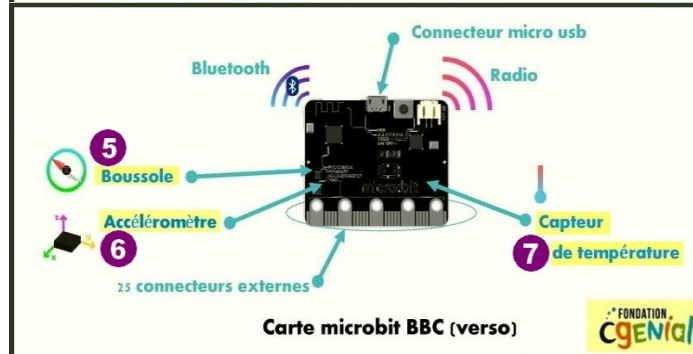
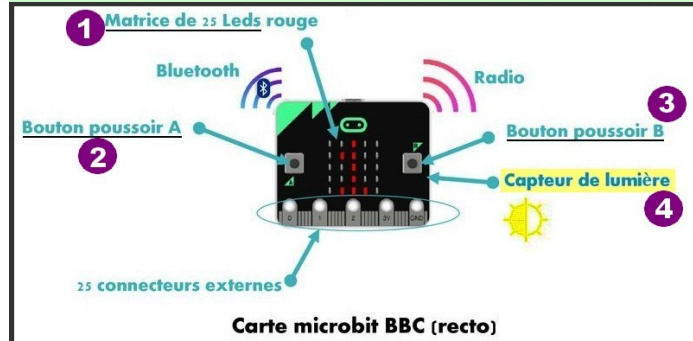
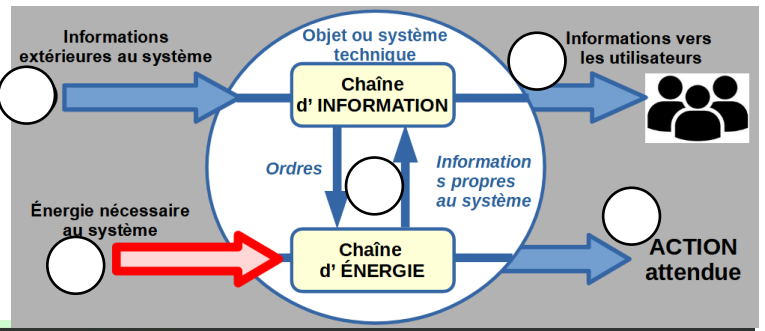


# Le Schéma fonctionnel

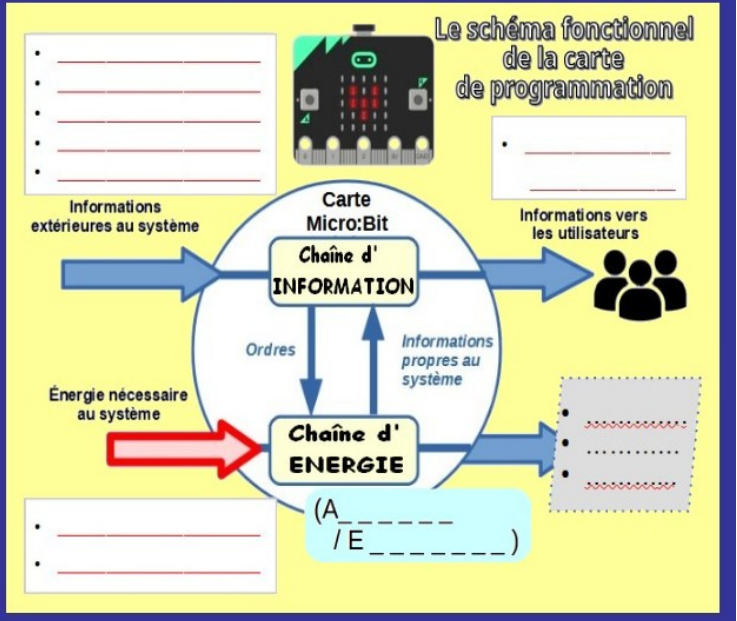
- Cette représentation permet d'identifier les flux d'énergies nécessaires aux actions et les flux des informations délivrées par les capteurs au sein du système, pour en expliquer la **STRUCTURE** et le **FONCTIONNEMENT**.

flux à compléter ⇒ (E) en Entrée / (S) en Sortie / (I) en Interne



## La constitution de la carte Micro:Bit

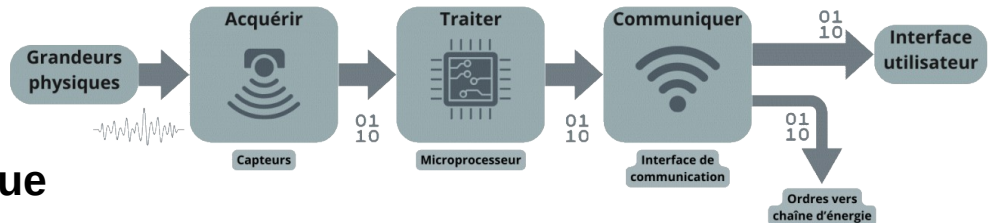
Compléter par les propositions suivantes : matrice de 25 leds / Boutons Poussoirs A & B / programme / INFORMATION / Allumer / Éteindre / Boussole / Accéléromètre / Capteur de lumière / Capteur de température



L'information peut être de deux natures :

**logique ou analogique**

Très souvent, les capteurs convertissent des **grandeurs physiques** (comme la lumière, la température ou un simple contact) en **signaux numériques**. Ensuite le microcontrôleur traite ces signaux numériques (code binaire en langage informatique).



Seq.3



**Ce Que je Dois Retenir !!!**

n° 1

**sur la Chaîne d'information**

- Un **système automatisé** a besoin d'**informations** pour fonctionner de manière **autonome**.
- La **chaîne d'information** comprend trois fonctions principales : **acquérir**, **traiter**, et **communiquer**.
- Le **flux d'information** représente la **circulation de l'information** à travers l'objet technique, qu'elle soit **logique** ou **analogique**.
- D'abord, les **grandeurs physiques** sont acquises par des **capteurs**.
- Ensuite, les signaux générés par ces capteurs (souvent analogiques) sont convertis puis transmis au **microcontrôleur** qui traite les informations.
- Enfin, les informations traitées donnent des ordres à la **chaîne d'énergie** et sont communiquées vers une **interface utilisateur** (écran, voyant, haut-parleur...) ou vers **un autre système** (ordinateur, smartphone...).



n°1

# La chaîne d'information de la carte micro:Bit

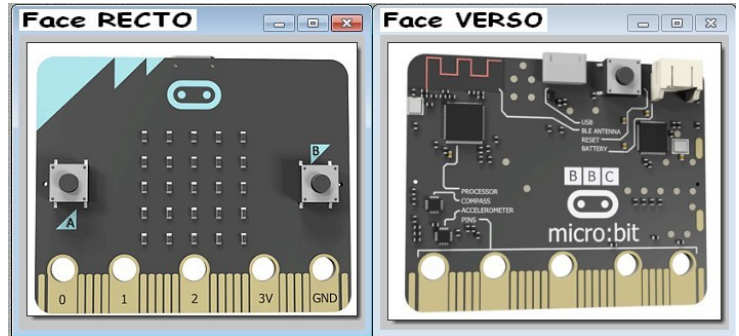
Seq.3 /-Comment s'organise le fonctionnement d'une carte de programmation ?

Entre flux d'informations et flux d'énergie...

Travail à effectuer : Compléter le document ci-dessous après avoir parcouru la description de la carte de programmation Micro:Bit ?

Détail en lien à parcourir ==>

<http://www.technobm.clg-gdm.fr/TEchno2016/C4-Cinquieme/Seq-ROBOTIQUE/carteMicroBit.jpg>

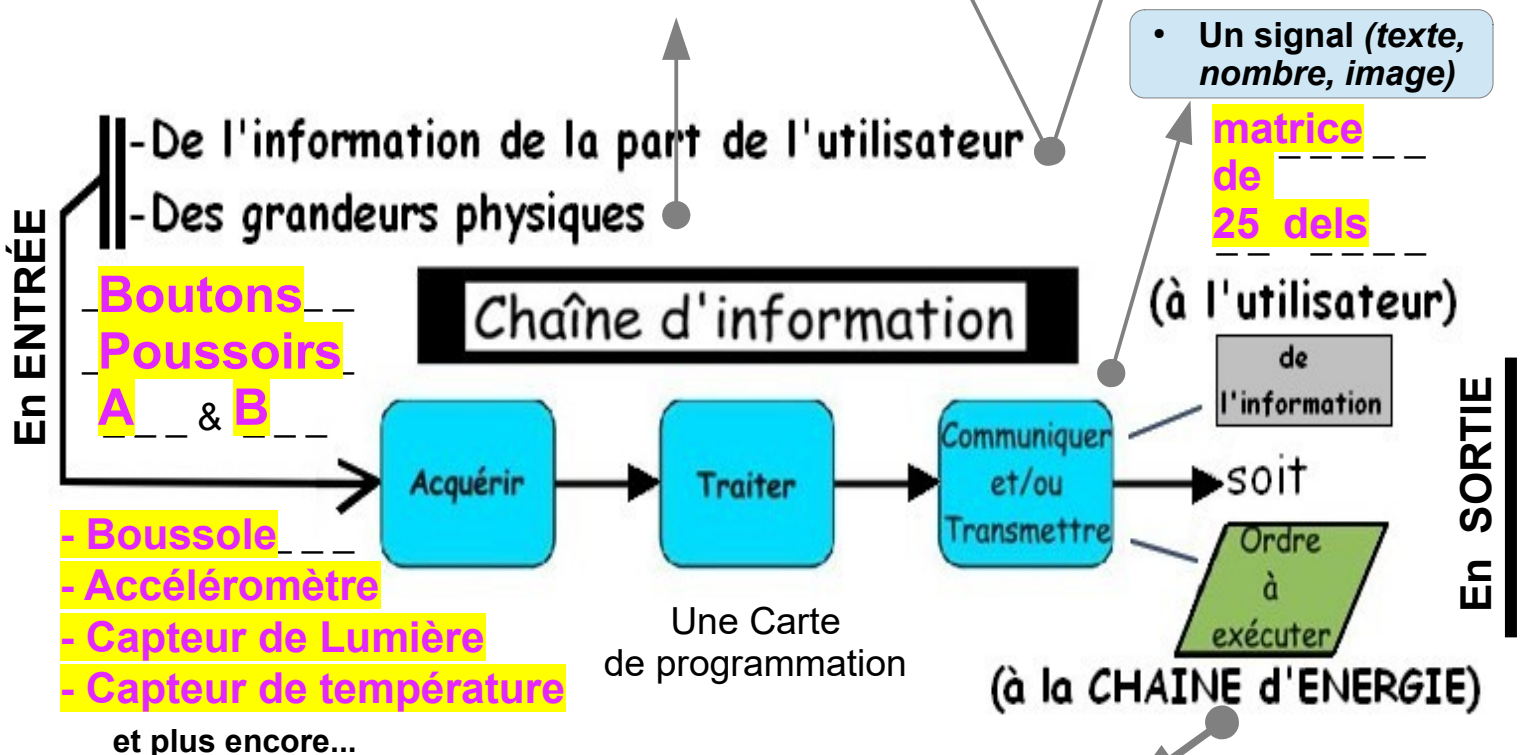


Programmer, c'est d'abord traiter de l' **INFORMATION** ...

-Sans rentrer dans les connaissances de l'électronique, décrire le fonctionnement d'un système peut s'effectuer en faisant **une représentation sous schéma**.

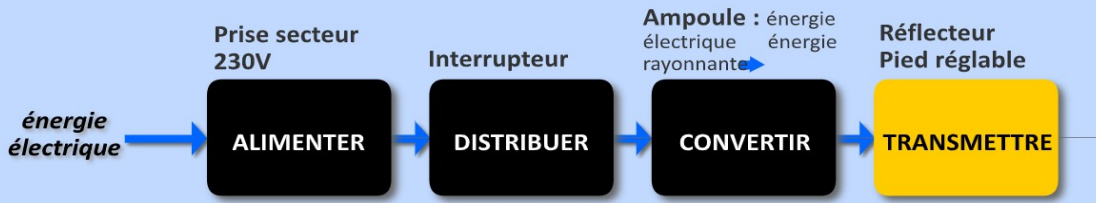
-**La Chaîne d'information** permet de décomposer le fonctionnement d'un objet technique sur **les flux d'informations internes et externes** au système.

- L'information utilisateur pour la carte Micro:Bit est l'action des **Boutons Poussoirs A & B** et la transmission du **programme** dans la carte Micro:Bit.
- Les grandeurs physiques sont diverses et liées aux capteurs internes de la carte **Boussole / Accéléromètre / Capteur de Lumière / Capteur de température**

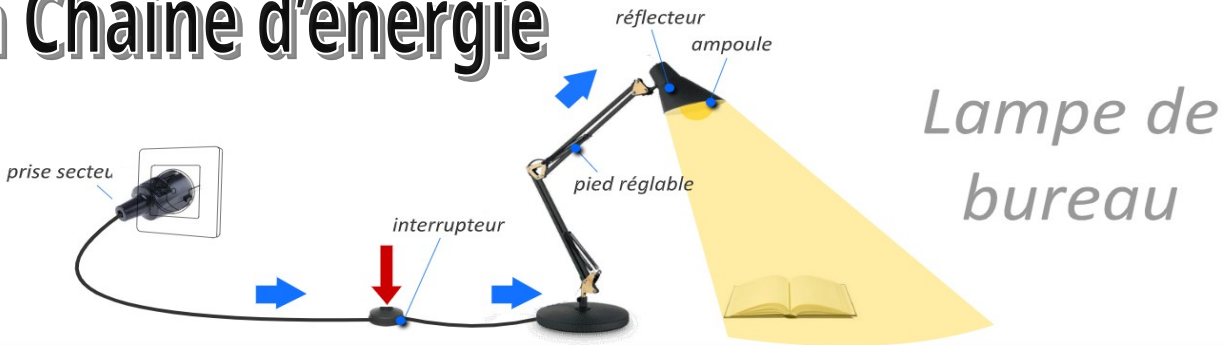


- L'enchaînement de l'animation sur la matrice de dels avec les actions d' **Allumer** et **Éteindre** les Dels.

Compléter par les propositions suivantes : matrice de 25 dels / Boutons Poussoirs A & B / programme / INFORMATION / Allumer / Éteindre / Boussole / Accéléromètre / Capteur de lumière / Capteur de température



# La Chaîne d'énergie



## Prenons l'exemple d'une voiture électrique.

⇒ Comment l'énergie circule dans ce système ?

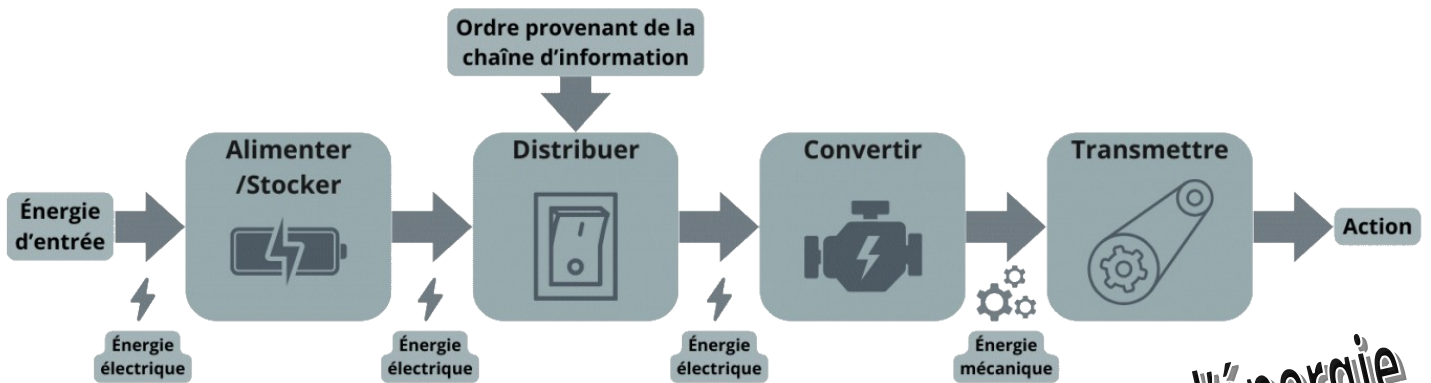
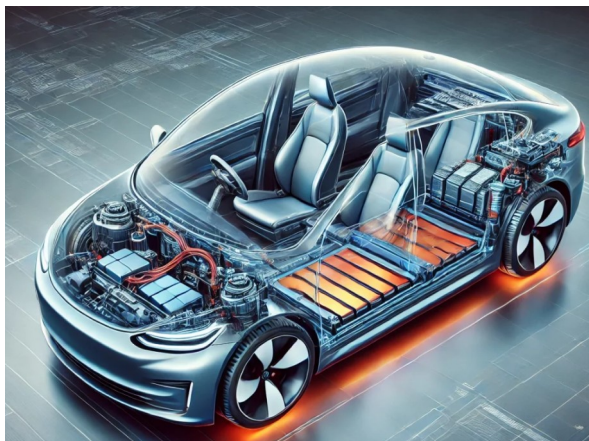
**Alimenter** : Une borne de recharge alimente la voiture.

**Stocker** : La batterie située sous les sièges de la voiture stocke l'énergie.

**Distribuer** : L'énergie stockée est ensuite distribuée à travers des câbles vers différents composants, dont le moteur.

**Convertir** : Le moteur électrique convertit cette énergie en énergie mécanique.

**Transmettre** : Enfin les mécanismes transmettent l'énergie mécanique aux roues pour faire avancer la voiture.



Seq.3



**Ce Que je Dois Retenir !!!**

n° 2

sur la Chaîne d'énergie

- Un **système** ou **objet technique** a besoin **d'énergie** pour fonctionner.
- Certains systèmes comme l'électroménager, n'ont pas besoin de la fonction de **stockage** d'énergie.
- La **chaîne d'énergie** comprend les étapes suivantes : **alimentation**, **stockage** (si nécessaire), **distribution**, **conversion**, et **transmission**.
- Le **flux d'énergie** décrit la **circulation de l'énergie** à travers ces différentes étapes.
- La compréhension de la chaîne d'énergie est essentielle pour **diagnostiquer** et **optimiser** le fonctionnement des systèmes techniques.

(Source PLAYHOCKY & TechnoFLASH)

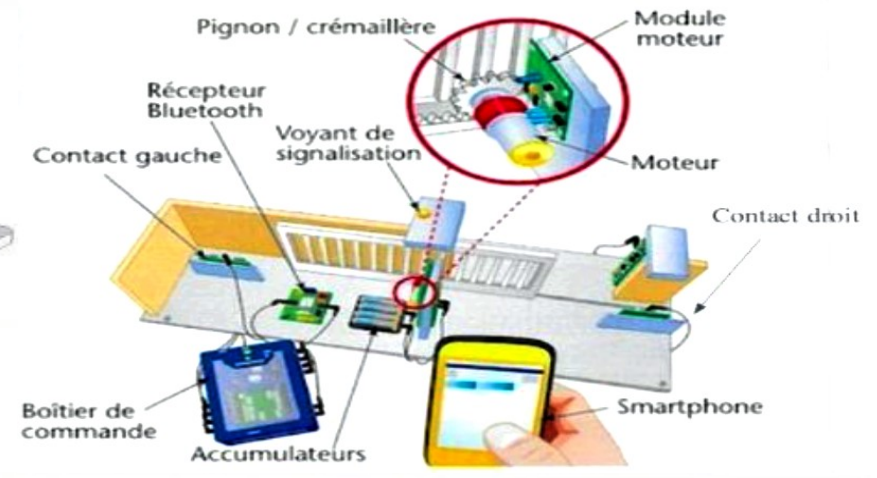
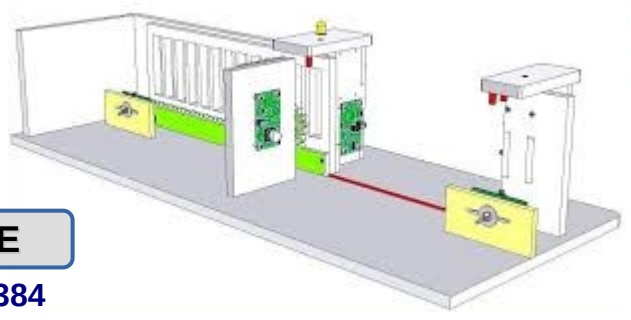
# Les chaînes fonctionnelles

## PORTAIL AUTOMATIQUE

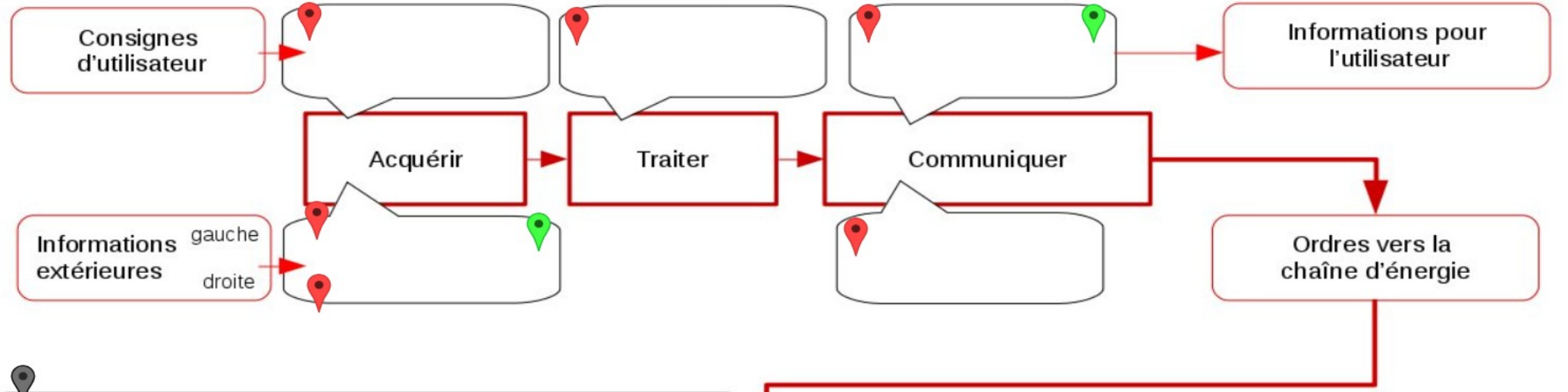
Lien ⇒ <https://learningapps.org/8082384>

### Travail à effectuer :

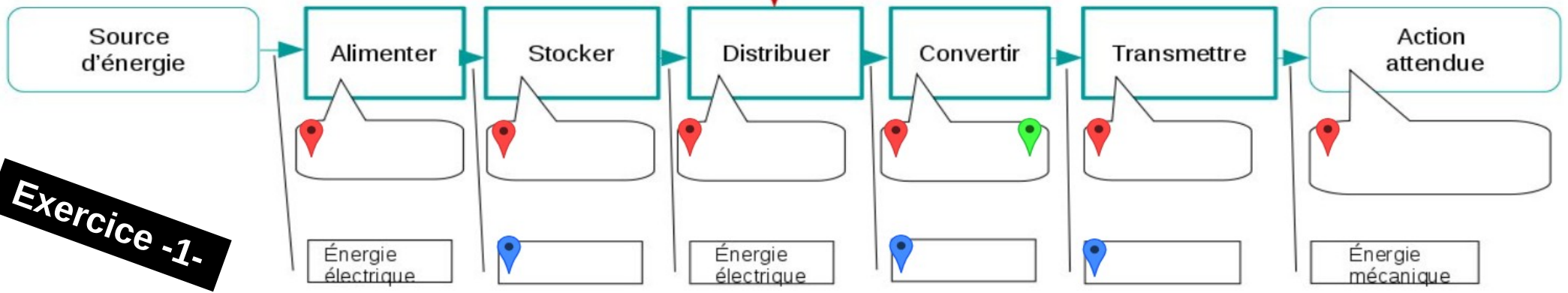
⇒ Suivre l'animation et en recopier le contenu



Empty box for notes.

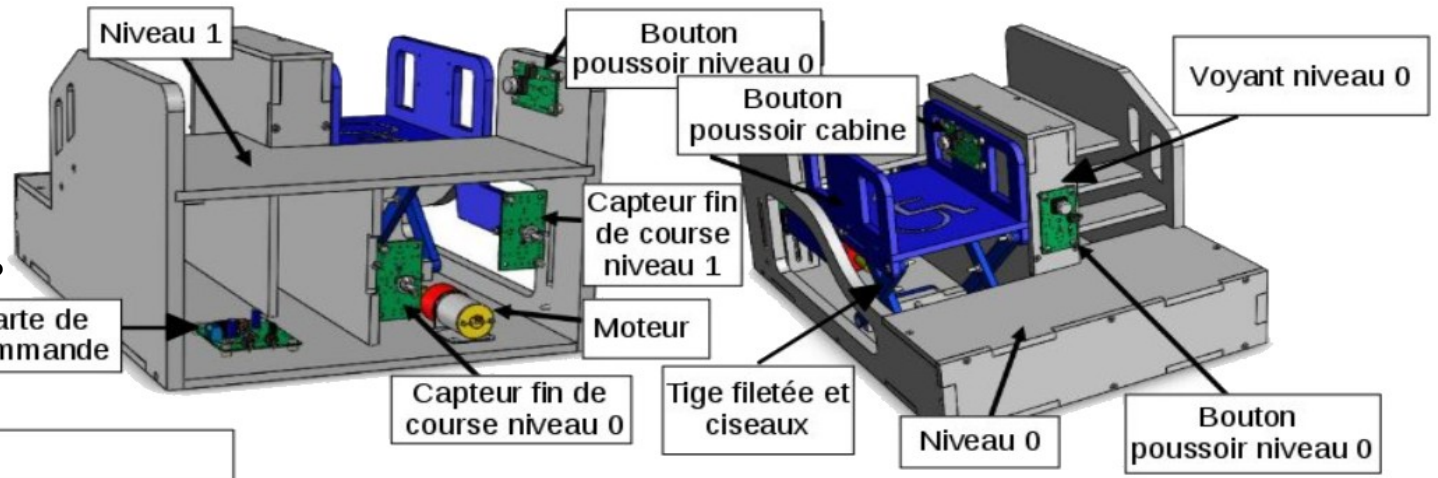


Empty box for notes.



**Exercice -1-**

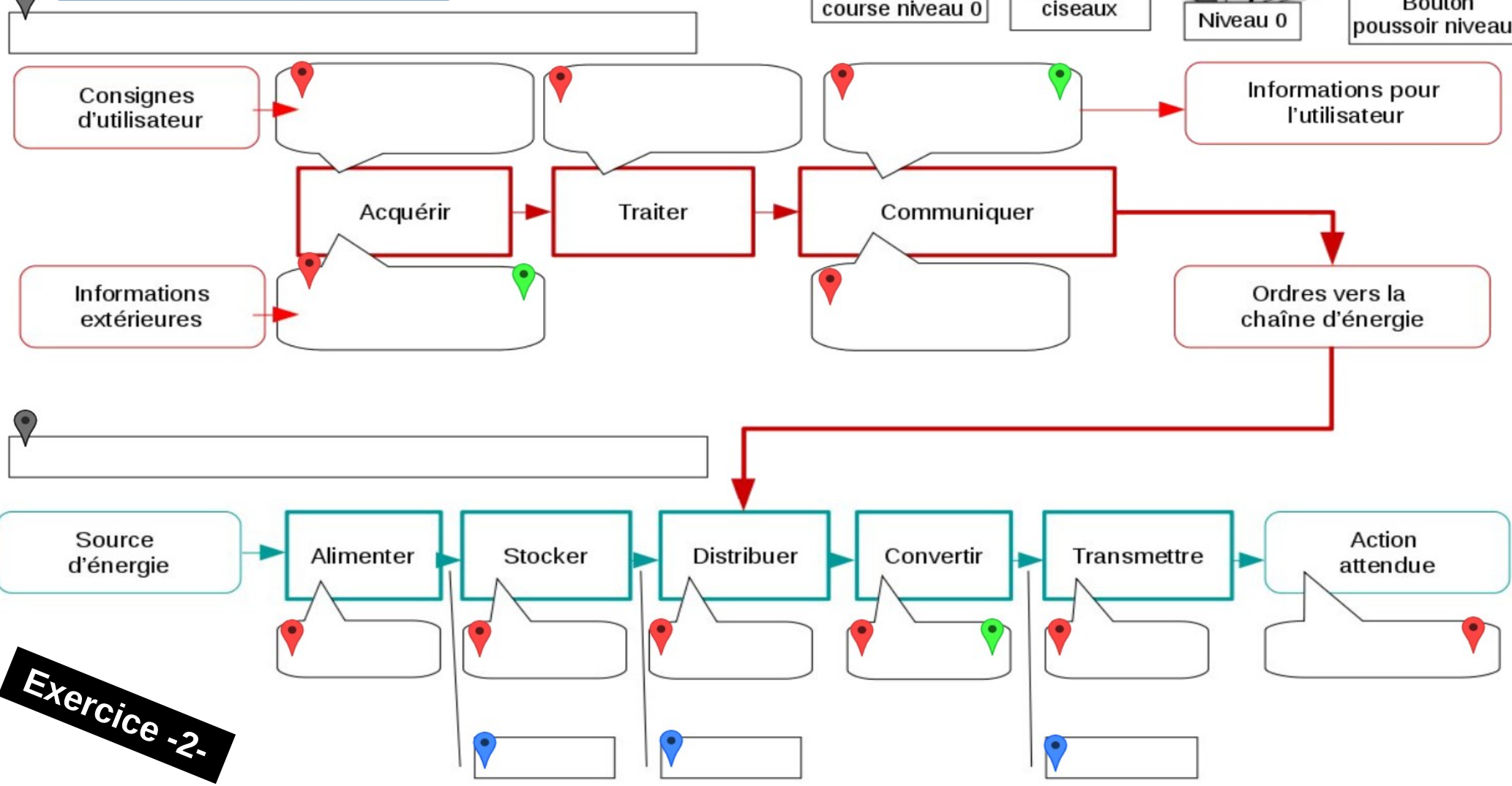
# Les chaînes fonctionnelles



**Travail à effectuer :**  
 ⇒ Suivre l'animation et en recopier le contenu ?

Lien ⇒ <https://learningapps.org/8083484>

## PLATEFORME ELEVATRICE



**Exercice -2-**

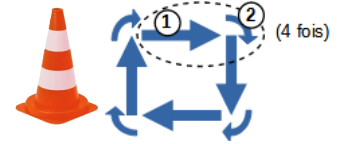
**Doc TP suivi...**

*(hors classeur)*

# TP1 /Parcours CARRE du robot Maqueen

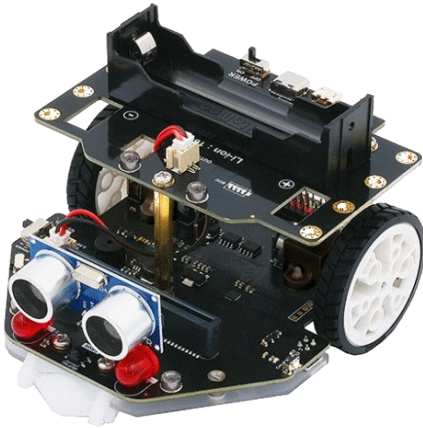
## -Les déplacements du robot

Pour contourner un plot



### ACTIVITÉ-1-

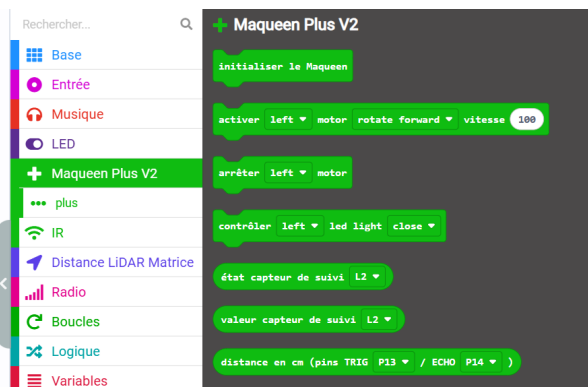
Avancer-S'arrêter-Reculer-Tourner à DROITE-Tourner à GAUCHE-Pivoter sur place



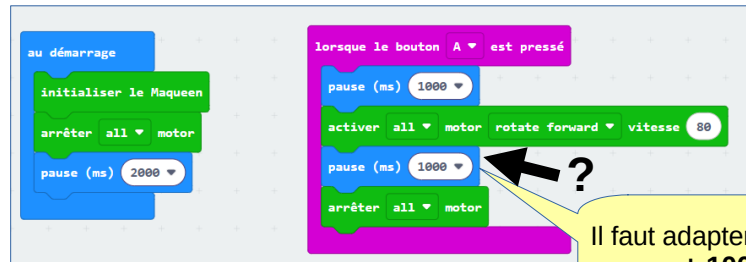
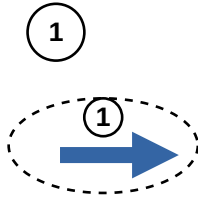
Nom de fichier :  
microbit-MaqueenPLUS-V2\_de\_DEPART

#### 1.1 -Prg initial de gestion des deux moteurs :

(attention à vérifier que les deux moteurs soient réglés pour AVANCER en LIGNE DROITE)

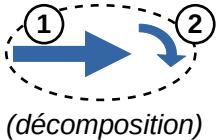


#### 11-AVANCER.hex



#### 1.2 -A vous d'adapter le programme pour chacune des fonctions ?

(121-Arrêter / 122-Reculer / 123-Tourner à DROITE / 123-Tourner à GAUCHE / 124-Pivoter sur place)

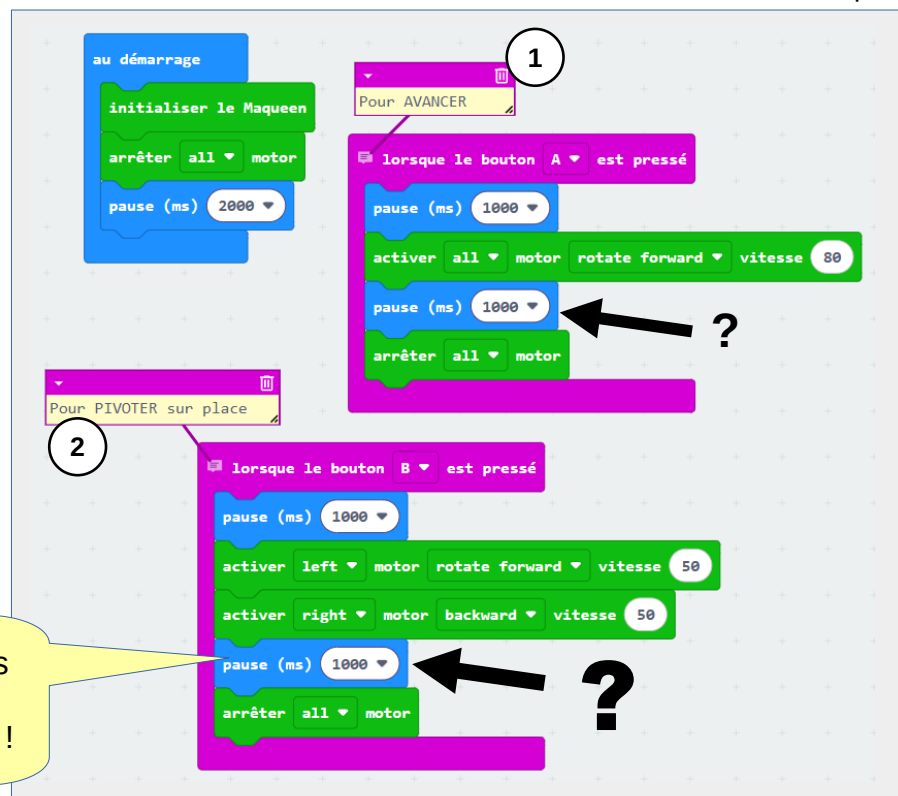


Nom de fichier :  
12-AVANCER-PIVOTER.hex



(commande par  
Bouton A & Bouton B)

Il faut adapter le temps du mouvement **± 1000 !**



### 1.3 -Adaptation du programme pour tracer un carré :

#### modèle n°1

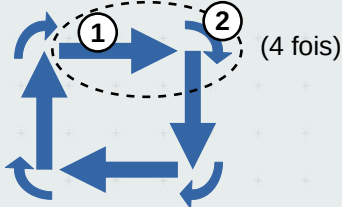
Nom de fichier :  
131-CARRE.hex

The Scratch code for 'modèle n°1' is as follows:

- au démarrage**
  - initialiser le Maqueen
  - arrêter all motor
  - pause (ms) 2000
- lorsque le bouton A est pressé**
  - répéter 4 fois**
    - faire**
      - pause (ms) 1000 (labeled 'Pour AVANCER')
      - activer all motor rotate forward vitesse 80 (labeled '1')
      - pause (ms) 1000 (with a question mark)
      - arrêter all motor
      - pause (ms) 1000 (labeled 'Pour PIVOTER sur place')
      - activer left motor rotate forward vitesse 50 (labeled '2')
      - activer right motor backward vitesse 50
      - pause (ms) 1000 (with a question mark)
      - arrêter all motor

Annotations: A diagram on the left shows a square path with arrows and labels '1' and '2'. A blue box on the left contains the text 'Adaptation pour l'obtention d'un CARRE'.

Adaptation  
pour l'obtention  
d'un CARRE



#### modèle n°2 (avec sous-programmes)

Nom de fichier :  
132-CARRE-SousPrg.hex

The Scratch code for 'modèle n°2' is as follows:

- au démarrage**
  - initialiser le Maqueen
  - arrêter all motor
  - pause (ms) 2000
  - répéter 4 fois**
    - faire**
      - pause (ms) 1000
      - appel AVANCER
      - pause (ms) 1000
      - appel PIVOTER

**fonction AVANCER**

- activer all motor rotate forward vitesse 80
- pause (ms) 1000 (with a question mark)
- arrêter all motor

**fonction PIVOTER**

- activer left motor rotate forward vitesse 50
- activer right motor backward vitesse 50
- pause (ms) 1000 (with a question mark)
- arrêter all motor

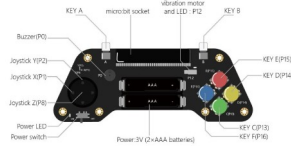
Annotations: A 'Fonctions' panel on the right shows 'Créer une fonction...', 'retourner 0', and 'Vos fonctions' with 'appel AVANCER' and 'appel PIVOTER'. Arrows labeled '1' and '2' point to the function creation and the function list respectively.



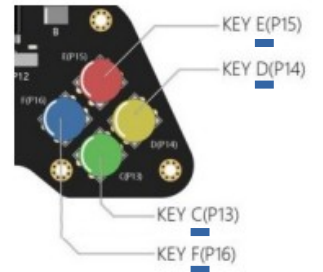
## ACTIVITÉ-2-

### Avancer-S'arrêter-Reculer-Tourner à DROITE-Tourner à GAUCHE

par télécommande



# TELECOMMANDE /modèle 2



### Ressource à utiliser : La COMMUNICATION RADIO

- Utilisation du principe de communication sur un canal Radio commun entre deux cartes Micro:Bit ...  
(prenez comme numéro de groupe le numéro d'îlot)

carte n°1 Poste 1

carte n°2 Poste 2

Solution /Accès au lien pour l'écran double EDITEUR  
=> <https://makecode.com/multi-editor>

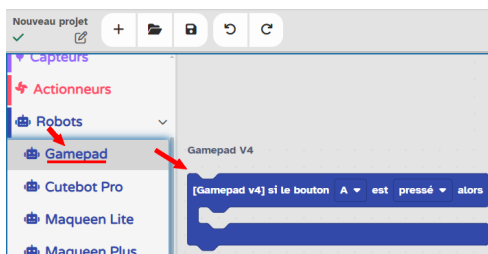
### Tâche -2- complexe à effectuer : La COMMUNICATION RADIO

- Réutiliser le principe de programmation ci-dessus pour rendre télécommandé le parcours carré autour d'un obstacle.  
(prenez comme numéro de groupe le numéro d'îlot)

Direct...

Maqueen v4

## Pour la TELECOMMANDE /Carte 1



Initialisation du Canal de COMMUNICATION  
(choisir le numéro d'îlot et du poste /exemple 11 pour l'îlot-poste1)

Au démarrage

[Radio] configurer Canal 0 Puissance 6 Taille des données 32 Groupe 0

Blocs sur carte Micro:Bit 1 TELECOMMANDE

Répéter indéfiniment

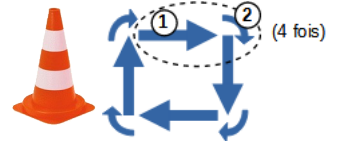
[Gamepad v4] si le bouton E est pressé alors

[Radio] envoyer le nombre ou la liste 1

attendre 100 millisecondes

# -Les déplacements du robot

Pour contourner un plot



## Ressource à utiliser : La COMMUNICATION RADIO

- Utilisation du principe de communication sur un canal Radio commun entre deux cartes Micro:Bit ...  
(prenez comme numéro de groupe le numéro d'îlot)

The left editor (Poste 1) contains the following code:

```

    toujours
    radio définir groupe 1

    lorsque le bouton A est pressé
    envoyer le nombre 1 par radio

    lorsque le bouton B est pressé
    envoyer le nombre 2 par radio
    
```

The right editor (Poste 2) contains the following code:

```

    toujours
    radio définir groupe 1

    quand une donnée est reçue par radio receivedNumber
    si receivedNumber = 1 alors
    afficher texte "A"
    si receivedNumber = 2 alors
    afficher texte "B"
    
```

Solution /Accès au lien pour l'écran double EDITEUR  
=> <https://makecode.com/multi-editor>

Direct...

Maqueen v4



<https://fr.vittascience.com/microbit/>



## Pour le Robot MAQUEEN /Carte 2

Blocs sur carte Micro:Bit 2  
Robot MAQUEEN Plus V2

**Répéter indéfiniment**

```

[Radio] si un nombre est reçu dans numberData alors
si numberData is 1 alors
AVANCER
si numberData is 2 alors
TOURNER
    
```

Initialisation du Canal de COMMUNICATION  
(choisir le numéro d'îlot et du poste /exemple 11 pour îlot1-poste1)

**Au démarrage**

[Radio] configurer Canal 8 Puissance 6 Taille des données 32 Groupe 0

**définir AVANCER**

```

[Maqueen Plus v2] avancer à La vitesse 80
attendre 1 seconde.s
[Maqueen Plus v2] arrêter le moteur droit & gauche
    
```

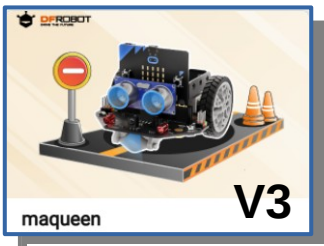
**définir TOURNER**

```

[Maqueen Plus v2] pivoter vitesse 30
attendre 1 seconde.s
[Maqueen Plus v2] arrêter le moteur droit & gauche
    
```

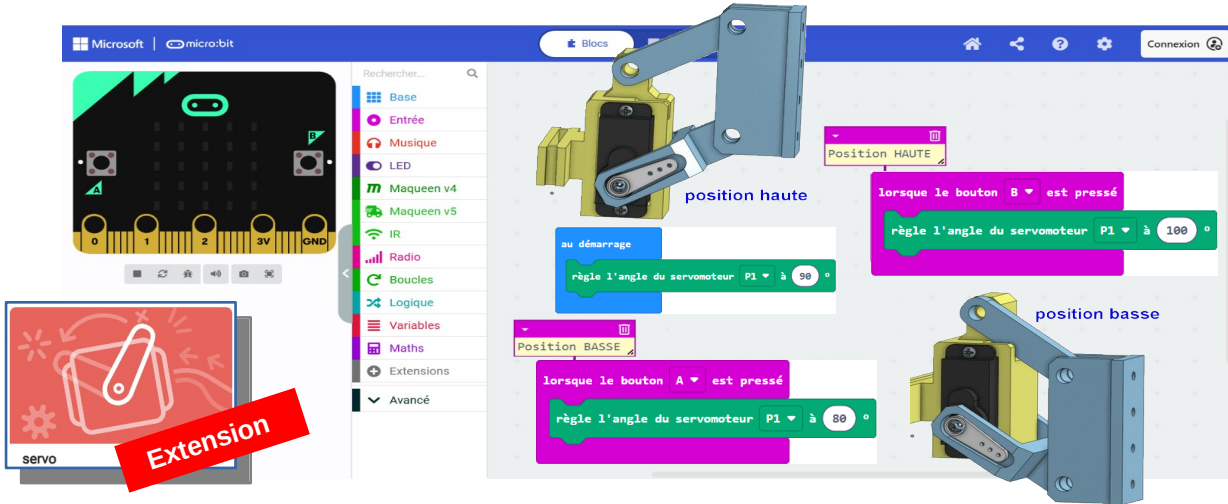
**Documents non abordés**  
(ne pas apprendre)

**à venir**  
**pour 2026-2027**



# TP2 / Pilotage des fourches du chariot élévateur

Remarque : sur Modèle Maqueen classique V3 / **Extension** SERVO  (P1 ou P2) obligatoire

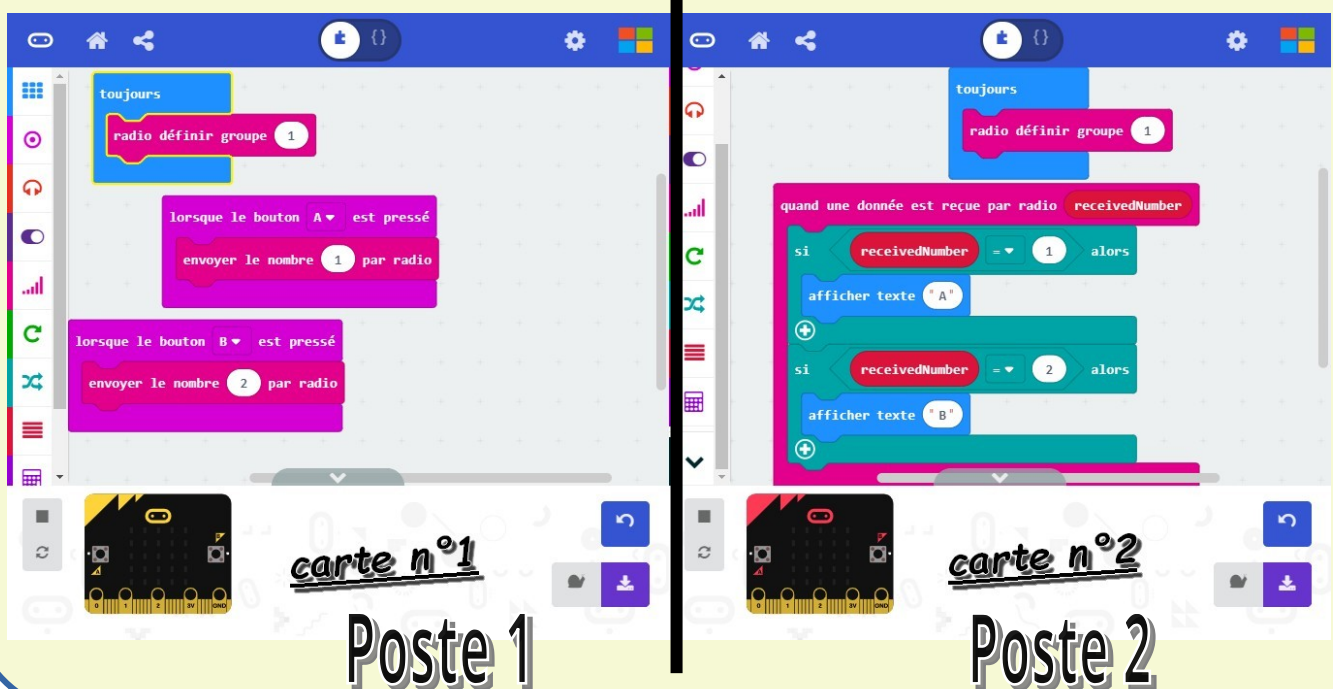


## Travail -1- à effectuer :

- Reproduire le programme de la figure ci-dessus ?
- Régler l'angle de la position BASSE ?
- Régler l'angle de la position HAUTE ?

## Tâche -2- complexe à effectuer : La COMMUNICATION RADIO

- Réutiliser le principe de programmation ci-dessous pour rendre télécommandé l'actionneur des fourches du chariot élévateur. *(prenez comme numéro de groupe le numéro d'îlot)*





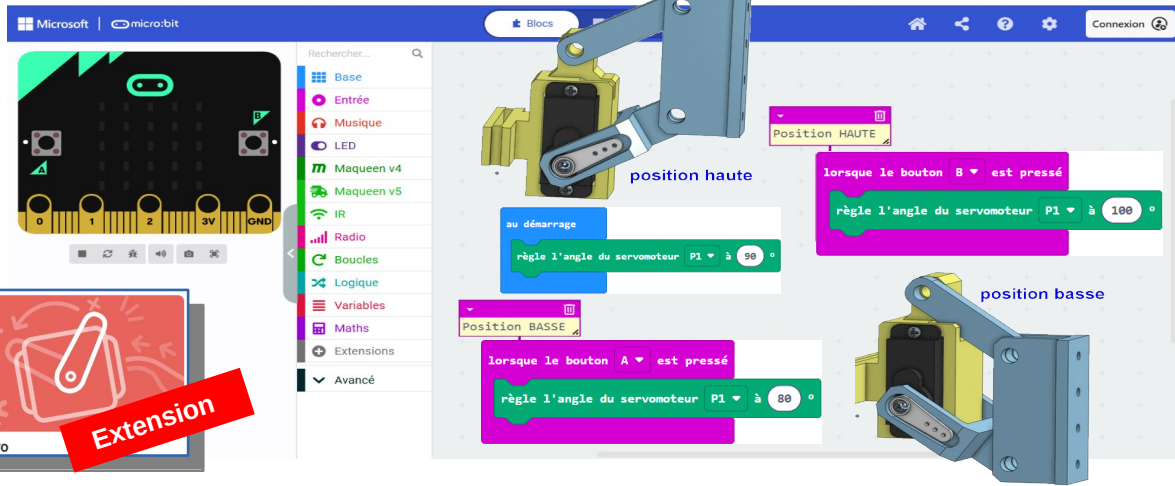
# TP2 / Pilotage des fourches du chariot élévateur

sur Modèle Maqueen Classique V4 soit direct

Maqueen v4

soit par / Extension

Servos

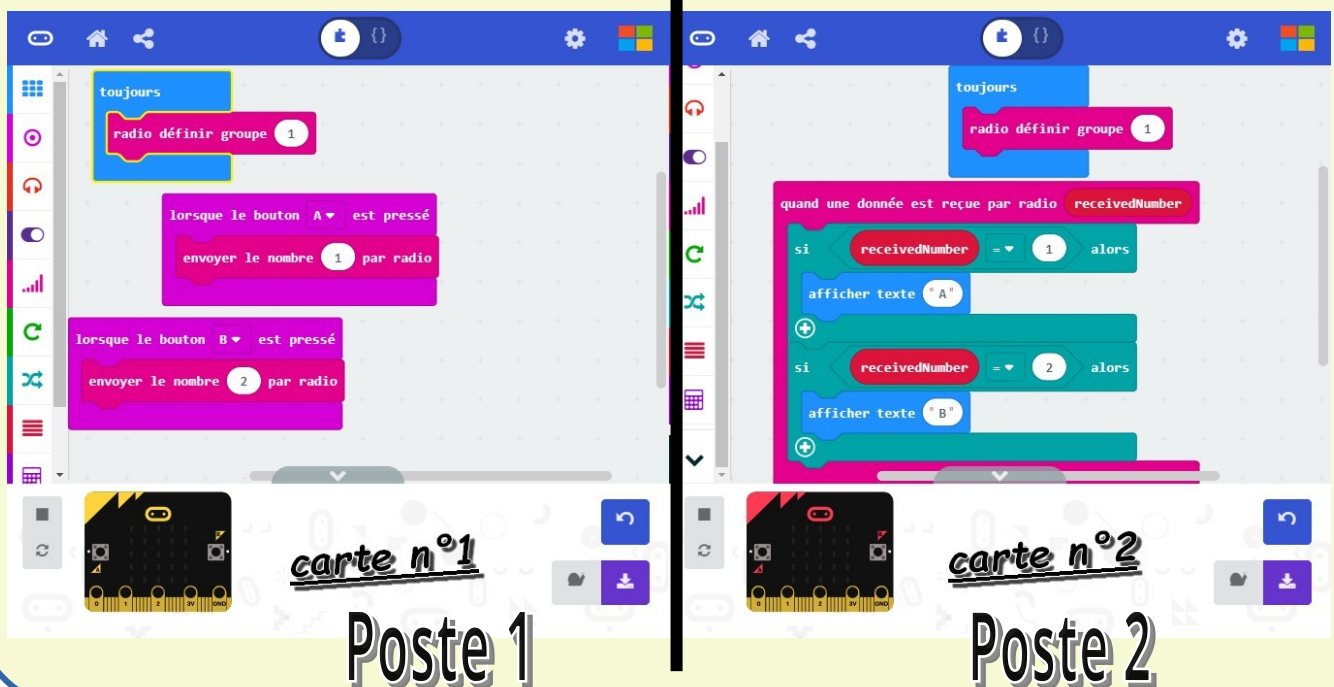


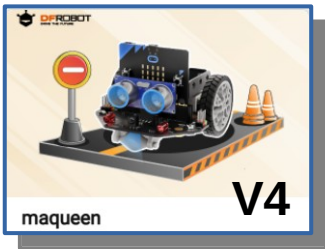
## Travail -1- à effectuer :

- Reproduire le programme de la figure ci-dessus ?
- Régler l'angle de la position BASSE ?
- Régler l'angle de la position HAUTE ?

## Tâche -2- complexe à effectuer : La COMMUNICATION RADIO

- Réutiliser le principe de programmation ci-dessous pour rendre télécommandé l'actionneur des fourches du chariot élévateur. *(prenez comme numéro de groupe le numéro d'îlot)*





# TP2 / Pilotage des fourches du chariot élévateur

sur Modèle Maqueen Classique V4 soit direct

Maqueen v4

soit par / Extension

Servos

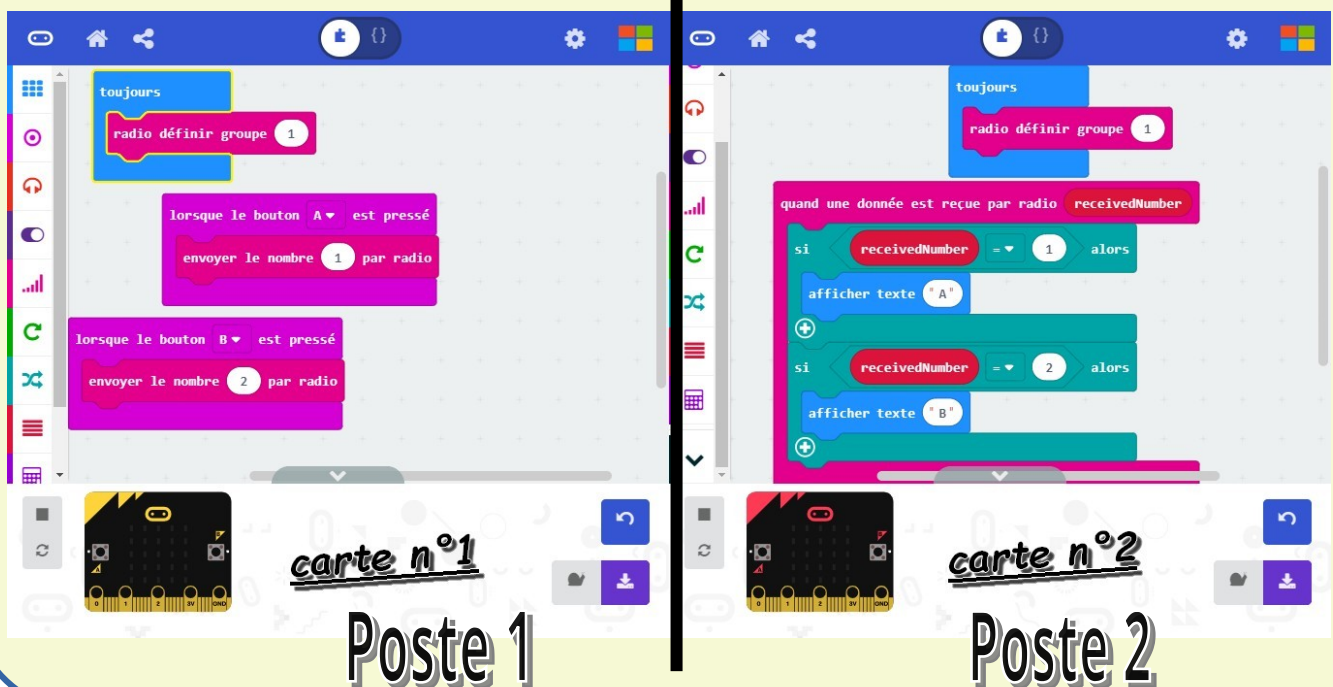


## Travail -1- à effectuer :

- Reproduire le programme de la figure ci-dessus ?
- Régler l'angle de la position BASSE ?
- Régler l'angle de la position HAUTE ?

## Tâche -2- complexe à effectuer : La COMMUNICATION RADIO

- Réutiliser le principe de programmation ci-dessous pour rendre télécommandé l'actionneur des fourches du chariot élévateur. *(prenez comme numéro de groupe le numéro d'îlot)*



Programmer, c'est ordonner une succession de conditions et actions

# Ce Que Je Dois Retenir...

Si il y a ça ...

Alors faire ceci  
Sinon faire cela

Tant qu'il y a cela

Faire ...



• Toute une question de langage :

Quoi ?	Algorithme	Organigramme ou Bloc	Code
Comment ? (QUEL outil ?)	Langage texte	Application informatique	programme
Pour Qui ?	Utilisateur	Ordinateur	Système
Pour Quoi faire ?	Décrire	Programmer	Fonctionner

• Une démarche : Algorithme - Organigramme ou Bloc - Code



**ALGORITHME**

- La lumière doit s'allumer.
- Après 0,5 seconde, la lumière doit s'éteindre.
- Après 0,5 seconde, la lumière doit se rallumer pour 0,5 seconde encore.

et ainsi de suite...

**ORGANIGRAMME** OU **BLOCS-programme**

**CHRONOGRAMME**

LED éteinte    LED allumée

**CODE de programmation**

```

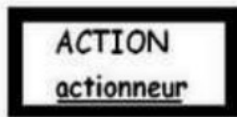
1 'BASiC converted from Ixile:
2 '
3 'Converted 2016-01-21 at 06:41:34
4
5 main:
6 do
7   if pin8.0 = 1 then
8     high C.1
9     pause 500
10    low C.1
11   else
12     low C.1
13   endif
14 loop
15 stop
        
```

## Les principes d'un LOGIGRAMME ou ORGANIGRAMME

-L'organigramme obéit à des règles d'écriture très simples :  
Il débute toujours par une case début et il n'y a que trois types de cases.



Un ovale qui correspond au Début ou Fin (si fin il y a) de l'organigramme.



Correspond à une action à effectuer.



Correspond à une question à laquelle on peut répondre uniquement par oui ou par non.

## succession d' ACTIONS et de CONDITIONS

Les Capteurs par les signaux qu'ils délivrent, fixent les CONDITIONS de déroulement du programme

## 1) STRUCTURATION sur les algorithmes

**1 Variable**

C'est un espace de **mémoire** pour **stocker** une **valeur** qui peut changer au cours du programme. Elle peut être de trois types : **Texte**, **Nombre** ou **Booléen** (Vrai ou faux, True/False en anglais). Une **instruction d'affectation** associe une valeur à une variable.

**2 Instruction itérative**

Permet de **répéter** un bloc d'instructions un certain nombre de fois ou indéfiniment. Exemples : **Tant que** (while) : répète tant qu'une condition est vraie. **Pour** (for) : répète pour un nombre déterminé de fois.

**3 Instruction conditionnelle**

Permet de tester si une ou plusieurs conditions sont **vraies** ou **fausses**, et de faire des choix en conséquence (**si**, **sinon**) (If, Else en anglais).

**4 Opérateurs**

**Opérateurs logiques** : utilisés pour combiner des expressions logiques (ET, OU) (AND, OR en anglais).  
**Opérateurs arithmétiques** : permettent de faire des opérations mathématiques (addition, soustraction, etc.).  
**Opérateurs de comparaison** : utilisés pour comparer deux valeurs (=, >, <, etc.).

**5 Listes**

Une liste c'est une **structure de données** qui permet de **stocker** plusieurs valeurs de **même type** (liste de noms, liste de notes, etc.).  
Exemple : [ voiture, vélo, avion, bateau, train ]

**6 Blocs d'instructions**

Ce sont des **groupes d'instructions** qui sont **exécutés ensemble**, souvent associés à des structures conditionnelles ou itératives.

**7 Événement**

Un événement est une **action** qui se **produit** pendant l'**exécution** du programme, comme un clic de souris ou une pression sur une touche.

**8 Déclenchement d'une séquence par un événement**

Quand un **événement se produit**, une partie du programme (une **séquence d'instructions**) peut être **exécutée**. Exemple : dans un jeu, déclencher le saut d'un personnage lorsqu'on appuie sur la touche « espace ».

**Exemple d'algorithme :** Une porte automatique.  
Le jour, en présence d'une personne, la porte s'ouvre. Au bout de 5 secondes, la porte se referme. Dans le noir, la porte ne s'ouvre pas.

```

Début
  Variable luminosité = Lire_Capteur_Luminosite()
  Variable présence = Lire_Capteur_Presence()
  Répéter indéfiniment
    Si présence = vrai ET luminosité > 500
      Alors
        ouvrir porte
        attendre 5 secondes
        fermer porte
      Fin Si
    Fin Répéter
  Fin
        
```

Texte-Texte-Texte-Texte-Texte-.....

### 1) STRUCTURATION sur les algorithmes

**1 Variable**  
C'est un espace de mémoire pour stocker une valeur qui peut changer au cours du programme. Elle peut être de trois types : Texte, Nombre ou Booléen (Vrai ou faux, True/False en anglais). Une instruction d'affectation associe une valeur à une variable.

**2 Instruction itérative**  
Permet de répéter un bloc d'instructions un certain nombre de fois ou indéfiniment. Exemples : Tant que (while) : répète tant qu'une condition est vraie. Pour (for) : répète pour un nombre déterminé de fois.

**3 Instruction conditionnelle**  
Permet de tester si une ou plusieurs conditions sont vraies ou fausses, et de faire des choix en conséquence (si, sinon) (if, Else en anglais).

**4 Opérateurs**  
Opérateurs logiques : utilisés pour combiner des expressions logiques (ET, OU) (AND, OR en anglais).  
Opérateurs arithmétiques : permettent de faire des opérations mathématiques (addition, soustraction, etc.).  
Opérateurs de comparaison : utilisés pour comparer deux valeurs (=, >, <, etc.).

**Exemple d'algorithme** : Une porte automatique.  
Le jour, en présence d'une personne, la porte s'ouvre. Au bout de 5 secondes, la porte se ferme. Dans le noir, la porte ne s'ouvre pas.

```

Début
Variable luminosité = Lire_Capteur_Luminosite()
Variable presence = Lire_Capteur_Presence()
Répéter indéfiniment
  Si
    presence = vrai ET luminosité > 500
  Alors
    ouvrir porte
    attendre 5 secondes
    fermer porte
  Fin Si
Fin Répéter
Fin
        
```

**5 Listes**  
Une liste c'est une structure de données qui permet de stocker plusieurs valeurs de même type (liste de noms, liste de notes, etc.).  
Exemple : [ voiture, vélo, avion, bateau, train ]

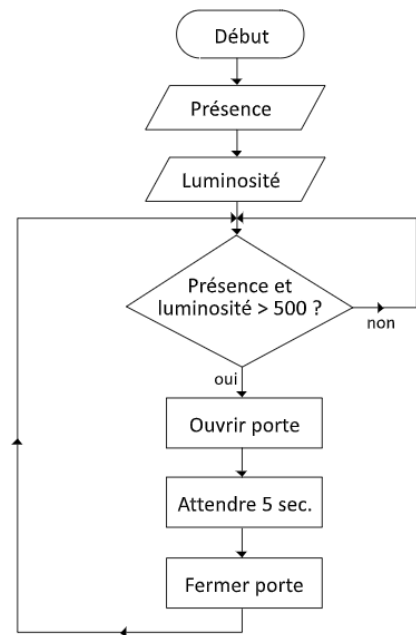
**6 Blocs d'instructions**  
Ce sont des groupes d'instructions qui sont exécutés ensemble, souvent associés à des structures conditionnelles ou itératives.

**7 Événement**  
Un événement est une action qui se produit pendant l'exécution du programme, comme un clic de souris ou une pression sur une touche.

**8 Déclenchement d'une séquence par un événement**  
Quand un événement se produit, une partie du programme (une séquence d'instructions) peut être exécutée. Exemple : dans un jeu, déclencher le saut d'un personnage lorsqu'on appuie sur la touche « espace ».

### 2) Adaptation aux ALGORIGRAMMES

- 1) Variable •
- 2) Instruction itérative •
- 3) Instruction conditionnelle •
- 4) Opérateurs •
- 5) Listes •
- 6) Blocs d'instructions •
- 7) Événement •
- 8) Déclenchement de séquence •



### 3) Adaptation aux BLOCS

- 1) Variable •
- 2) Instruction itérative •
- 3) Instruction conditionnelle •
- 4) Opérateurs •
- 5) Listes •
- 6) Blocs d'instructions •
- 7) Événement •
- 8) Déclenchement de séquence •

lorsque vous cliquez sur [icône]

luminosité à valeur mesurée par le capteur de lumière

présence à valeur mesurée par le capteur de présence

pour toujours

si présence = vrai et luminosité > 500 alors

ouvrir porte

attendre 5 sec

fermer porte