

- la phase de développement où le système est relié à l'ordinateur afin de mettre au point un programme ;
- la phase d'utilisation où le programme est terminé. Il est alors « téléversé » dans la carte du système. Celui-ci peut être déconnecté de l'ordinateur et fonctionner de manière autonome.



## 2 STRUCTURES ALGORITHMIQUES

Afin de présenter les différentes structures algorithmiques, nous allons utiliser scratch et réaliser une implémentation du jeu « pong », l'un des premiers jeux à avoir été développé sur un ordinateur.

## 2.1 Blocs d'instructions

### Définitions :

#### Expression

Une expression est l'évaluation d'un calcul. Un résultat est retourné.

#### Instruction

Une instruction est une action utilisée dans un algorithme. Une instruction peut inclure une expression. Lorsqu'on écrit une suite d'instructions successives on parle de **bloc d'instructions**.

### Exemples – Scratch



Instructions réalisables avec scratch



Bloc d'instructions

### Exemple – Pong

Une des premières étapes consiste à gérer le comportement de la balle seule.

Un mouvement de la balle correspond à un déplacement de 10 pixels. De plus, on souhaite qu'elle rebondisse quand elle rencontre un mur.



Le problème est qu'il est nécessaire relancer indéfiniment le script pour que la balle évolue dans le cadre.

## 2.2 Répétition – Boucle infinie

### Définition :

Structure algorithmique permettant de répéter un nombre infini de fois des instructions.

### Exemple – Pong

On va maintenant faire en sorte que la balle avance « toute seule » et qu'elle rebondisse sur les bords.

Avec le premier bloc d'instructions, on se rend compte que la balle ne se déplace que de droite à gauche puis de gauche à droite indéfiniment.

Pour qu'elle couvre l'écran, il va falloir réaliser une **initialisation**.

#### Initialisation



#### Évolution



#### ⇒ Lien à faire avec le programme de mathématiques

⇒ Thème D : espace et géométrie – (Se) repérer sur une droite graduée, dans le plan muni d'un repère orthogonal, dans un parallélépipède rectangle ou sur une sphère.

## 2.3 Gestion événementielle élémentaire

### Définition :

La programmation événementielle permet de prendre en compte des événements pouvant apparaître aléatoirement et d'interférer sur le déroulement du programme.

Cela permet donc de prendre en compte un clic de souris, un appui sur une touche du clavier etc.

### Exemple – Pong

On va créer un événement déclencheur pour démarrer le jeu. On va pour cela utiliser le drapeau.



Dans le cadre de l'utilisateur d'une carte électronique comme organe de traitement, les signaux émis par les capteurs peuvent être utilisés comme événement faisant évoluer le programme (exemple : capteur ultrason détectant un mur).

## 2.4 Structures conditionnelles – Si, Sinon, Alors

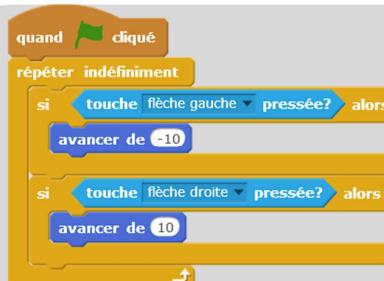
### Définition :

Les instructions conditionnelles permettent d'effectuer une instruction (ou un bloc d'instruction) si une condition est vraie.

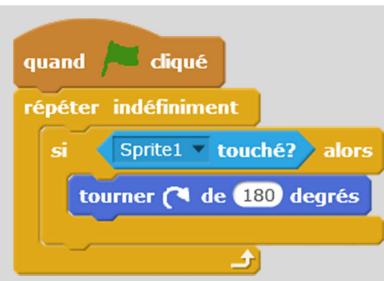
### Exemple – Pong

On va utiliser les itérations infinies, les instructions conditionnelles et les événements pour gérer le deuxième acteur du jeu à savoir la raquette.

L'instruction conditionnelle permet aussi de gérer le mouvement de balle sur la raquette.



Mouvement de la raquette



Mouvement de la balle sur la raquette

L'instruction Si « Condition » Alors « Instruction » nécessite l'emploi d'une condition dite booléenne qui peut être vraie ou fausse. Dans le cas où la condition est vraie, l'instruction située après Alors est exécutée.

### Définition :

Si la condition est fausse et qu'une instruction doit être réalisée, on utilise la commande :

**Si « Condition » Alors « Instruction1 »**

**Sinon « Instruction2 »**

Plusieurs conditions peuvent être imbriquées les unes dans les autres.

## 2.5 Structures itératives – Tant que, Faire – Répéter n fois ...

### Définition :

Une instruction itérative permet de répéter un bloc d'instructions un nombre défini de fois.

### Exemple



On peut utiliser une variable qui se modifie automatiquement à chaque itération.

## 3 FONCTION-PROCÉDURE

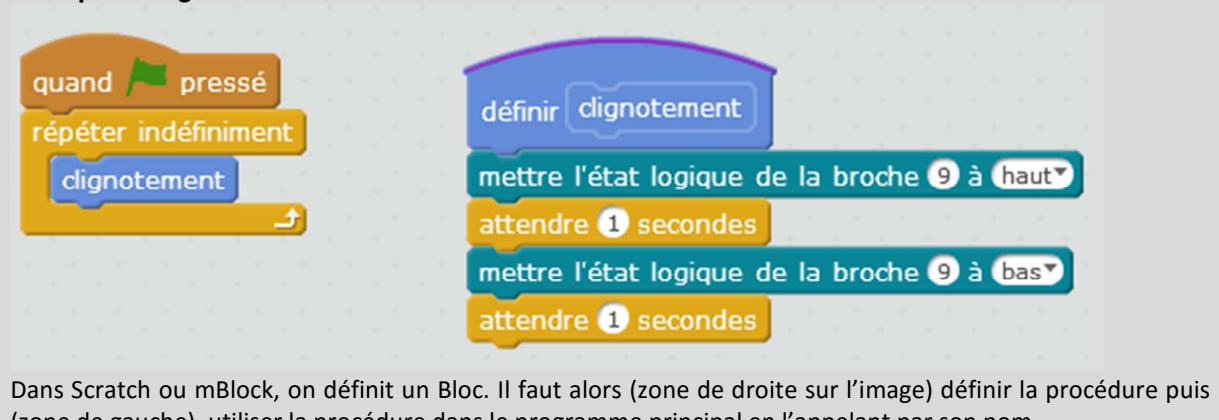
### 3.1 Procédure

#### Définition :

Une procédure est une portion de code, un sous-programme, effectuant une ou plusieurs tâches indépendamment du reste du programme.

C'est une boîte noire pour celui qui utilise cette procédure : il n'est pas nécessaire de savoir ce qu'il y a à l'intérieur ou comment est faite la procédure, il faut juste savoir ce que fait cette procédure.

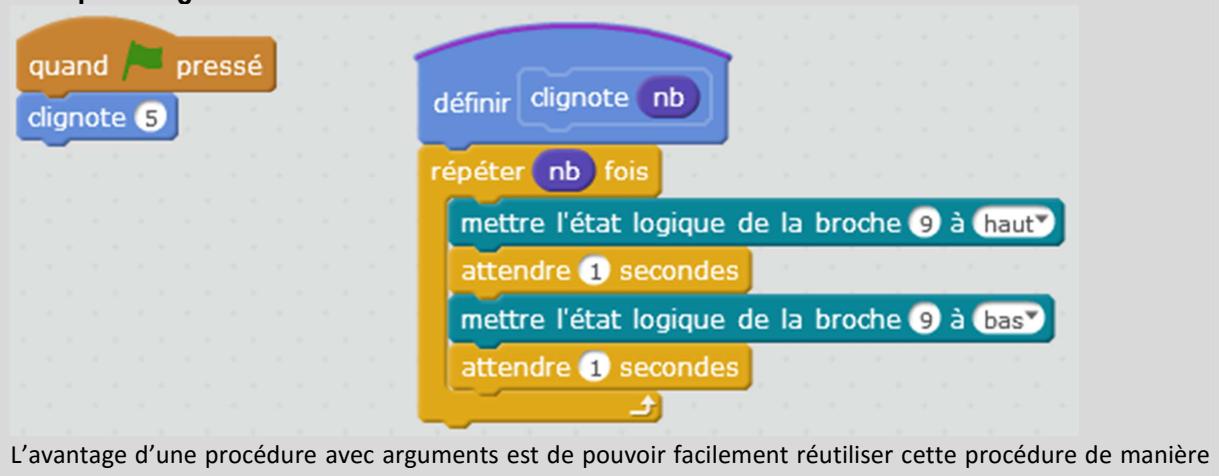
#### Exemple – Clignotement d'une led



Dans Scratch ou mBlock, on définit un Bloc. Il faut alors (zone de droite sur l'image) définir la procédure puis (zone de gauche), utiliser la procédure dans le programme principal en l'appelant par son nom.

Une procédure peut comporter ou non des paramètres d'entrée qui sont utilisés à l'intérieur de celle-ci.

#### Exemple – Clignotement d'une led n fois



L'avantage d'une procédure avec arguments est de pouvoir facilement réutiliser cette procédure de manière un peu différente en fonction des arguments.

## 3.2 Fonction

### Définition :

Une fonction est une procédure mais qui réalise en général un calcul et qui renvoie une ou plusieurs valeurs à la fin de son exécution.

Aucune différence n'est faite entre procédure et fonction dans Scratch, on parle alors de blocs d'instructions avec ou non des paramètres d'entrée.

## CHAPITRE 2

## NATURE DES INFORMATIONS TRANSITANT DANS UN SYSTÈME

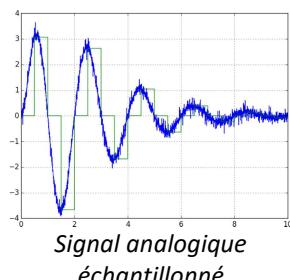
### Connaissances

Forme et transmission du signal

Systèmes embarqués

Notion de variable informatique

Capteur



<b>1</b>	<b>Nature des informations.....</b>	<b>2</b>
<b>1.1</b>	<b>Informations analogiques .....</b>	<b>2</b>
1.1.1	Quelques capteurs .....	2
1.1.2	Notion de gain.....	2
1.1.3	Notion d'offset.....	3
<b>1.2</b>	<b>Informations logiques .....</b>	<b>3</b>
1.2.1	Conversion numérique en binaire.....	3
<b>1.3</b>	<b>Informations numériques .....</b>	<b>4</b>
1.3.1	Notions d'échantillonnage.....	4
1.3.2	Notions de quantification .....	5
1.3.3	Notion d'espace mémoire et de temps de calcul.....	5
<b>2</b>	<b>Entrées sorties d'un système embarqué.....</b>	<b>5</b>
<b>3</b>	<b>Variables logicielles .....</b>	<b>6</b>

# 1 NATURE DES INFORMATIONS

## 1.1 Informations analogiques

### Définition : grandeur analogique

Une grandeur analogique est une grandeur physique (tension, son, vitesse,...) qui évolue de manière continue en fonction du temps.

### 1.1.1 Quelques capteurs

Les grandeurs analogiques sont mesurées par des instruments de mesure ou des capteurs :

- tachymètre (mesure de vitesse d'un solide en rotation) ;
- voltmètre ;
- luxmètre (mesure de l'intensité lumineuse);
- thermomètre, thermistance ;
- jauge d'extensométrie (mesure d'un effort ou d'un couple) ;
- potentiomètre (mesure d'une position angulaire ou linéaire) ;
- capteur ultra son ;
- microphone...

### 1.1.2 Notion de gain

Beaucoup de capteurs permettent de transformer une grandeur analogique en une tension en Volt (souvent entre 0 et 5 volts ou -5 et 5 volts. Ce capteur est caractérisé par un gain.

### Définition : gain d'un capteur

Facteur permettant de faire le lien entre l'unité de sortie et l'unité d'entrée. Ainsi, en termes d'unités, on a :

$$[\text{Unité du gain}] = \frac{[\text{Unité de sortie}]}{[\text{Unité d'entrée}]}$$

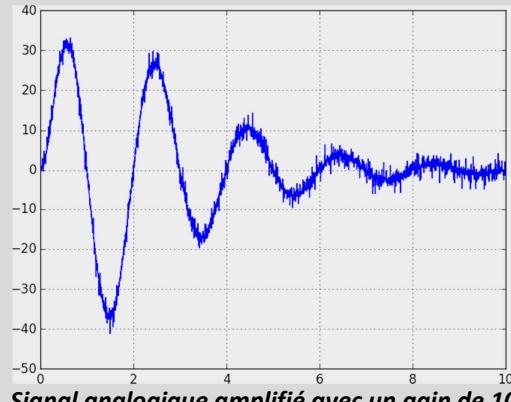
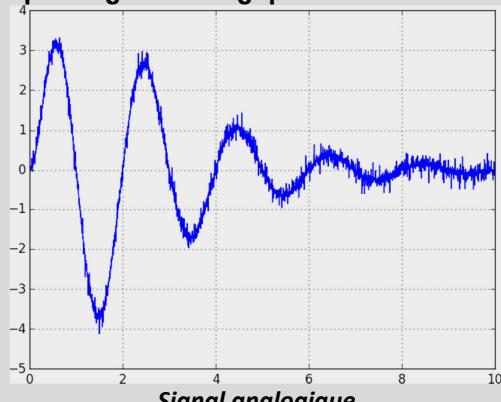
### Exemple : Potentiomètre rotatif

Les potentiomètres rotatifs sont des capteurs que l'on peut retrouver sur des interrupteurs variateurs de lumière ou sur des tables de mixage pour modifier le volume, les basses etc...



L'unité du capteur sera en [Volts/Degrés].

### Exemple : Signal analogique



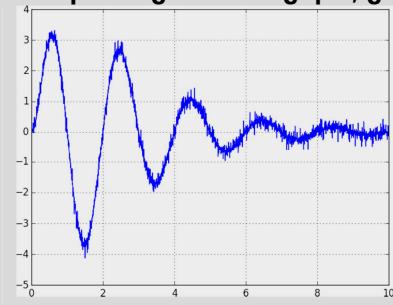
### 1.1.3 Notion d'offset

Certains capteurs ne sont pas purement linéaires. Cela signifie que l'entrée et la sortie ne sont pas linéaires, mais qu'il existe un décalage.

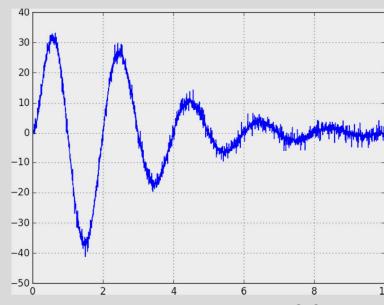
Ainsi, si l'offset et le gain du capteur sont connus (données constructeurs ou détermination expérimentale), on aura :

$$\text{Sortie} = \text{Entrée} \times \text{Gain} + \text{Offset}.$$

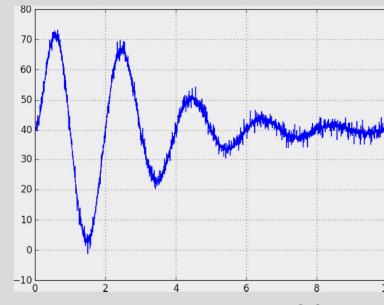
#### Exemple : Signal analogique, gain et offset



Signal analogique



Signal analogique amplifié avec  
un gain de 10



Signal analogique amplifié avec  
un gain de 10 et un offset de 40

## 1.2 Informations logiques

### Définition : grandeur logique

Une grandeur logique est une grandeur ne pouvant prendre que deux valeurs (0 ou 1, Vrai ou Faux, Tout ou Rien...). On parle de grandeur booléenne.

Les grandeurs logiques sont nécessaires pour (au moins) deux raisons :

- Gérer des informations binaires (interrupteur ouvert ou fermé) ;
- Coder les informations numériques en vue d'être communiquées, traitées ou stockées. En effet, les composants électroniques et les ordinateurs ne sont capables que de traiter des informations binaires ;
- Gérer des informations événementielles (clic de souris, ...)

#### Exemple : Bouton poussoir



### Remarque :

Pour faire transiter une valeur binaire dans une carte électronique, on utilise des niveaux de tensions, par exemple : un signal de 1 (ou interrupteur) serait codé par un signal de 5 Volts, le signal 0 serait codé par un signal de -5 Volts.

### 1.2.1 Conversion numérique en binaire

La conversion numérique en binaire ne fait pas l'objet de cette formation. Elle se réalise par une succession de divisions par 2.

Code décimal	0	1	2	3	4	5	6	7
Code binaire	000	001	010	011	100	101	110	111

## 1.3 Informations numériques

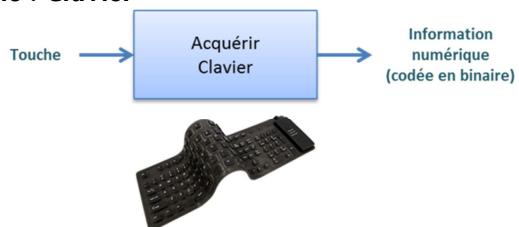
### Définition : grandeur numérique

Une grandeur numérique est une grandeur quantifiée. Cela signifie que la grandeur peut prendre un nombre limité de valeurs. Chaque valeur numérique est décrite par un ensemble de bits (0 et 1).

Les informations numériques sont nécessaires car les grandeurs analogiques sont des nombres réels. Ces derniers ayant une infinité de décimale, il ne serait pas possible de les stocker. Il faut donc les « discréteriser ».

Certains capteurs permettent de délivrer directement une valeur numérique c'est le cas des codeurs absolus. Souvent les grandeurs analogiques sont converties en grandeurs numériques grâce à des convertisseurs analogiques-numériques.

### Exemple : Clavier



Il existe une table de vérité entre les lettres d'un clavier et le signal numérique en « sortie du clavier ».

### 1.3.1 Notions d'échantillonnage

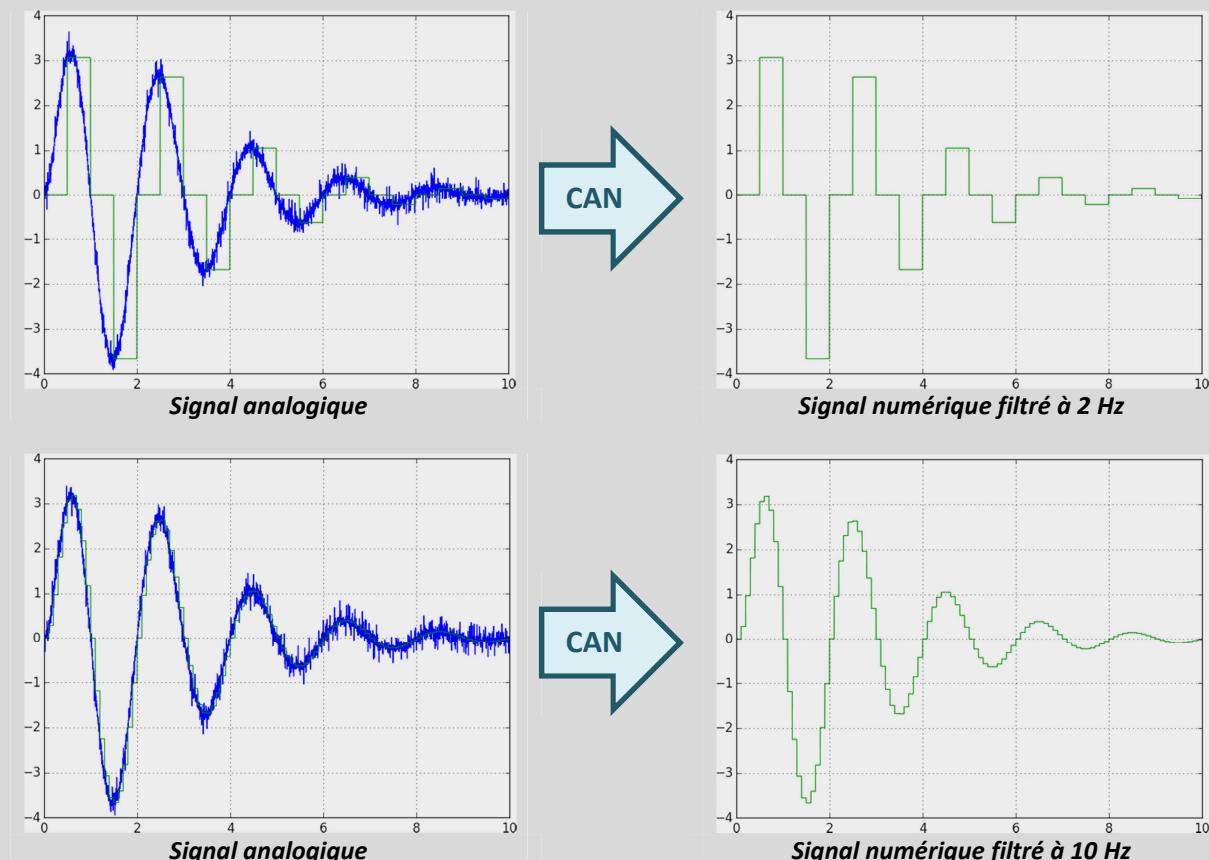
Pour convertir une donnée analogique en donnée numérique, il faut d'abord discréteriser le temps, cela signifie qu'un capteur va mesurer des informations en continu, mais que ces informations seront prélevées par pas de temps.

### Définition : fréquence d'échantillonnage

On définit la fréquence d'échantillonnage comme le nombre de mesures que l'on va prendre par unité de temps. La fréquence d'échantillonnage s'exprime en Hertz (Hz).

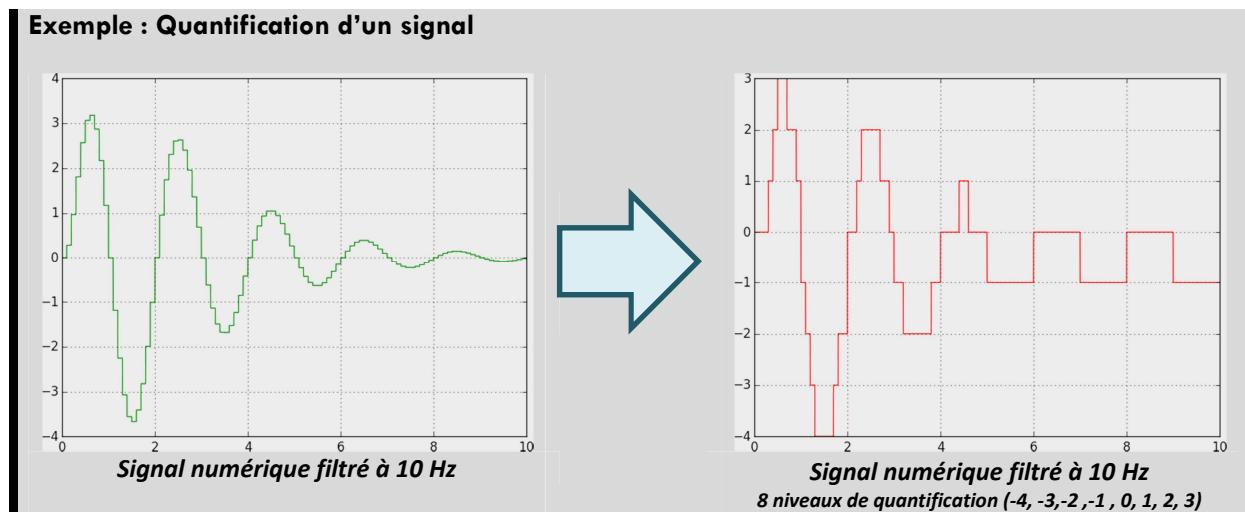
Pour un échantillon de 10Hz on réalisera 10 mesures en 1 seconde.

### Exemple : Signal analogique et numérique – Convertisseur Analogique Numérique (CAN)



### 1.3.2 Notions de quantification

Suivant les performances des systèmes d'acquisition et de traitement des informations des problèmes de quantifications peuvent apparaître. La quantification est une discréétisation de la valeur du signal.



Un nombre codé sur 8 bits correspondra à des valeurs numériques constituées de 8 valeurs binaires 0 et 1.  
Exemple 255 = 11111111.

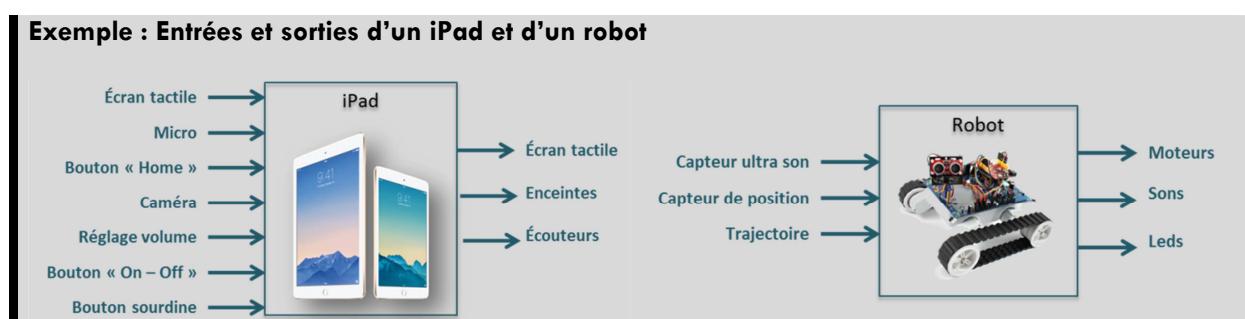
### 1.3.3 Notion d'espace mémoire et de temps de calcul

Si on augmente le temps d'échantillonnage et la valeur de quantification, le signal numérisé devient de plus en plus fidèle au signal d'origine. En revanche, les données à stocker sont de plus en plus nombreuses et de plus en plus grandes. Il sera donc plus difficile de les stocker et les calculs lors du traitement pourront être plus lents.

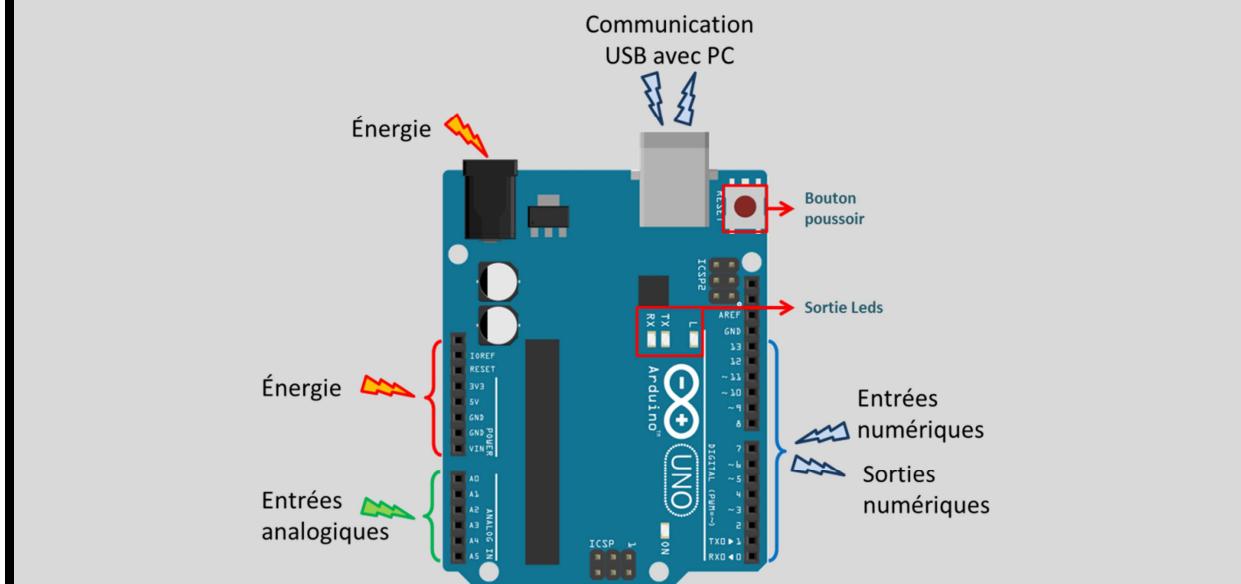
## 2 ENTRÉES SORTIES D'UN SYSTÈME EMBARQUÉ

Afin de pouvoir communiquer, un système reçoit des informations d'entrées via les fonctions acquérir ou communiquer. Ces informations peuvent provenir de capteurs, de l'utilisateur ou de d'autres systèmes.

Après les avoir traitées, le système restitue des informations afin de communiquer avec l'extérieur ou de commander la chaîne d'énergie (allumer des lumières, commander des moteurs, afficher des informations sur un écran...).



### Exemple : Entrées et sorties d'une carte Arduino Uno



## 3 VARIABLES LOGICIELLES

Pour être traitées par un ordinateur ou une carte électronique, les grandeurs numériques ou logiques sont stockées dans des variables.

### Définition : variable

Une variable permet de stocker des informations dans un espace mémoire. Elle est caractérisée par :

- Un identificateur (nom de la variable) ;
- Un type (nombre entier, booléen, chaîne de caractères...) ;
- Une valeur ;
- Une référence (adresse dans la mémoire) ;
- Des opérations.

### Exemples :

#### Variable de type entier

La variable peut prendre « toutes » les valeurs de  $\mathbb{Z}$  : 0, 1, 2, 3, 4..10..100.. -1, -9, -98....

Parmi les opérations possibles on peut noter l'addition, la soustraction, la division, la multiplication.

#### Variable de type booléen

Une variable booléenne ne peut prendre que les valeurs 0 ou 1 (Vrai ou Faux)

« Addition » booléenne	« Multiplication » booléenne
$0 + 0 = 0$	$0 \cdot 0 = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$
$1 + 1 = 1$	$1 \cdot 1 = 1$

#### Variable de type réel

La variable peut prendre n'importe quelle valeur : 0.1, 0.23, 10.46, -52.45...

La précision des nombres dépend de la précision de la machine

#### Variable de type texte ou chaîne de caractères

Exemple : « Bonjour »

La taille en mémoire dépend de l'encodage des caractères. (ordre de grandeur, 1 octet par caractère).

Normalement, on ne peut faire des opérations qu'entre variables du même type. Dans certains langages des conversions automatiques sont réalisées pour pourvoir passer d'un type à l'autre.

### Exemples :

« 3 » + « 5 » = « 35 » mais  $\text{int}(\text{« 3 »}) + \text{int}(\text{« 5 »}) = 8$

Dans le premier cas, on parle de concaténation de deux caractères. Dans le second cas, les chaînes sont converties en entier puis sommées.

### Cas de Scratch ou mBlock

- Dans ce logiciel, on définit les variables dans l'onglet Blocs et variables puis en cliquant sur Créer une variable :
- Ces variables n'ont pas de type vraiment bien défini. Ce peut être des entiers, ou réel ou texte. Le logiciel s'occupe automatiquement de convertir les valeurs.
- Les formes utilisées dans les différentes instructions permettent de voir quel type d'information est nécessaire :



: rond pour une valeur numérique (entier ou réel)



: hexagone pour une valeur booléenne (condition)



: carré pour une valeur de type texte

Lorsque l'on souhaite utiliser plusieurs variables qui jouent le même rôle (par exemple, on doit stocker les résultats de différents joueurs pour les afficher dans un tableau de score), on utilise des listes.

### Définition : liste

- Une liste est une structure de données permettant de regrouper des données de manière à pouvoir y accéder librement.

### Exemples : Meilleurs chronos pour le jeu pong (s)

Scores=[100,250,600,732,960]

On accède à un élément de la liste en demandant sa position ou indice.

### Exemples :

La valeur 250 correspond à l'élément d'indice 2 dans Scores.

	Variable Scores				
Indices	1	2	3	4	5
Valeur	100	250	600	732	960

Différentes opérations sont alors possibles :

- obtention de la longueur de la liste (du nombre d'éléments) ;
- obtention du dernier élément ;
- suppression d'un élément donné par son indice ;
- insertion d'un élément donné par son indice ;
- remplacement d'un élément donné par un autre ;
- recherche d'un élément dans la liste.

# FORMATION À L'ENSEIGNEMENT DE L'INFORMATIQUE AU COLLÈGE

INFORMATIQUE  
COLLÈGE

## CHAPITRE 4

### COMMUNICATION ENTRE LES SYSTÈMES

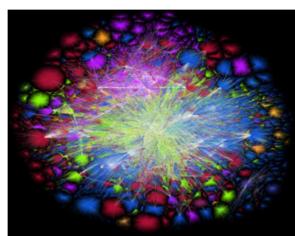
#### Connaissances

##### Comprendre le fonctionnement d'un réseau informatique

Programme de  
technologie

Composants d'un réseau, architecture d'un réseau local, moyens de connexion d'un moyen informatique.

Notion de protocole, d'organisation de protocoles en couche, d'algorithme de routage  
Internet



Représentation du Web  
<http://www.opte.org/>

<b>1</b>	<b>Communication USB – Série .....</b>	<b>2</b>
1.1	<b>Composition du câble .....</b>	<b>2</b>
1.2	<b>Communication USB.....</b>	<b>2</b>
<b>2</b>	<b>Communication dans un réseau « internet » .....</b>	<b>2</b>
2.1	<b>Client – Serveur.....</b>	<b>2</b>
2.2	<b>Architecture d'un réseau local.....</b>	<b>3</b>
2.3	<b>Communication dans un réseau.....</b>	<b>3</b>
2.4	<b>Routage .....</b>	<b>3</b>
<b>3</b>	<b>Communication réseau sans fil .....</b>	<b>4</b>
3.1	<b>Le protocole Bluetooth.....</b>	<b>4</b>

## 1 COMMUNICATION USB – SÉRIE

Un des moyens les plus aisés de communiquer avec un ordinateur est la liaison USB série (Universal Serial Bus). Elle est notamment utilisée pour le clavier, la souris, les imprimantes, on encore pour communiquer avec certaines cartes électroniques, lecteurs MP3, smartphones etc.



Connecteurs mâles de type A et B

### 1.1 Composition du câble

Un câble USB est composé de 2 paires de câbles :

- une paire de fils permettant l'alimentation du périphérique (GND et VBUS).
- une paire de fils (D+ et D-) destinés à faire transiter les données. Suivant le différentiel de tension entre les fils D+ et D-, le périphérique (émetteur ou récepteur) l'intercepte comme un 1 ou un 0. Le débit théorique de 0 ou de 1 est de 5 Gbits/s pour de l'USB 3.

### 1.2 Communication USB

Lors d'une transaction, les périphériques échangent des **paquets**. Un paquet est une succession de bits (0 ou 1). L'information transmise est donc numérique. C'est le périphérique hôte (celui sur lequel est branché l'USB) qui gère la transaction, par exemple un PC.

Une transaction a la forme suivante :



- Les bits de début de transaction indiquent... le début de la transaction.
- Le paquet de rapport permet de vérifier si le paquet a été correctement transmis ou indique l'état du périphérique (bloqué, interrompu temporairement...).
- Le paquet de données contient les informations utiles que l'on souhaite transmettre. Il existe des procédures de tests permettant de vérifier que les données n'ont pas été perdues ou modifiées.
- Le paquet de jetons permet de savoir (entre autre) si le PC veut recevoir ou envoyer des informations.

## 2 COMMUNICATION DANS UN RÉSEAU « INTERNET »

Les communications Internet se sont développées à partir des années 1990 et ont connu une croissance exponentielle. Le point clé pour permettre de nombreuses communications par réseau internet est d'utiliser des serveurs qui dialoguent entre eux et des clients.

### 2.1 Client – Serveur

Deux ordinateurs sont nécessaires pour créer un réseau.

#### Définition : serveur

Un serveur est un ordinateur ou un logiciel permettant de fournir des services à un ou plusieurs ordinateurs clients : serveur de fichiers, serveur de courrier électronique, serveur web...

#### Définition : client

Un client est un ordinateur ou un logiciel qui demande des services à un serveur.

#### Exemples :

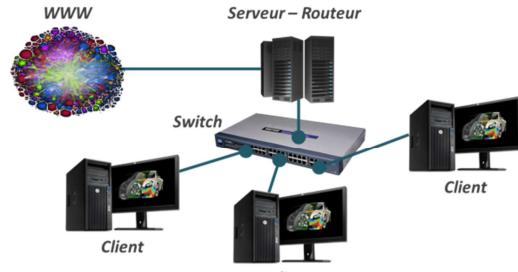
Service installé sur le serveur	Logiciel installé sur le client utilisant le service
Serveur WEB Apache	Navigateur (Chrome, Firefox, Iceweasel, Safari,...)
Serveur de diffusion streaming VLC serveur	Client lecteur de flux video : VLC Client
Serveur de mail : Postfix	Lecteur de mail : Thunderbird

Pour les utilisateurs, en général, seul le nom du logiciel « client » est connu.

## 2.2 Architecture d'un réseau local



Architecture d'un réseau domestique



Architecture d'un réseau collégien

Dans le cas d'un réseau domestique, le serveur (box) est en fait relié à un serveur du fournisseur d'accès, lui-même relié à internet. Dans ce cas, le serveur joue le rôle de **routeur**.

Dans un réseau de collège, le routeur a la particularité de relier un réseau d'ordinateurs à Internet. Il gère aussi la communication entre les ordinateurs du réseau du collège.

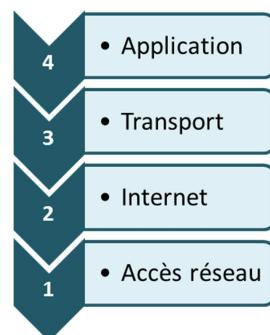
## 2.3 Communication dans un réseau

Sur un réseau, chaque ordinateur à une adresse nommée adresse IP. Cette adresse est indispensable pour que les ordinateurs communiquent entre eux. L'adresse IP est l'équivalent d'un numéro de téléphone ou d'une adresse postale. Pour atteindre son destinataire, il est souvent nécessaire de passer par plusieurs routeurs.

Pour transférer des données le client et le serveur échangent des paquets « TCP/IP ». Pour représenter ce protocole, on utilise le modèle de couches. Suivant les choix de représentation, on peut considérer 4 ou 5 couches.

Une couche fournit des services à la couche supérieure et utilise des services de la couche inférieure.

Un paquet IP (ou trame) a une taille maximale de 65535 octets. Sa structure est la suivante (chaque colonne correspond à un bit, un regroupement de 8 bits correspond à un octet, un regroupement de 4 bits correspond à un caractère hexadécimal) :



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31										
Version IP	Long. en-tête	Type de service								Longueur de la trame																															
Identification								Flags		Fragment offset																															
Durée de vie				Protocole (TCP)								Somme de contrôle de l'en-tête																													
Adresse IP source																Adresse IP Destination																									
Options + DONNEES																																									

L'ensemble de la trame est codé en binaire et transmise sur le réseau. La trame fait apparaître une adresse source et une adresse destination de la trame. Ces informations permettent le routage du paquet.

## 2.4 Routage

Une adresse IP V4 est codée sur 32 bits regroupés par paquets de 8 bits. 8 bits permettant de coder 256 informations, les adresses vont donc de 0.0.0.0 à 255.255.255.255. Ainsi, chaque ordinateur connecté à internet a donc une adresse unique (certaines adresses sont réservées).

Chaque ordinateur appartient à un sous réseau. Dans ce sous réseau, il existe un serveur particulier nommé routeur dont le but est de faire communiquer les sous réseaux les uns avec les autres. Celui-ci a donc deux interfaces : une interface vers un sous réseau « local » une interface vers le web.

Le routeur permet de faire transiter des paquets alors qu'il n'en est pas destinataire. Il se comporte comme un aiguilleur.

Suivant la configuration du serveur, lorsqu'un ordinateur se connecte sur un réseau, l'ordinateur peut se voir automatiquement attribuer une adresse IP. Ce réseau à la particularité d'avoir un masque. Ce masque permet de connaître toutes les adresses IP du réseau. Les machines d'un même sous-réseau n'ont pas besoin de routeur pour communiquer entre elles.

La connaissance d'un masque permet de connaître toutes les machines appartenant au réseau. Pour cela, on réalise un ET logique entre l'adresse IP et le masque. Cela permet d'obtenir l'identifiant du réseau.

Pour obtenir l'identifiant de l'hôte, il faut réaliser un complément à 1 du masque de sous réseau avec lequel on fait un ET logique de l'adresse IP.

#### **Exemple :**

Mon adresse IP est : 192.168.1.100. Mon masque de sous-réseau est : 255.255.255.0

On a donc :

Adresses	Décimal	Binaire
Adresse IP	192.168.1.100	$\Rightarrow$ 11000000.10101000.00000001.01100100
Adresse du masque	255.255.255.0	$\Rightarrow$ 11111111.11111111.11111111.00000000
ID Réseau	192.168.1.0	$\Leftrightarrow$ 11000000.10101000.00000001.00000000
Complément à 1 du masque		00111111.01010111.11111110.11111111
ID de l'hôte	000.000.000.100	$\Leftrightarrow$ 00000000.00000000.00000000.01100100

## **3 COMMUNICATION RÉSEAU SANS FIL**

### **3.1 Le protocole Bluetooth**

Le Bluetooth est un protocole de communication sans fil. Il a vu le jour à la fin des années 1990 et n'a vraiment percé que dans les années 2000.

La légende dit que ce terme vient d'Harald la dent bleue, célèbre Viking, qui a réussi en son temps le véritable tour de force d'unifier toutes les tribus des environs en trouvant un moyen pour qu'elles communiquent enfin entre elles au lieu de se taper dessus... En reliant les objets entre eux (ordinateur et casque sans fil, smartphone et voiture, ...) le Bluetooth établit une connexion sécurisée de proximité. Bien que cela puisse apparaître comme une des limites de ce protocole, il faut que les appareils soient proches les uns des autres ; c'est en réalité son principal argument en regard de la sécurité: l'interception du signal ne peut pas se faire à distance.

Le système Bluetooth opère dans les bandes de fréquences ISM\* (Industrial, Scientific and Medical) 2,4 GHz dont l'exploitation ne nécessite pas de licence vu la faible puissance d'émission et le risque faible d'interférences. La nouvelle version du protocole, Bluetooth LE (pour Low Energie), nécessite beaucoup moins d'énergie que le WiFi, et les développeurs travaillent sur la possibilité de faire un réseau maillé, ce qui permettrait à plusieurs composants de communiquer entre eux.

Le Bluetooth nécessite une communication bidirectionnelle, deux modules peuvent communiquer ensemble en même temps. Le comportement utilisé est de type "maître/esclave". Un esclave pourra parler avec un seul maître, mais un maître pourra dialoguer avec plusieurs esclaves.

L'utilisation se passe en plusieurs étapes :

- le maître se met en mode "reconnaissable" et fait une recherche des périphériques Bluetooth avoisinant ;
- l'esclave trouve le maître et envoie son nom pour qu'une connexion puisse être établie ;
- le maître accepte la connexion en validant par un code à 4 chiffres la connexion avec l'esclave si besoin ;
- les périphériques sont alors appairés (ou associés) ;
- la communication peut commencer.

Ensuite, selon le type de composant que vous utilisez (une oreillette bluetooth, une manette de jeu-vidéo etc) la communication pourra se faire selon un protocole ou un autre. Dans la plupart des cas cela consistera

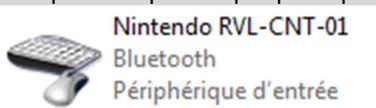
simplement en une liaison série. Au final, nous aurons donc le même fonctionnement qu'une liaison série habituelle en reliant un câble USB avec la carte Arduino mais toute la partie "filaire" de la communication sera englobée dans des trames Bluetooth gérées par le périphérique. C'est totalement transparent pour l'utilisateur.

### **Exemple de connexion d'une manette Bluetooth Wiimote avec un ordinateur (Windows) équipé d'une clé Bluetooth.**

Lorsque la clé Bluetooth (type 4.0) est connectée au PC et que les drivers sont bien installés, un icône apparaît en bas à droite de l'écran dans la barre des tâches.



Cliquer sur cet icône et choisir Ajouter un périphérique. Appuyer sur le bouton rouge de la Wiimote. Des drivers sont installés sur le PC spécifiquement pour le périphérique et le nom suivant apparaît dans la fenêtre



de recherche de périphériques . Une fois l'opération réussie le PC et la manette sont appairés. Un TP est disponible pour tester l'utilisation de la manette comme périphérique d'entrée d'un jeu vidéo simple.