



# WAJUG

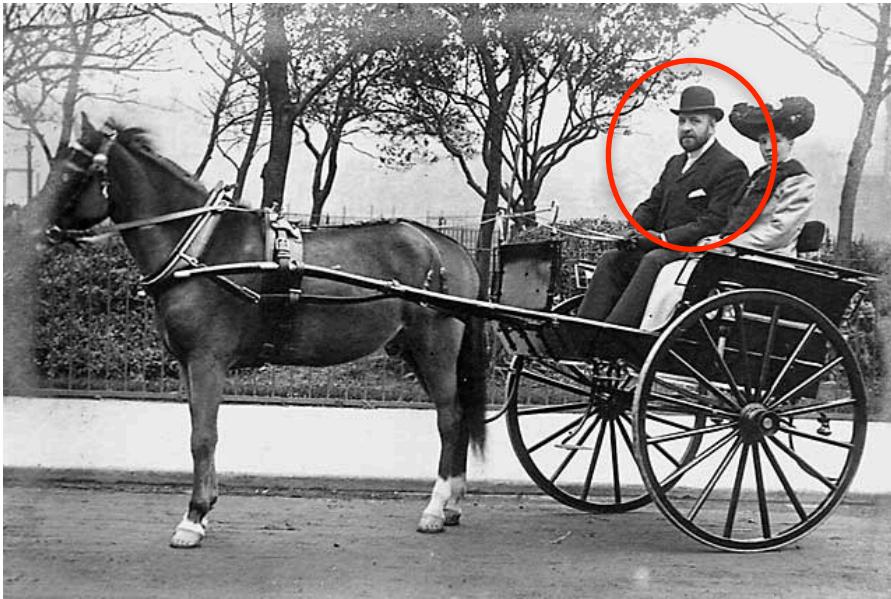


# Véhicule autonome avec Lego Mindstorms EV3

Version 1.0 - 21 janvier 2017

Niveau  
Avancé

# Quel est le point commun ?



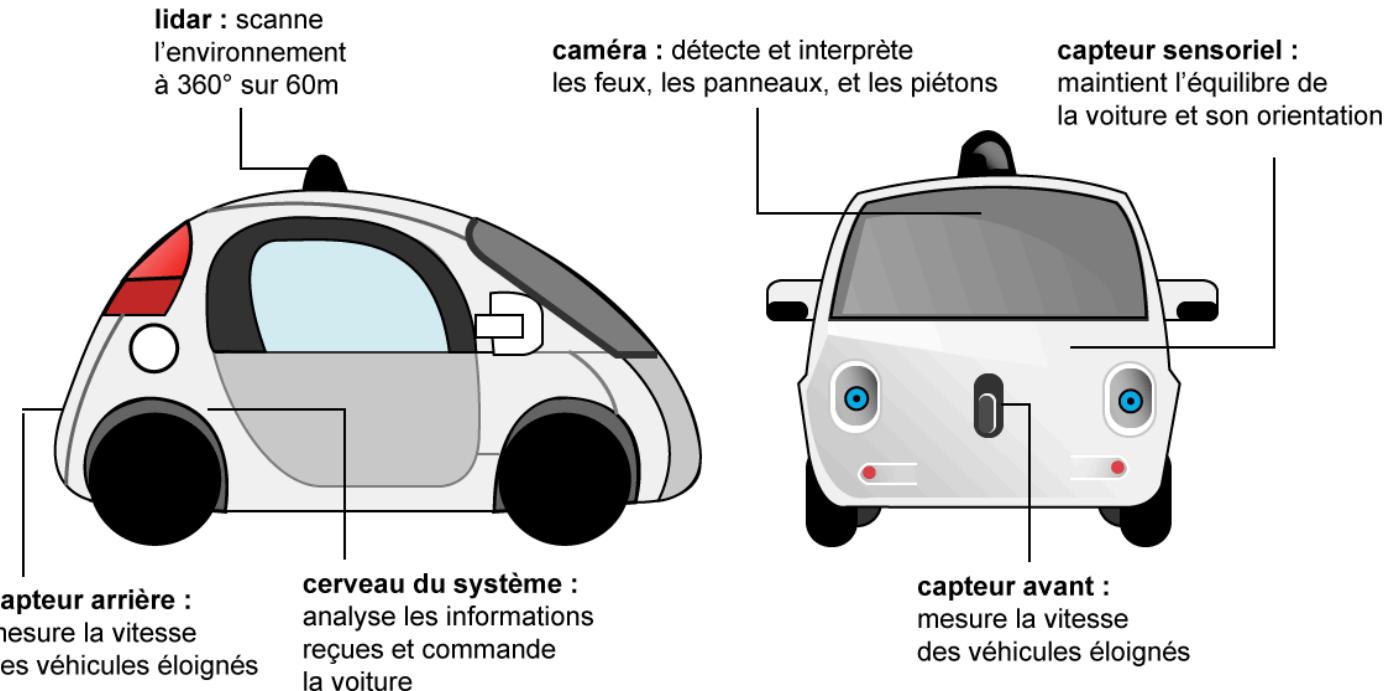
# Problèmes: Bouchons, pollution, accidents



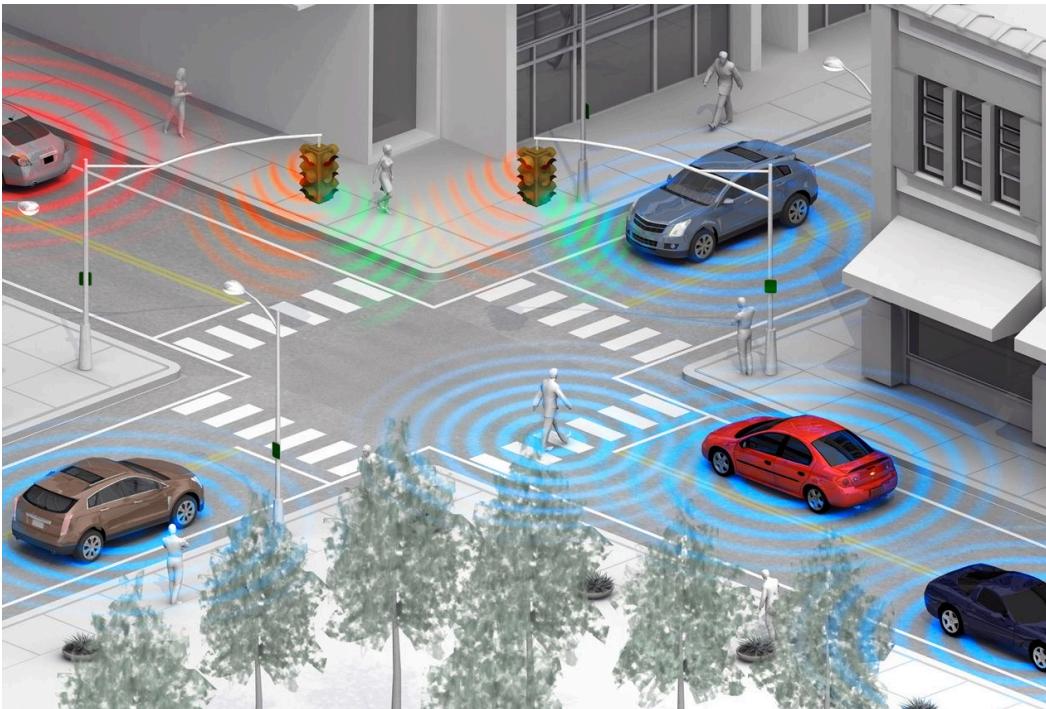
# Solution 1: Voiture électrique



# Solution 2: Voiture autonome



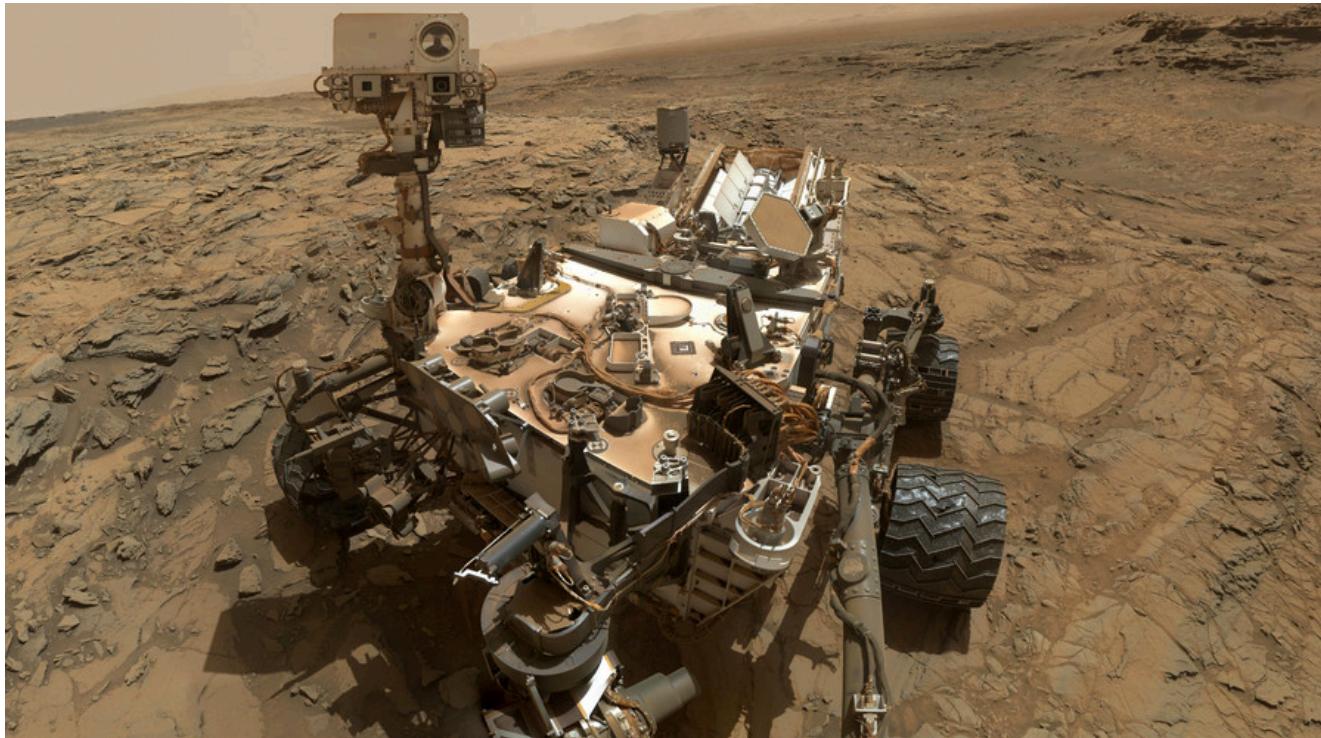
# Fluidifie et sécurise le trafic ...



# ... et nous rend moins bête ;-)



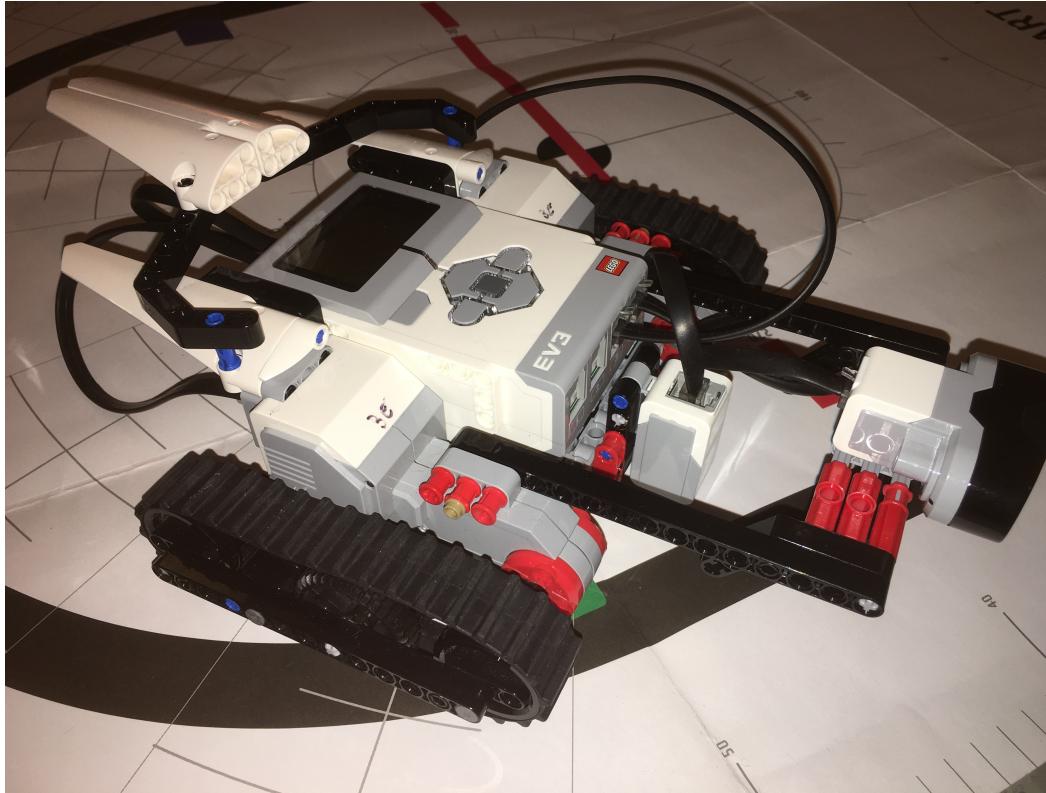
# Sur Terre et même sur Mars



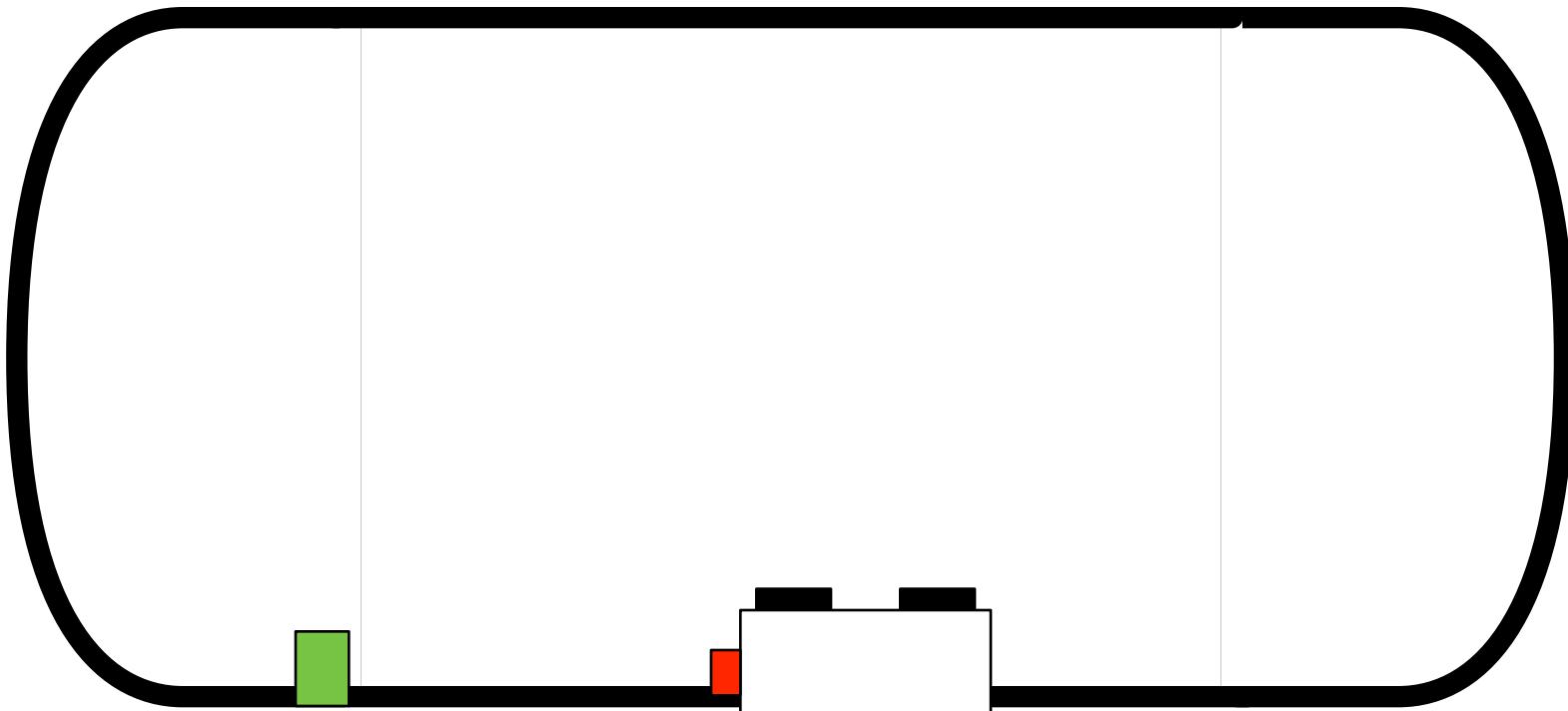
Véhicule autonome Mars Curiosity



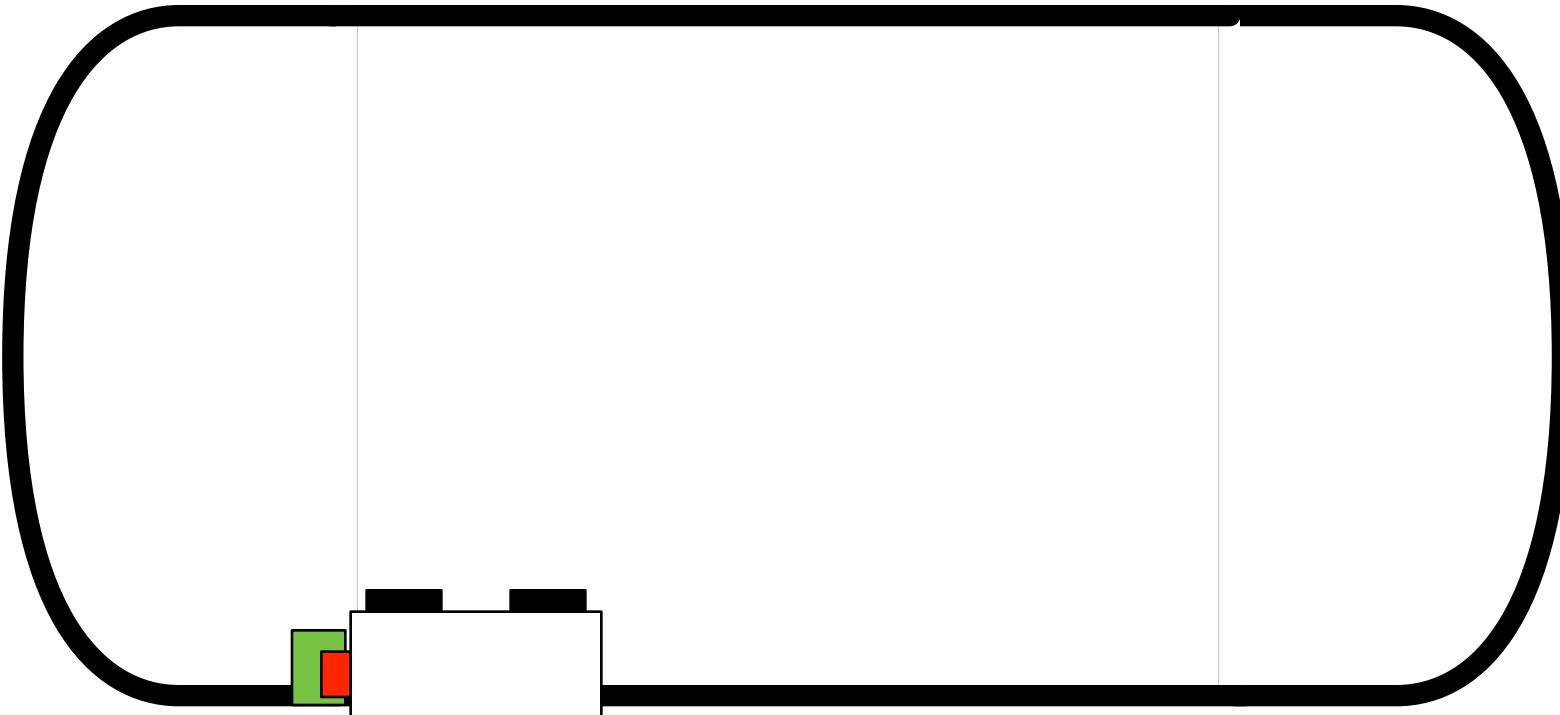
# Notre smart-car



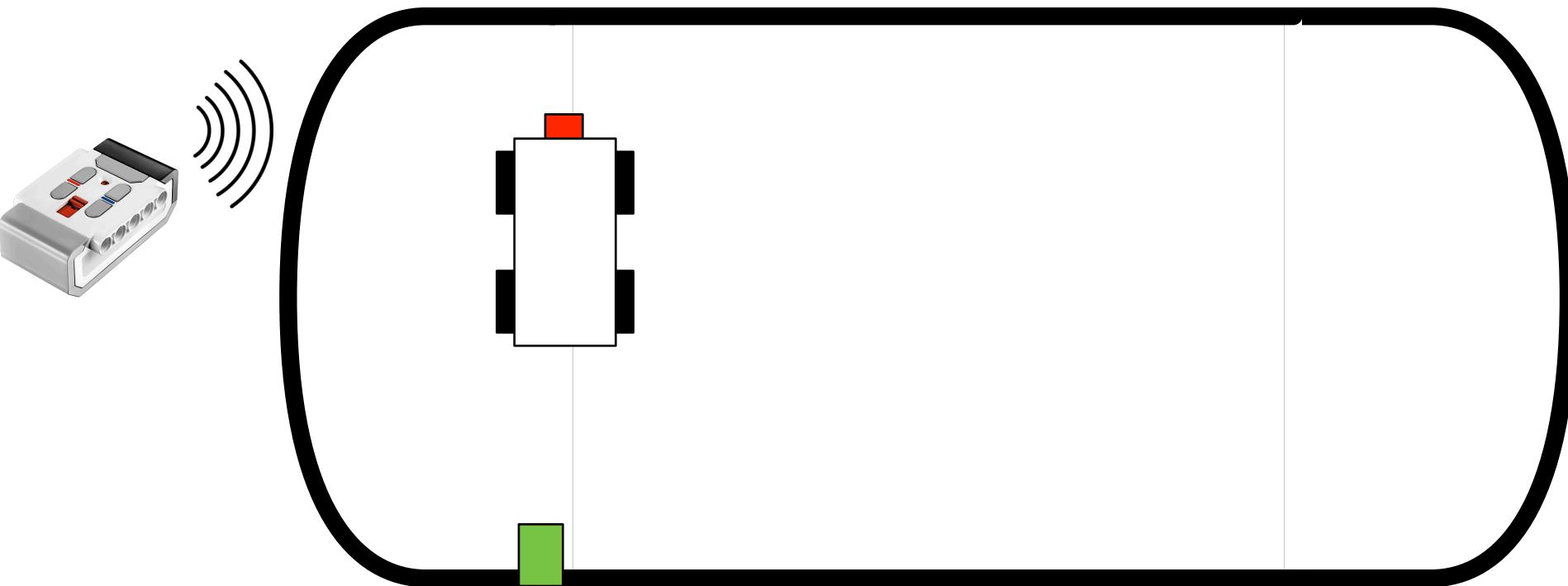
# Trouver l'entrée du parking



# Se parquer

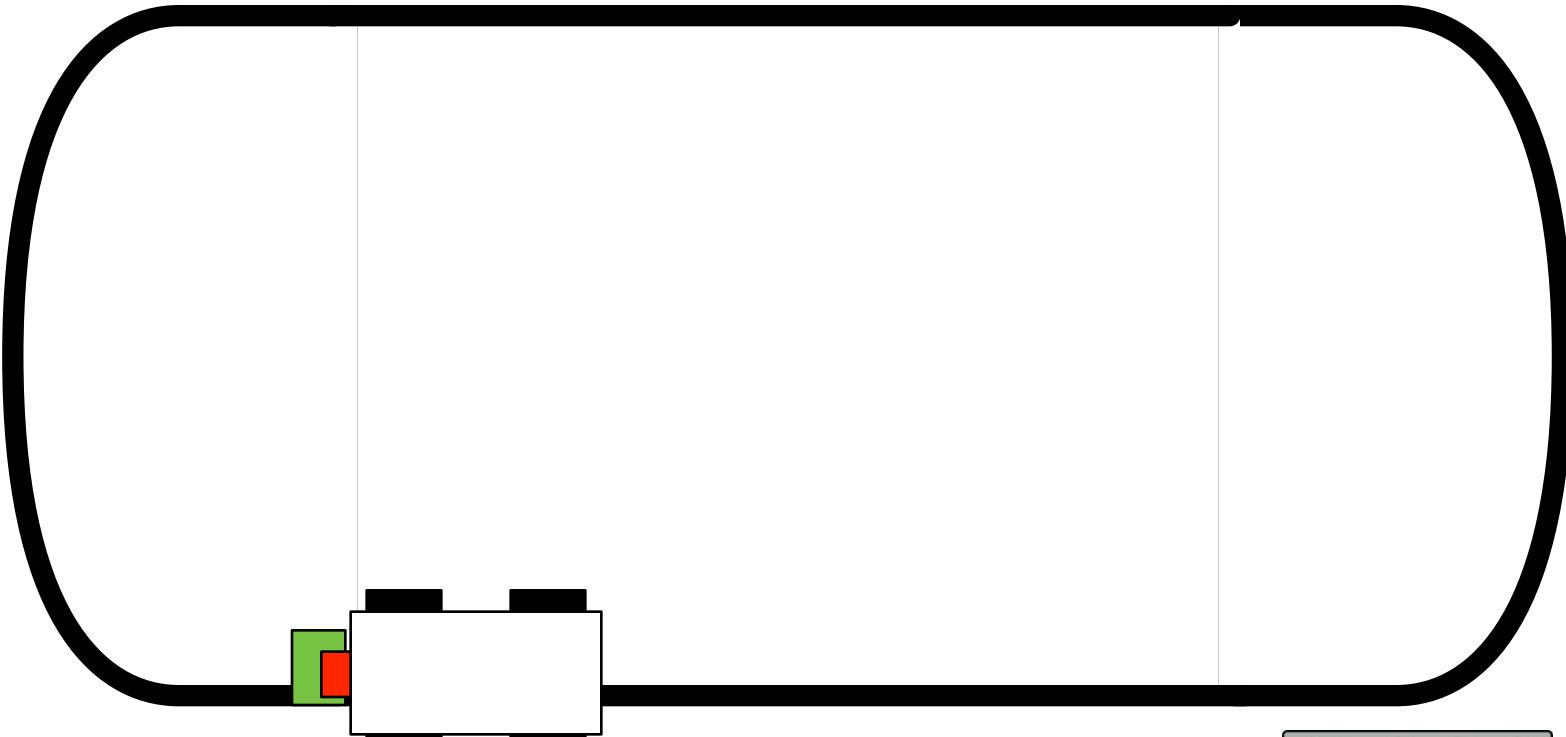


# Quitter le parking

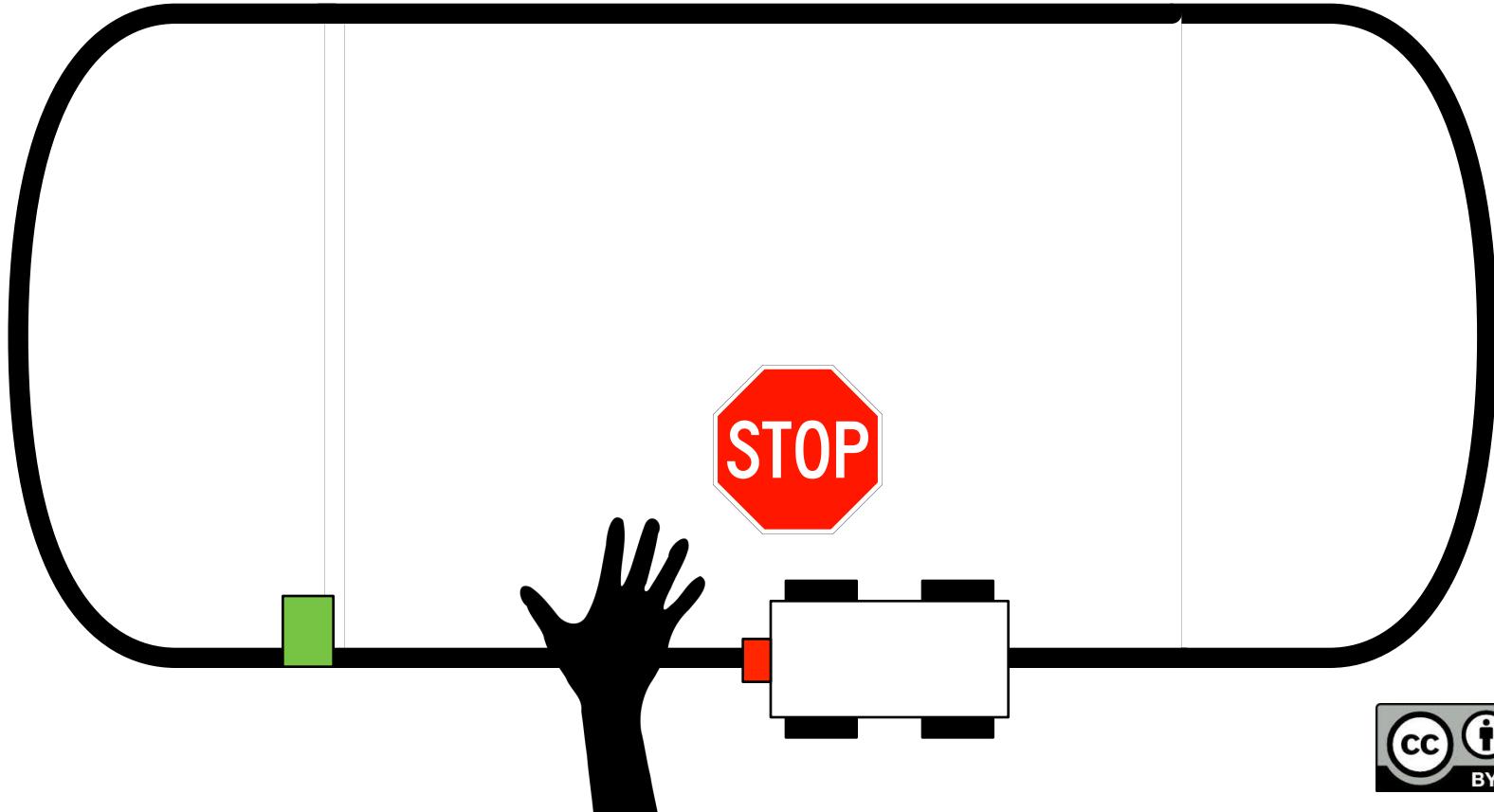


BY NC SA

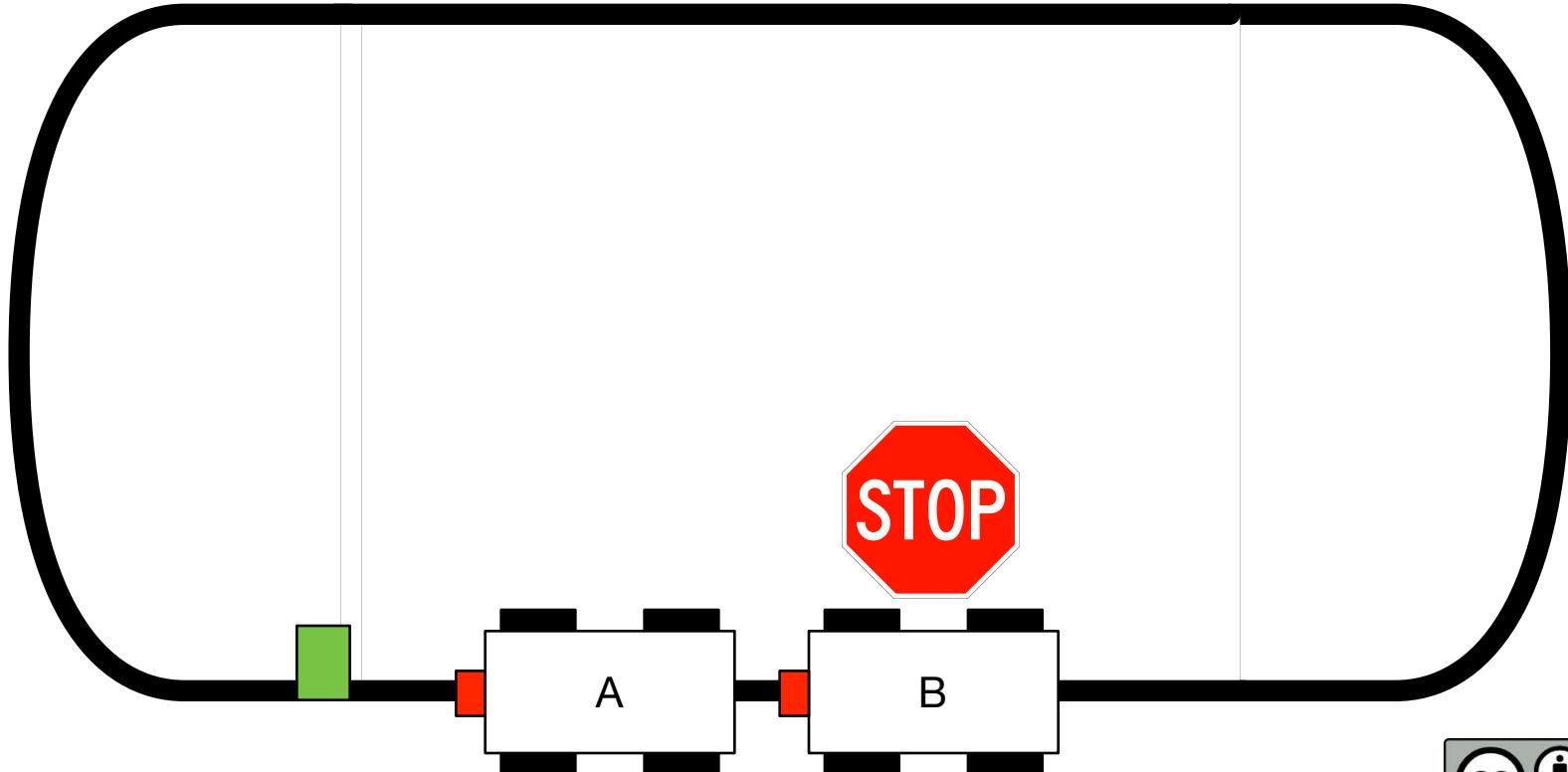
# Reprendre la route



# Éviter les collisions



# Éviter les collisions



# De quoi est composé ton robot ?



- Une **brique EV3** sur laquelle sont reliés les capteurs et les moteurs : c'est le **cerveau** du robot. Dans cette brique, nous téléchargerons les programmes faits sur le PC.



- 2 **moteurs de grande puissance** pour diriger le robot.



# De quoi est composé ton robot ?



- 1 capteur de couleur capable de détecter 7 couleurs, mais également la luminosité ambiante et l'intensité de la lumière réfléchie.
- 1 capteur infrarouge pour détecter les objets à distance (mode proximité - 70 cm max) et recevoir des ordres transmis via la commande à distance (mode balise - portée de 200 cm).



# De quoi est composé ton robot ?



- 1 télécommande infrarouge pour transmettre des commandes au capteur infrarouge.



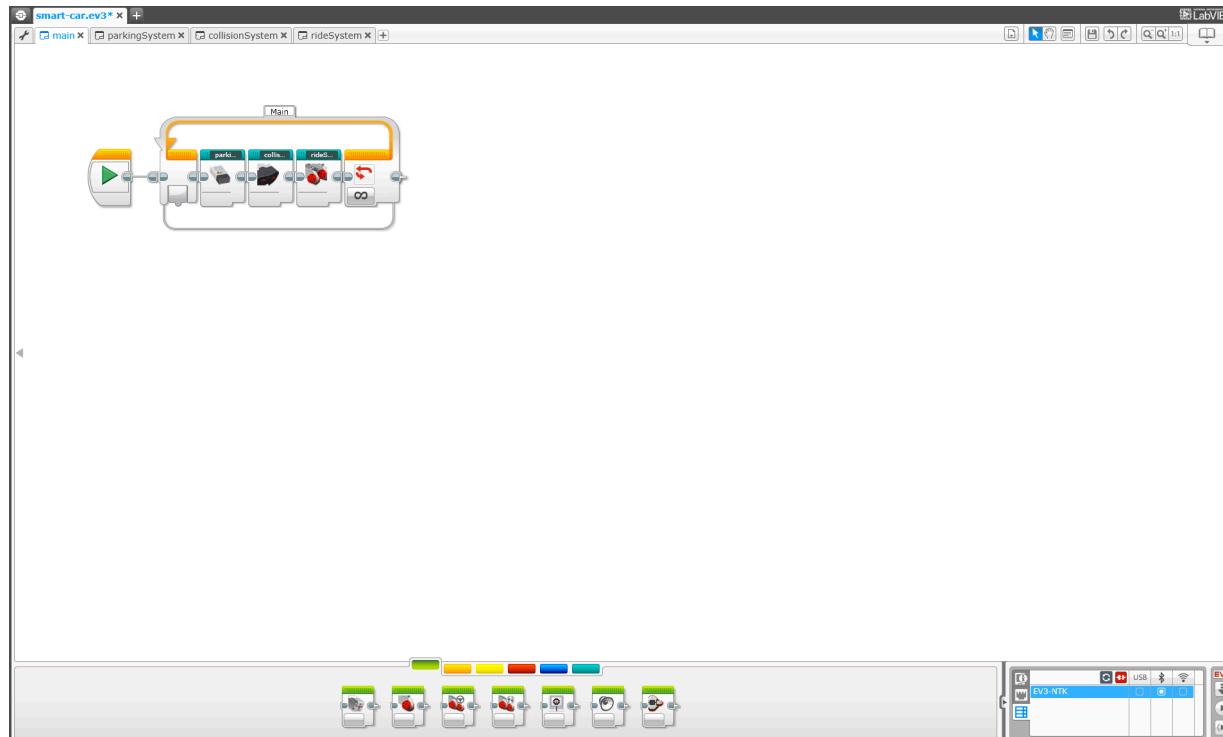
# Comment programmer pour EV3 ?



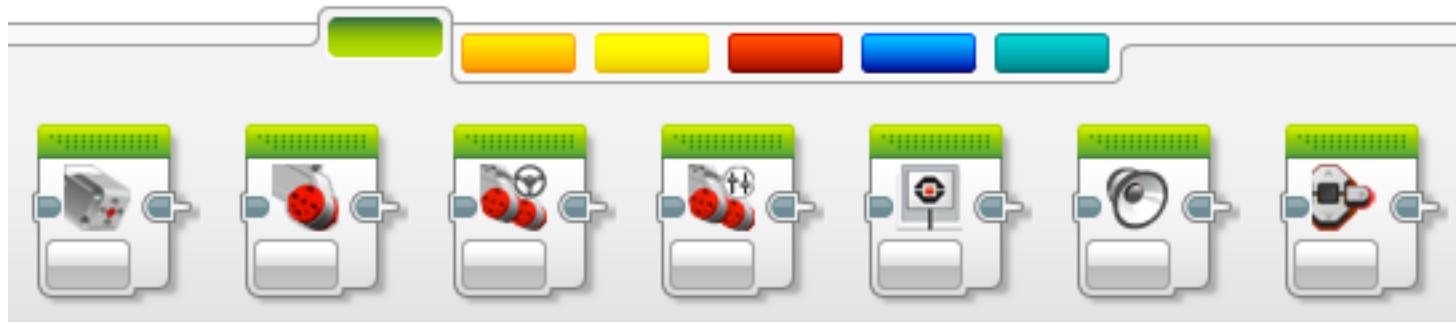
- Les instructions sont sous forme de **briques graphiques** que l'on va ajouter les unes aux autres pour construire le programme. On peut :
  - faire **avancer** le robot,
  - le faire **attendre**,
  - lui faire **jouer un son**,
  - lui faire **afficher des images**, etc.



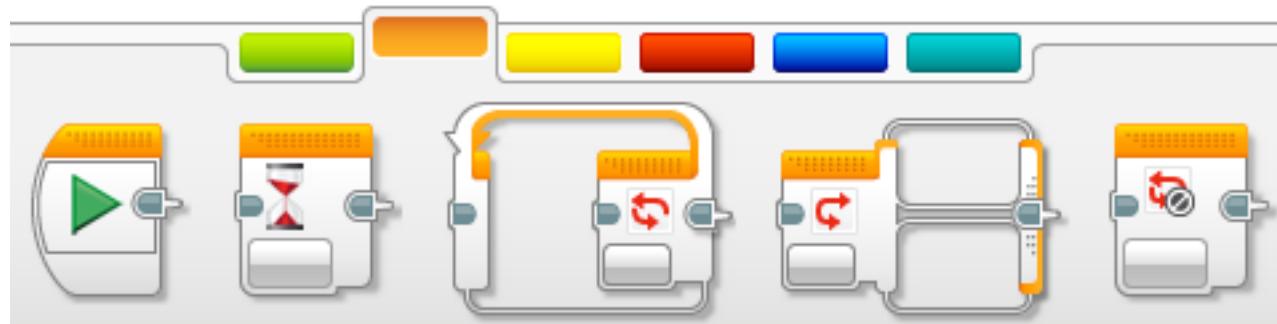
# Espace de travail



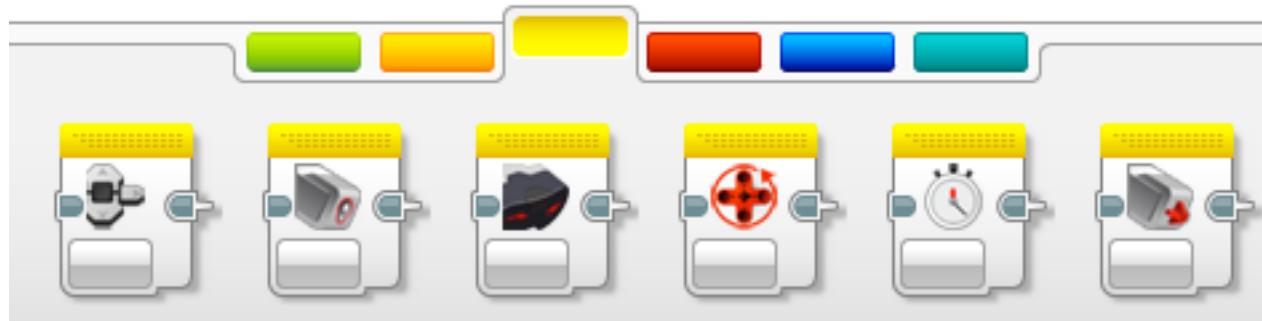
# Actions



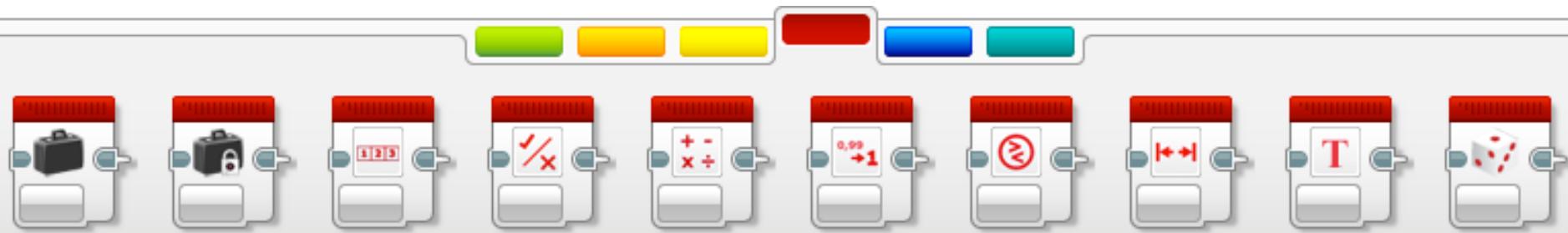
# Contrôles du déroulement



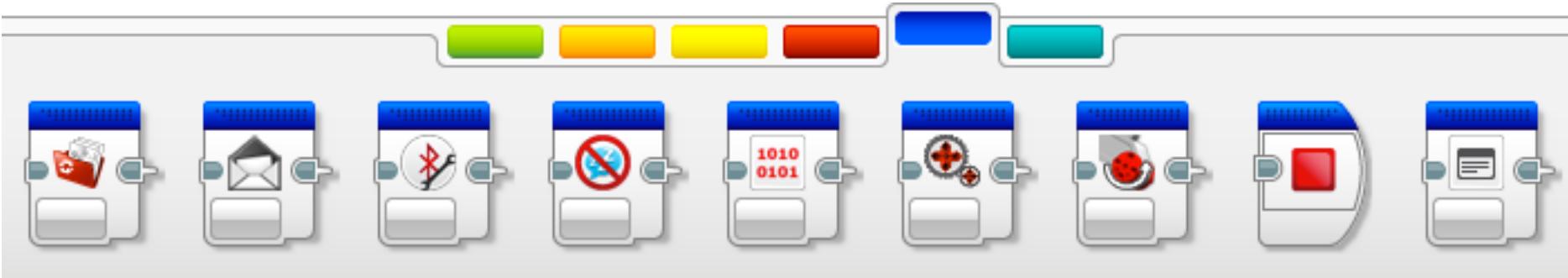
# Capteurs



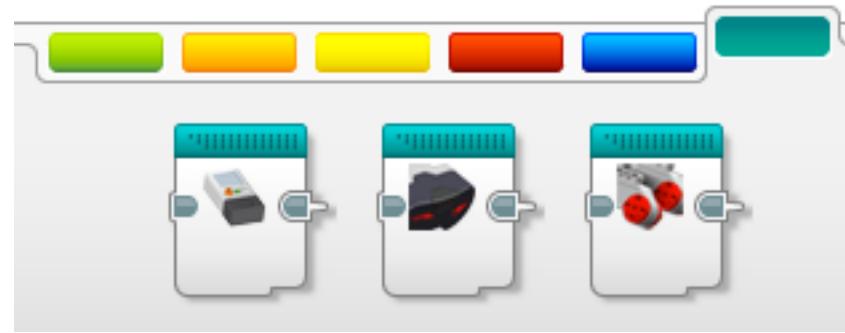
# Opérations sur les données



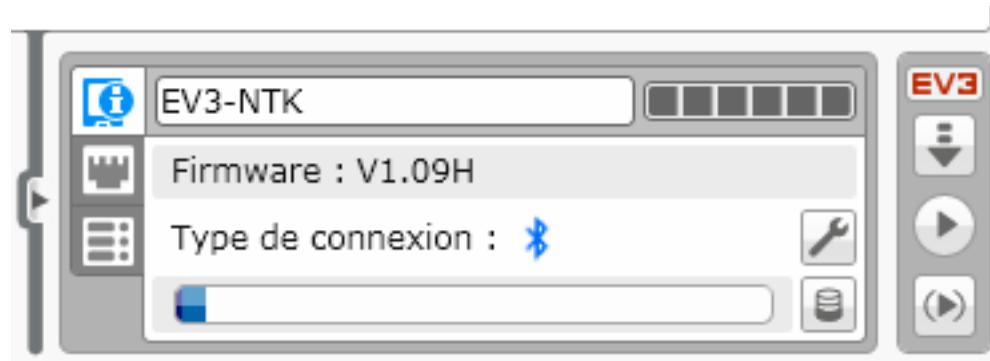
# Mode Avancé



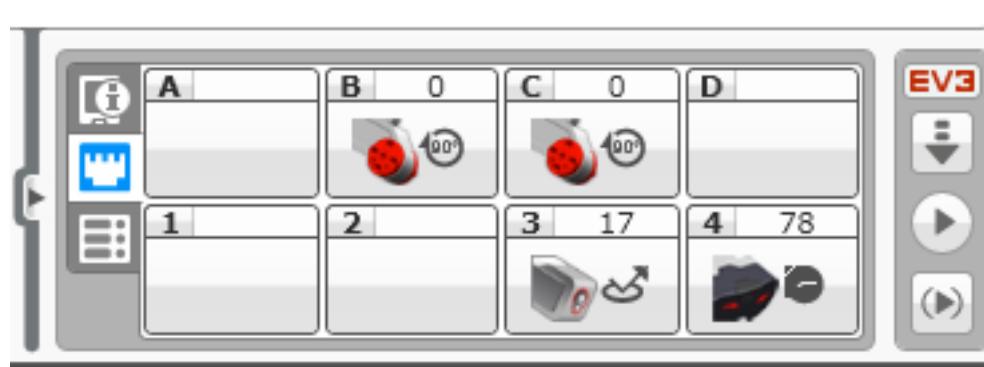
# Mes blocs



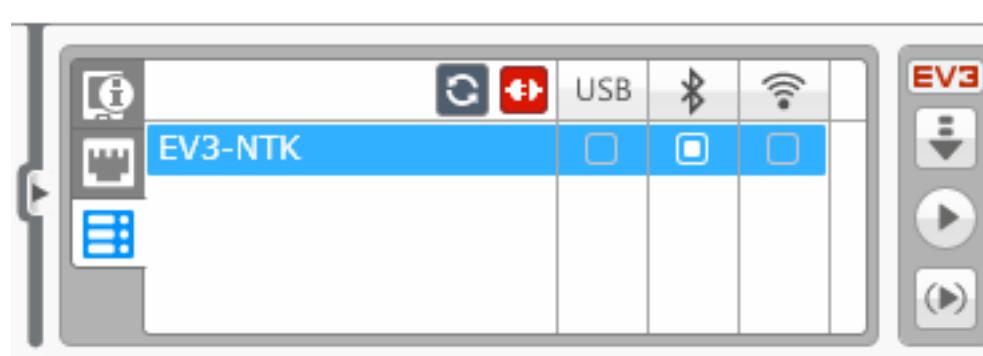
# Informations de la brique EV3



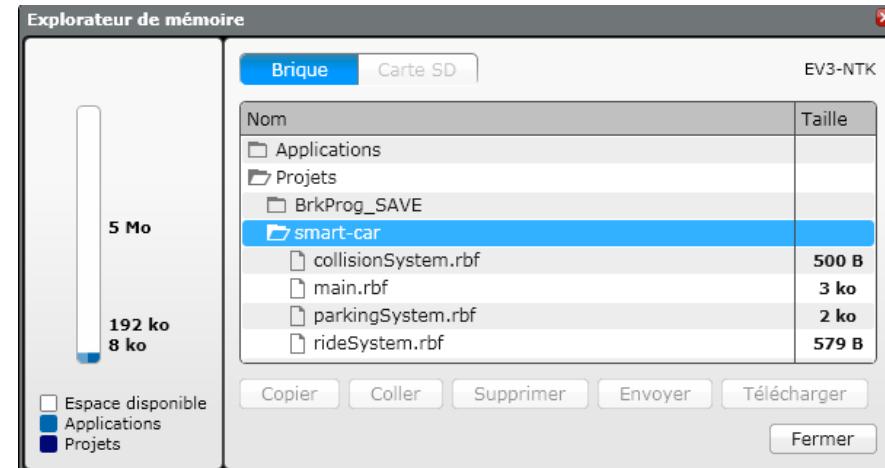
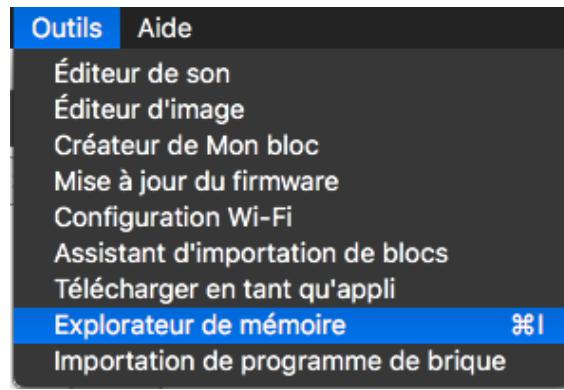
# Affichage des ports



# Se connecter à EV3



# Explorer la mémoire de EV3



# Propriétés du projet

smart-car.ev3\* x +

Titre du projet : Smart-Car

1 IMAGE DU PROJET

2

Ce projet est destiné à illustrer quelques fonctionnements d'un véhicule autonome.

Les fonctionnalités sont les suivantes:

- Suivre une piste
- Garer le véhicule lorsque la zone verte est détectée
- Quitter la zone de parking lorsqu'un bouton de la télécommande est appuyé
- Arrêter le véhicule lorsqu'un obstacle est détecté

Mode Connexion en série

Type	Nom
main.ev3p	

Programmes Images Sons Mes blocs Variables Objets exportables

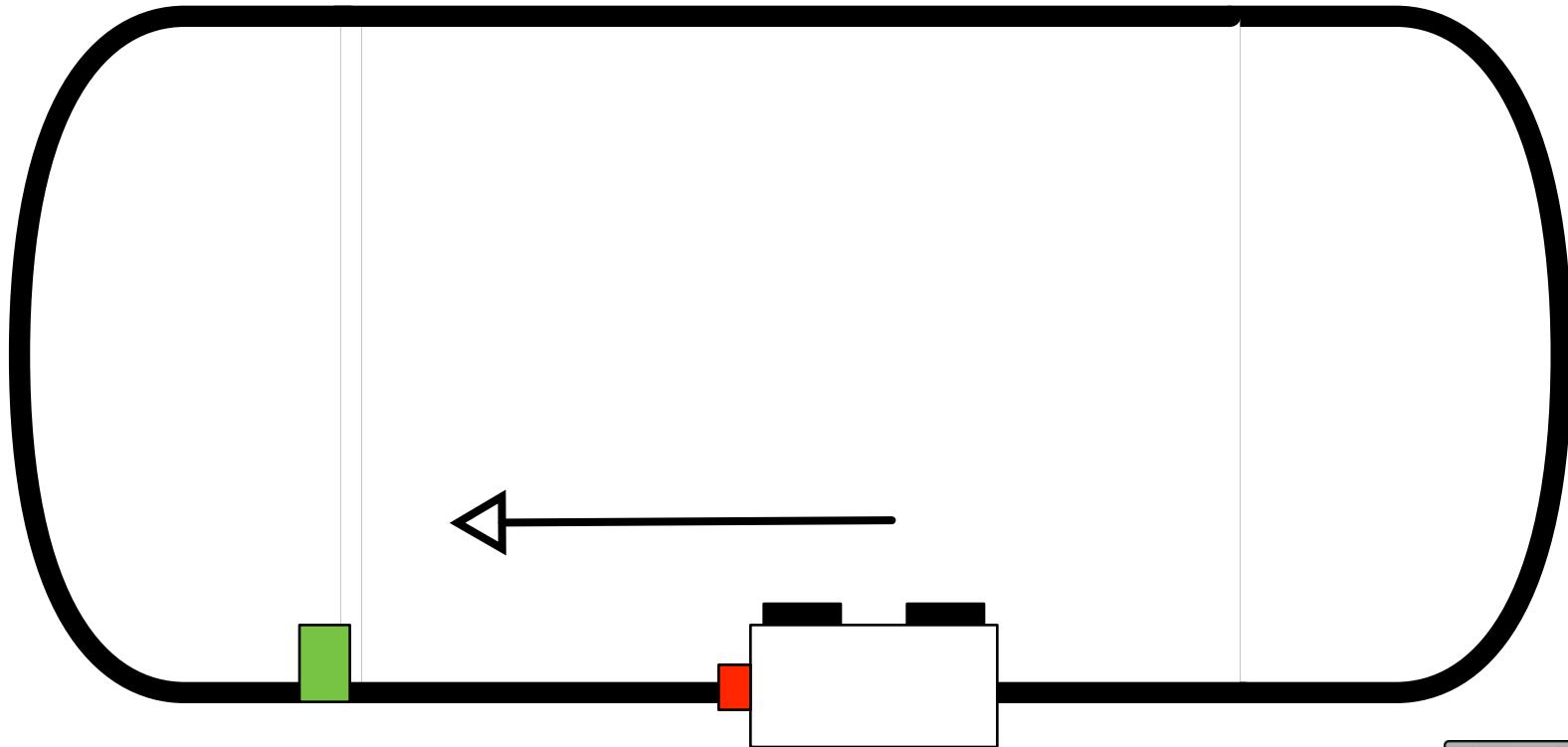


# Notre mission

1. Le véhicule roule sur la piste (ligne noire)
2. Lorsque la balise verte est détectée, le véhicule se gare
3. Le véhicule revient sur la piste lorsque le bouton de la télécommande est appuyé
4. Si le véhicule détecte un obstacle, il s'arrête et reprend sa route lorsque l'obstacle n'est plus présent



# Mission 1: Suivre la piste



# Mission 1: Instructions

Crée un nouveau projet (“missions”) et renomme le programme “mission01”.

Le véhicule suit la piste noire du Test Pad.

La puissance des moteurs doit être ajustée pour permettre au véhicule de rester sur la piste.

Tu auras besoin des blocs suivants:



# Mission 1: Formules

Formules à utiliser pour ajuster la puissance des moteurs:

- moteur de droite =  $(r_{\max} - v) * \text{constante}$
- moteur de gauche =  $(v - r_{\min}) * \text{constante}$

**r\_max:** réflexion maximale obtenue à l'aide de l'affichage des ports

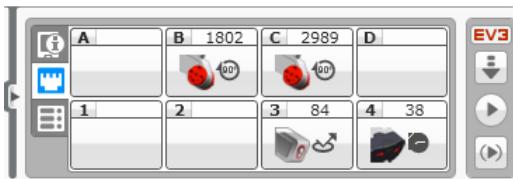
**r\_min:** réflexion minimale obtenue à l'aide de l'affichage des ports

**constante:** valeur comprise entre 0 et 1 pour obtenir un mouvement fluide

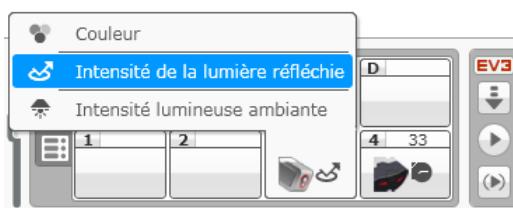


# Mission 1: Réflexion max et min

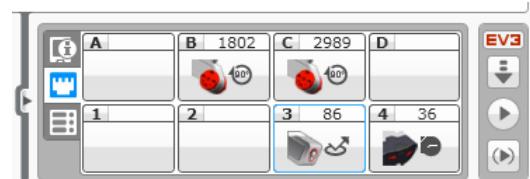
Récupère les valeurs de réflexion maximale et minimale comme suit:



Affichage des ports



Clique sur le capteur de couleur et choisis “Intensité de la lumière réfléchie”



Mesure les valeurs avec un fond blanc et noir



# Mission 1: Exécute le programme

Connecte le robot à l'ordinateur avec le câble USB

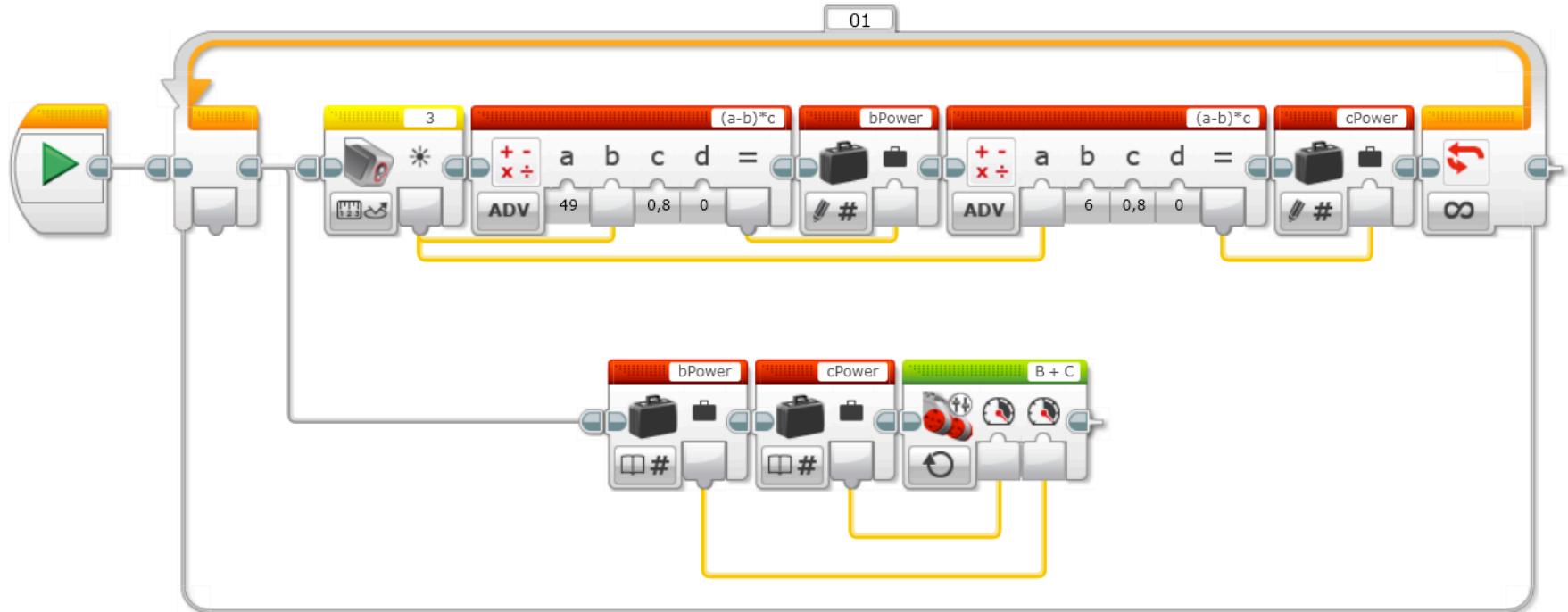
Télécharge le programme dans la brique du robot



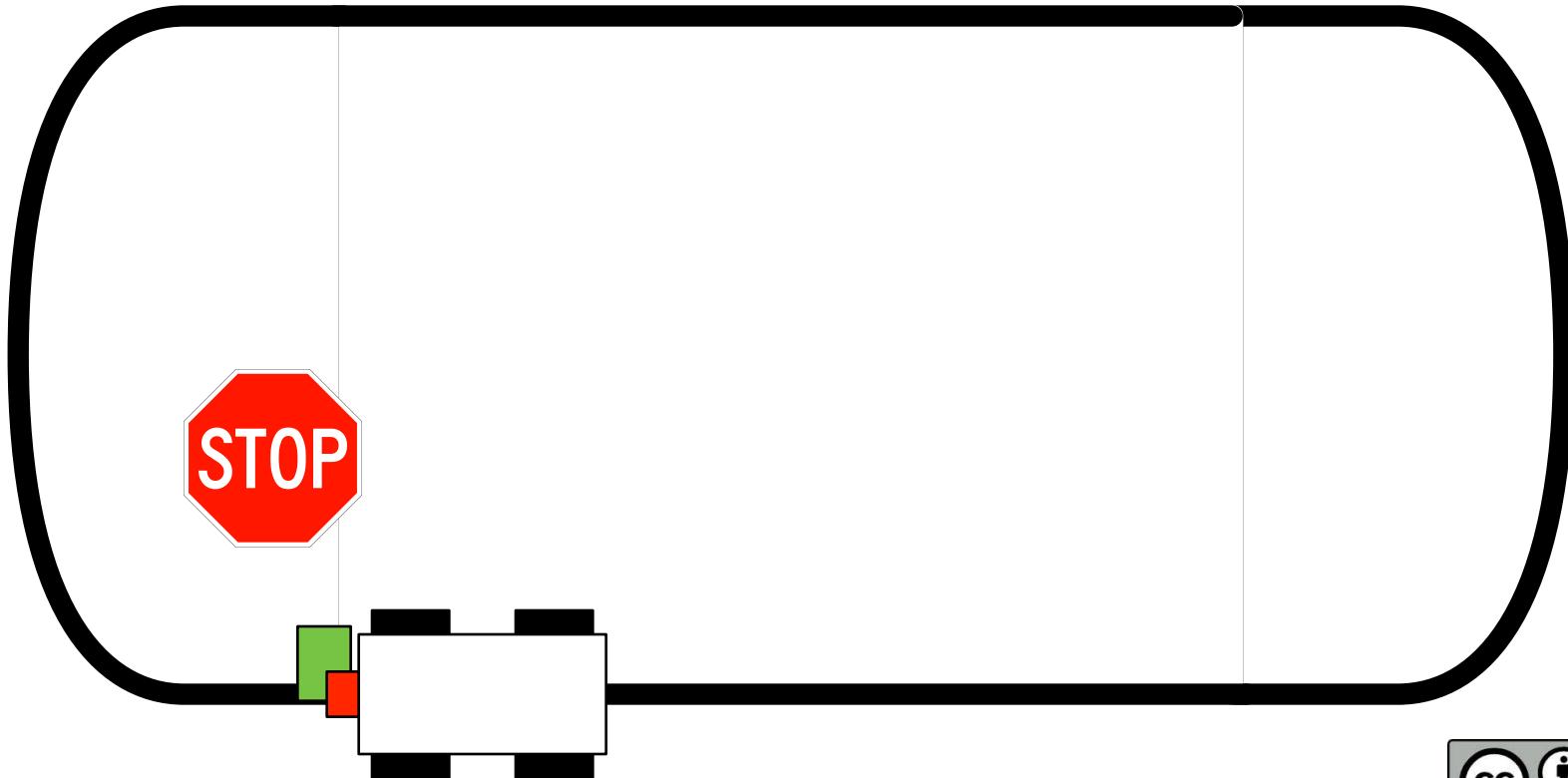
Débranche le robot de l'ordinateur, pose-le sur le Test Pad et lance le programme (Icône Dossier > missions > mission01)



# Mission 1: Une solution



# Mission 2: Détecte l'entrée du parking



# Mission 2: Instructions

Crée un nouveau programme et nomme-le “mission02”.

Reprends le code de la “mission01”.

Lorsque la balise verte est détectée, le véhicule s’arrête.

En plus des éléments de la “mission01”, tu auras besoin de l’élément suivant:



# Mission 2: Exécute le programme

Connecte le robot à l'ordinateur avec le câble USB

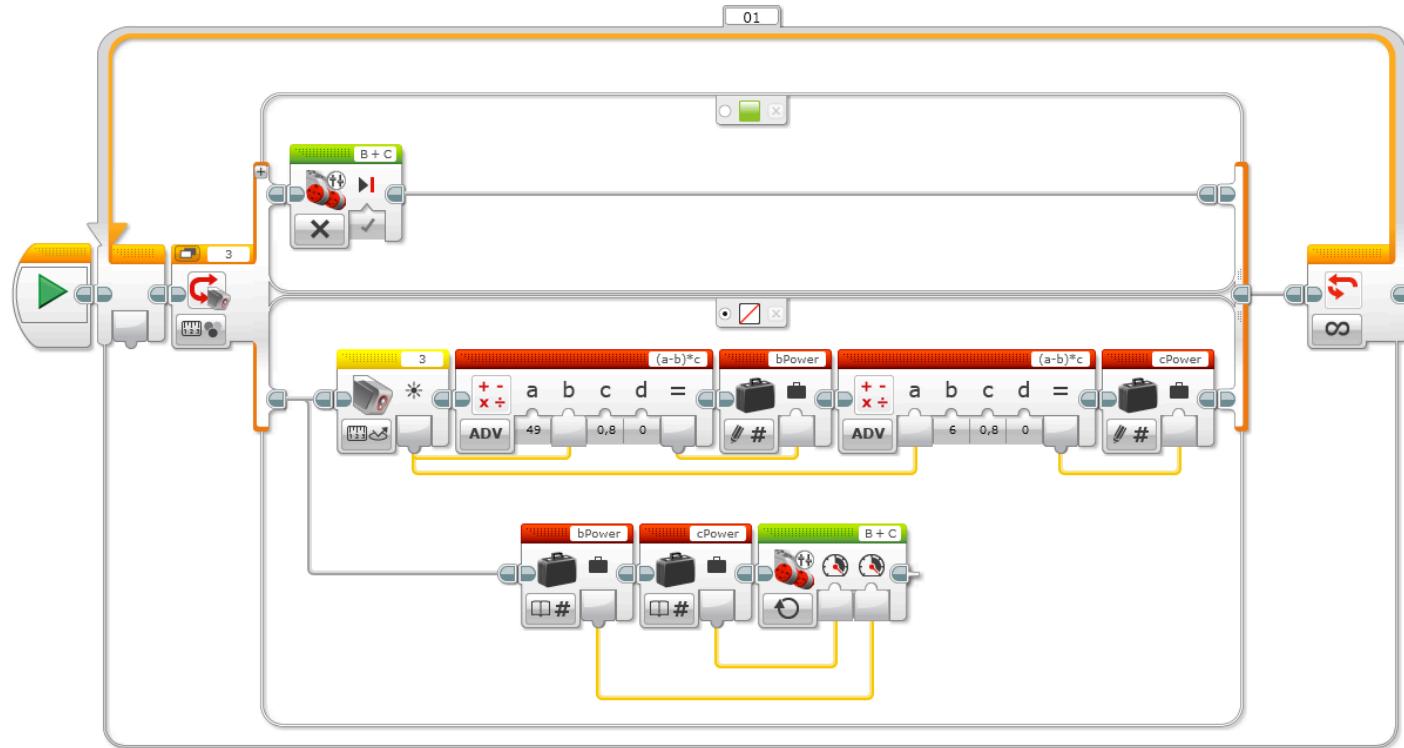
Télécharge le programme dans la brique du robot



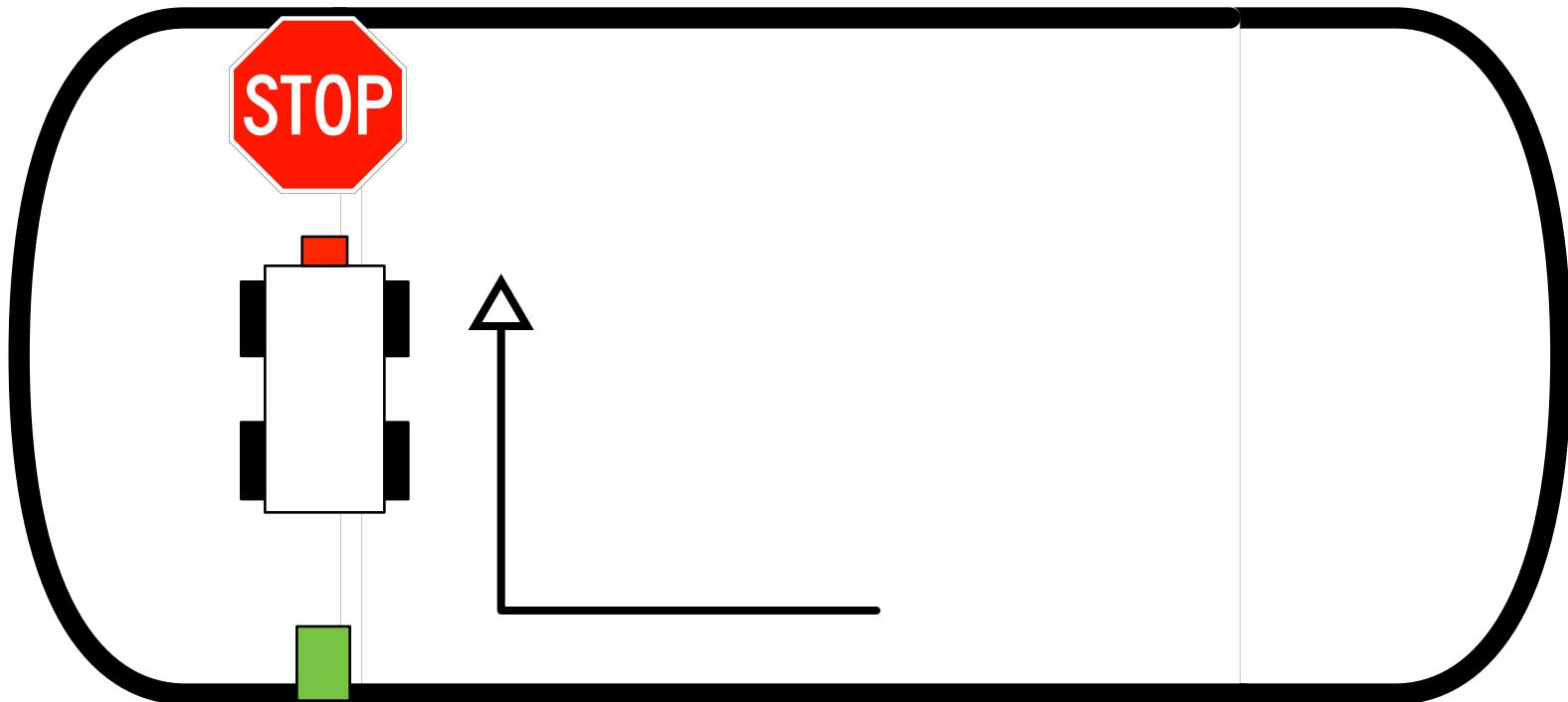
Débranche le robot de l'ordinateur, pose-le sur le Test Pad et lance le programme (Icône Dossier > missions > mission02)



# Mission 2: Une solution



# Mission 3: Se parker



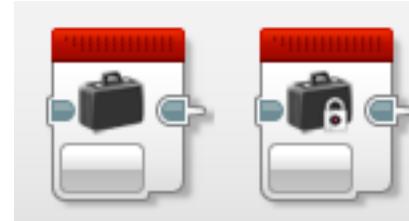
# Mission 3: Instructions

Crée un nouveau programme et nomme-le “mission03”.

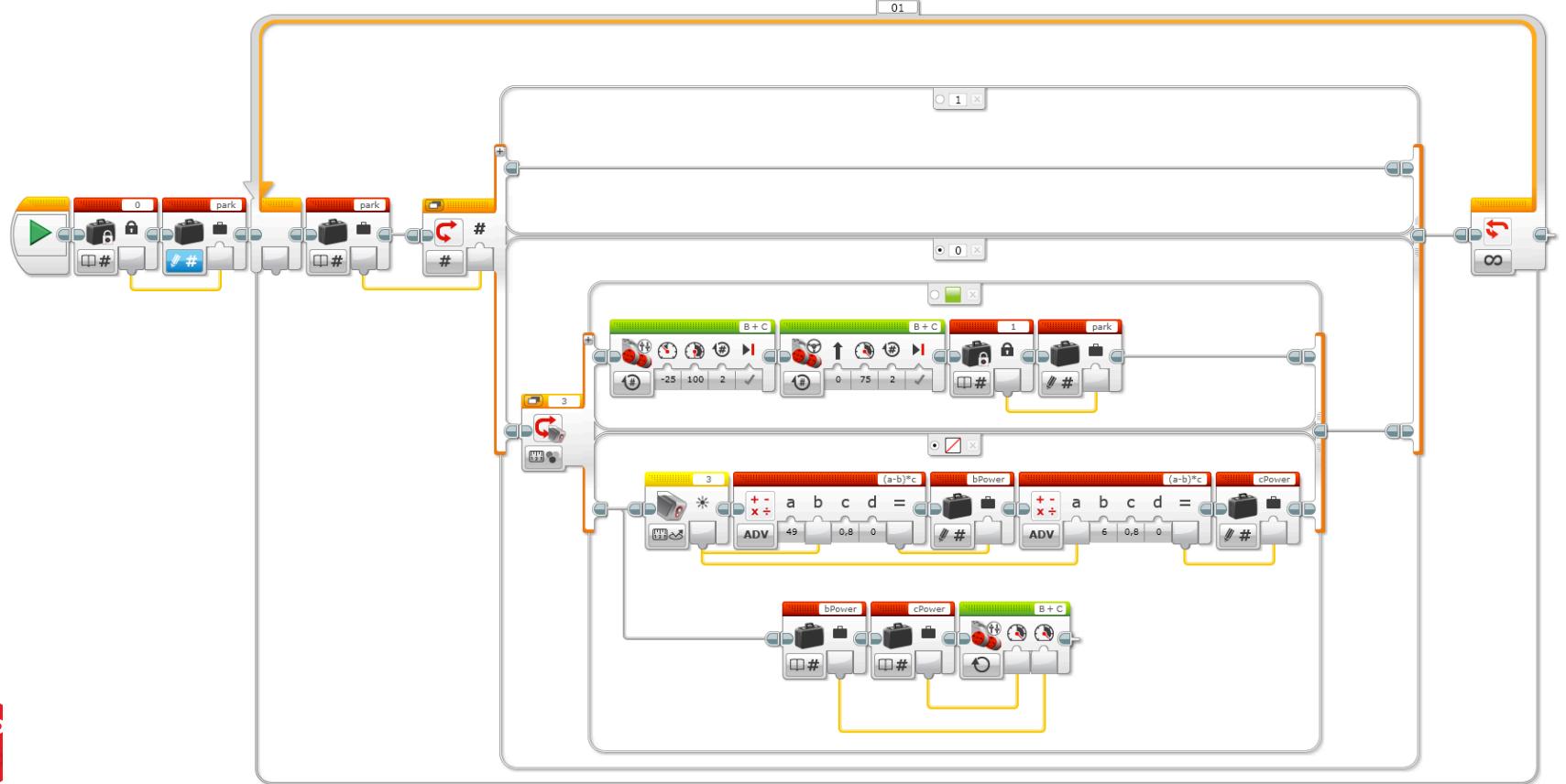
Reprends le code de la “mission02”.

Lorsque la balise verte est détectée, le véhicule se parque et s’arrête.

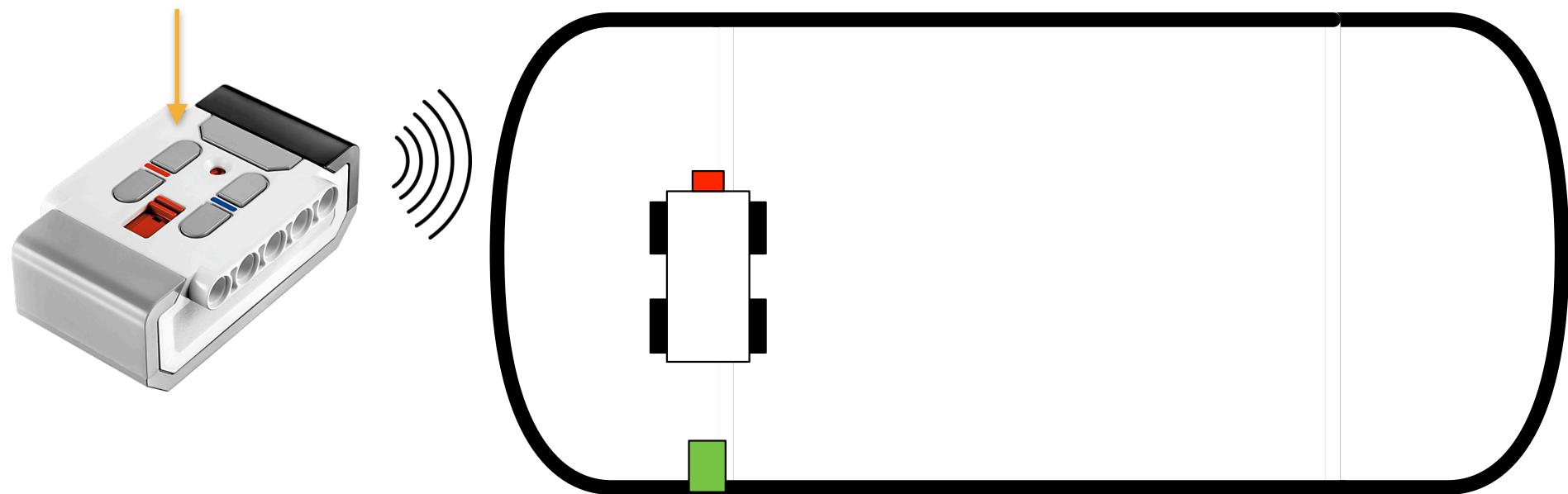
En plus des éléments de la “mission02”, tu auras besoin des éléments suivants:



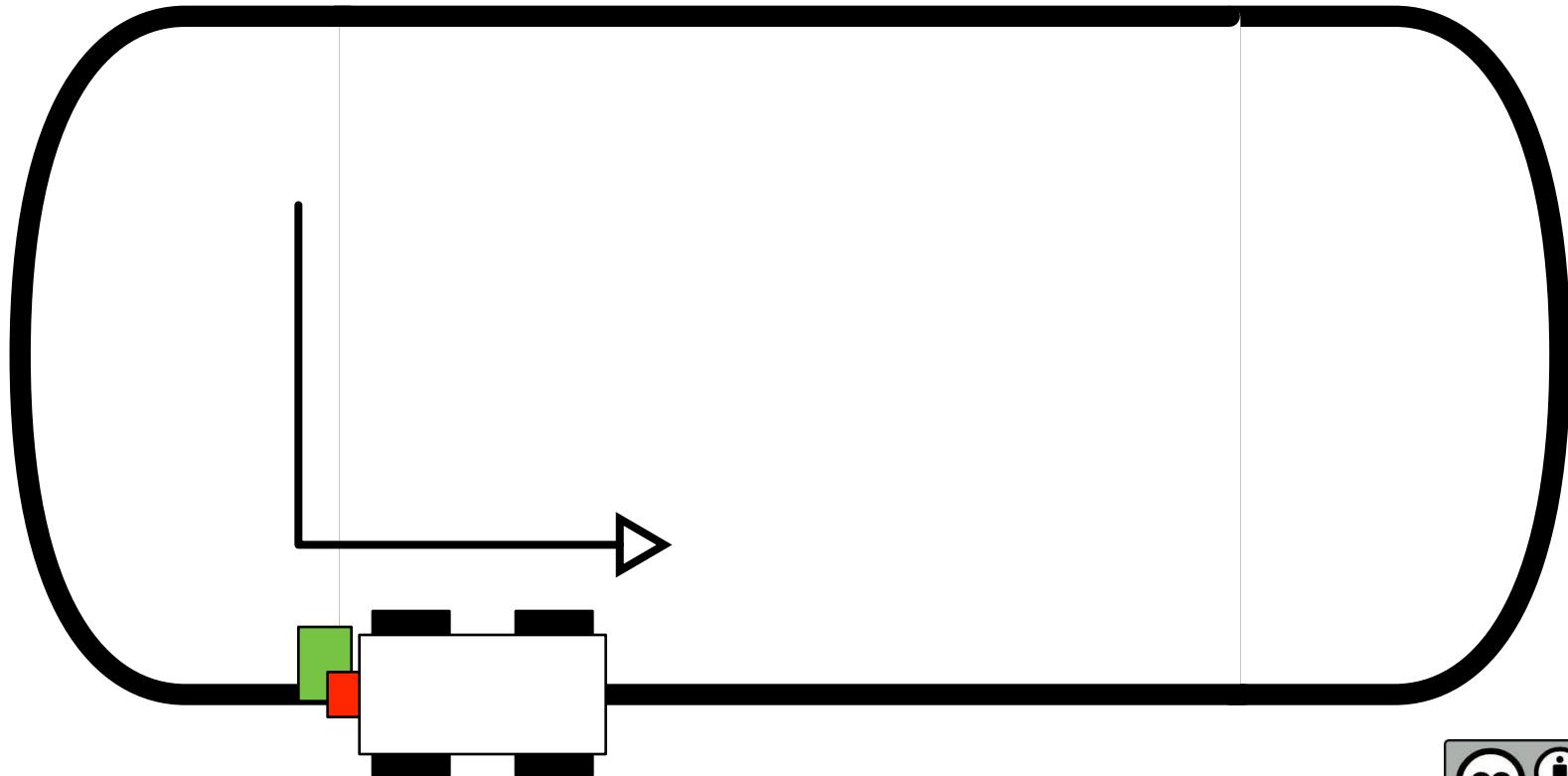
# Mission 3: Une solution



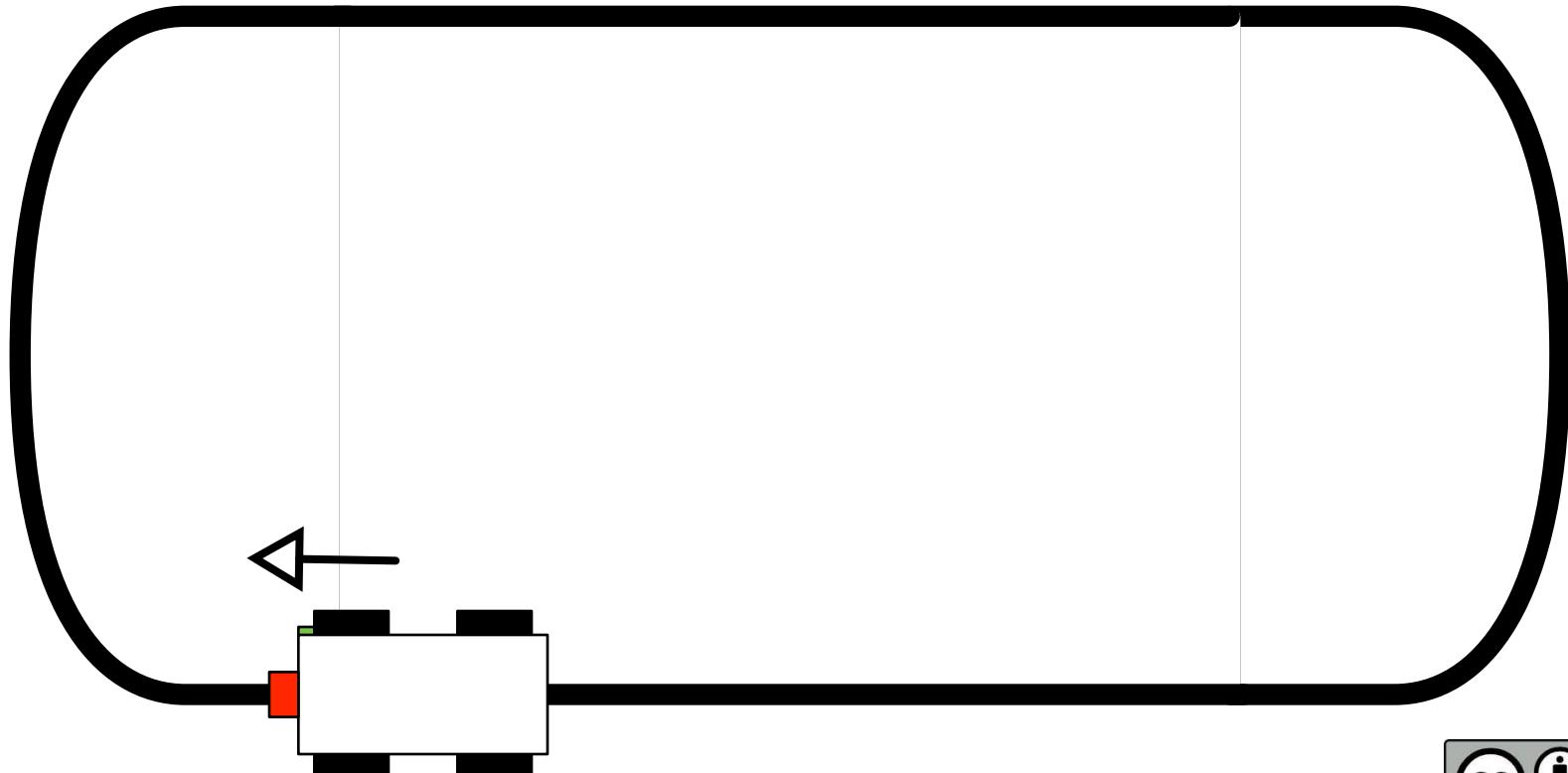
# Mission 4: Revenir sur la piste



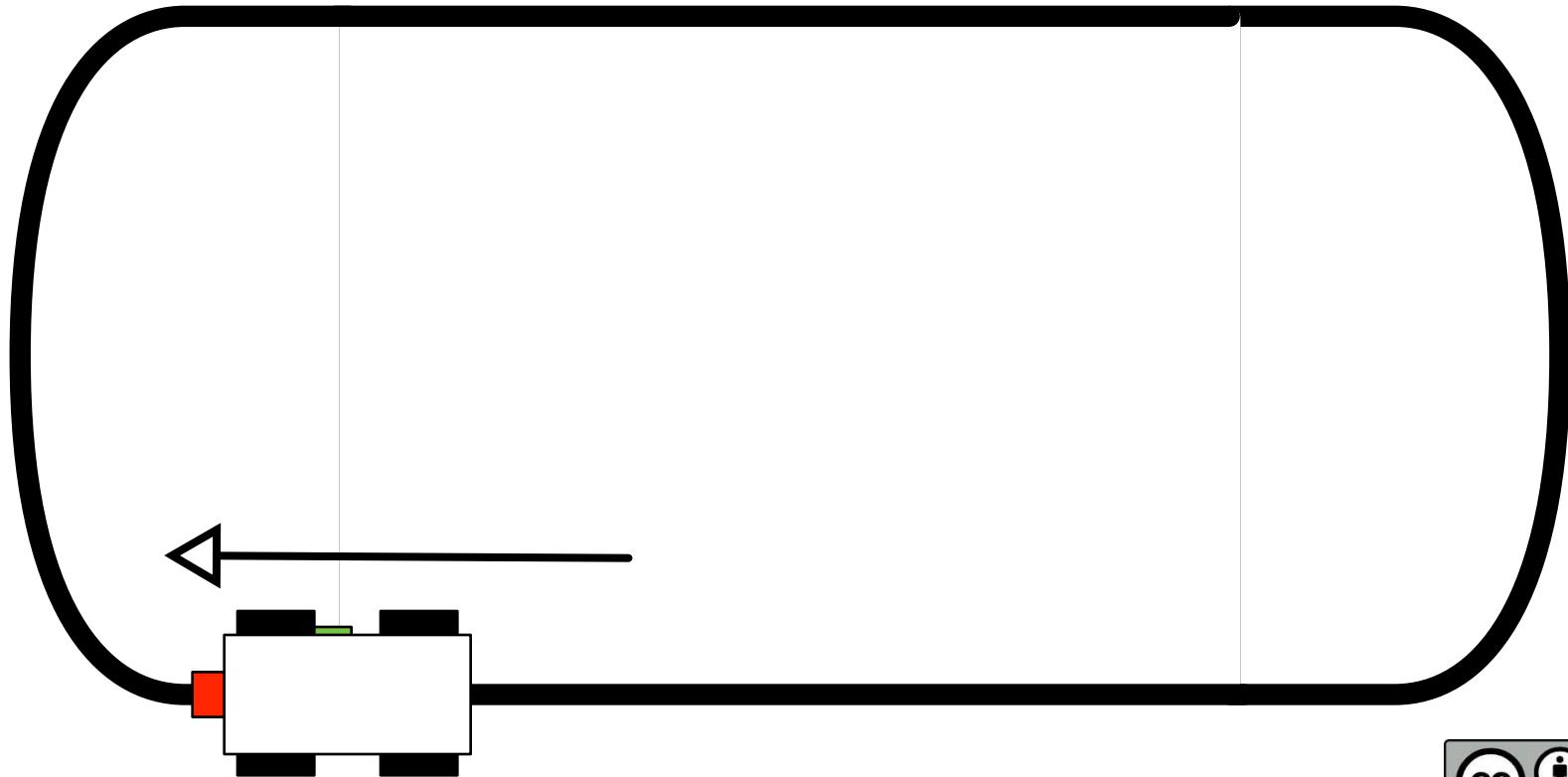
# Mission 4: Revenir sur la piste



# Mission 4: Revenir sur la piste



# Mission 4: Revenir sur la piste



# Mission 4: Instructions

Crée un nouveau programme et nomme-le “mission04”.

Reprends le code de la “mission03”.

Lorsque le bouton 1 de la télécommande est appuyé, le véhicule fait marche arrière et revient sur la piste.

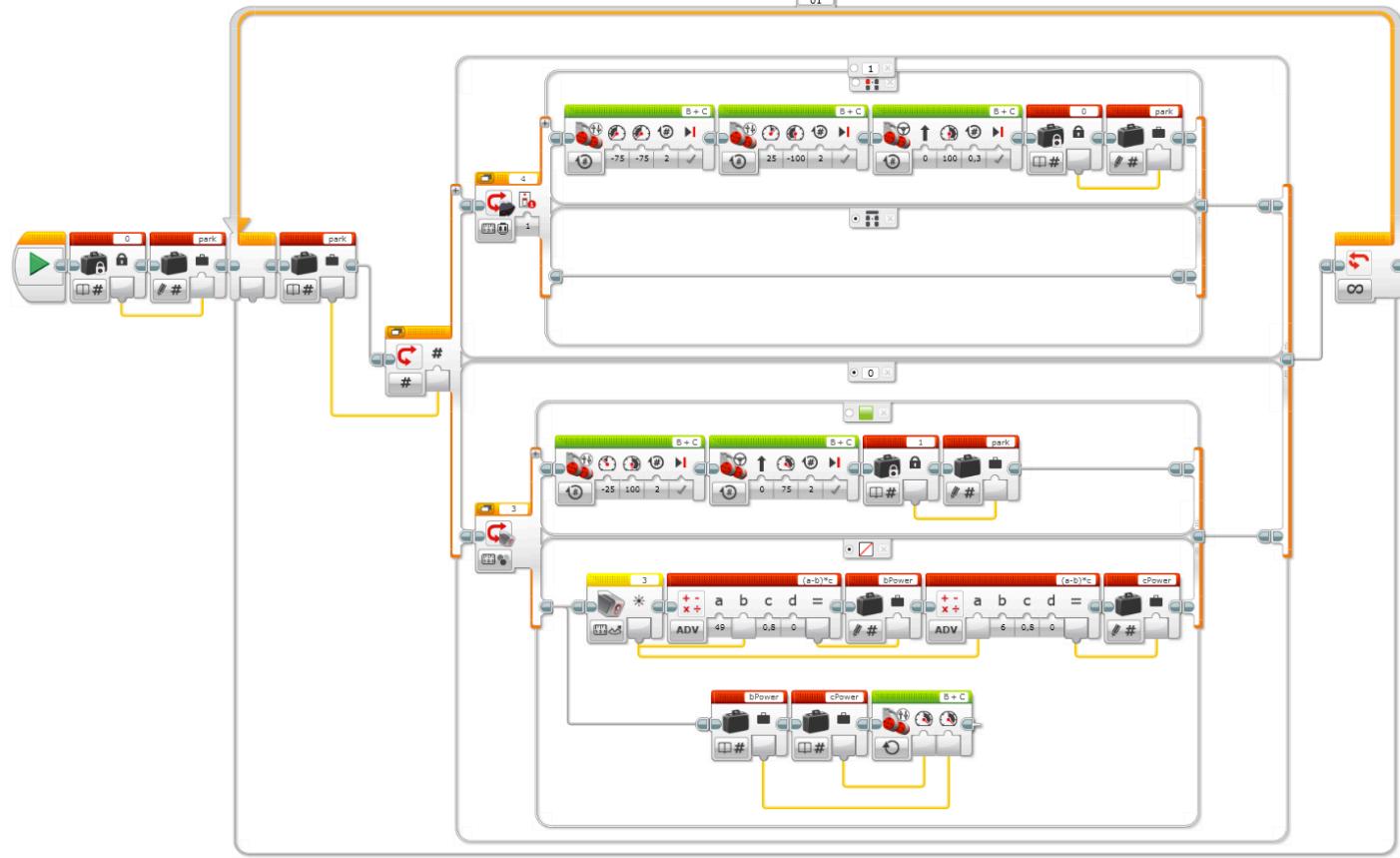
Le véhicule fait un petit déplacement pour quitter la balise verte et reprend automatiquement sa route.

Tu devras utiliser un sélecteur et l'ajuster sur: “Capteur infrarouge” - “Mesure” - “Télécommande”.

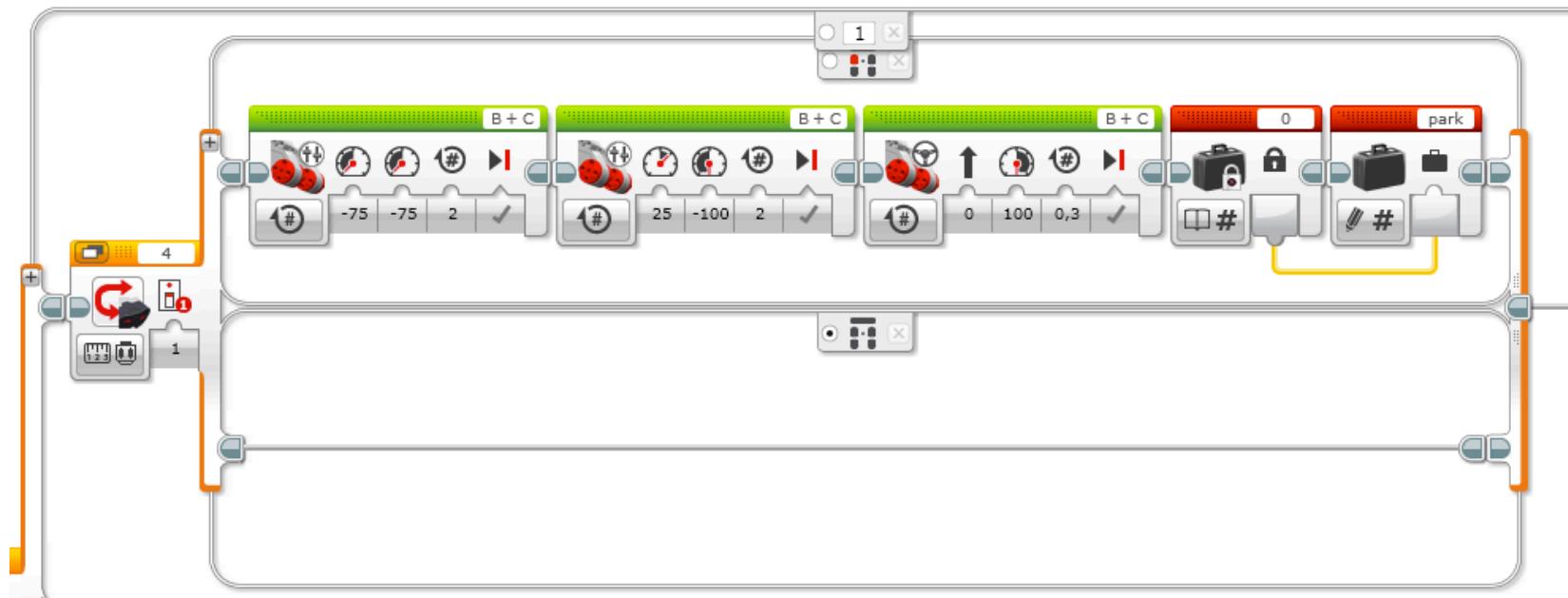
Choisis un canal de la télécommande qui ne va pas entrer en conflit avec les autres véhicules.



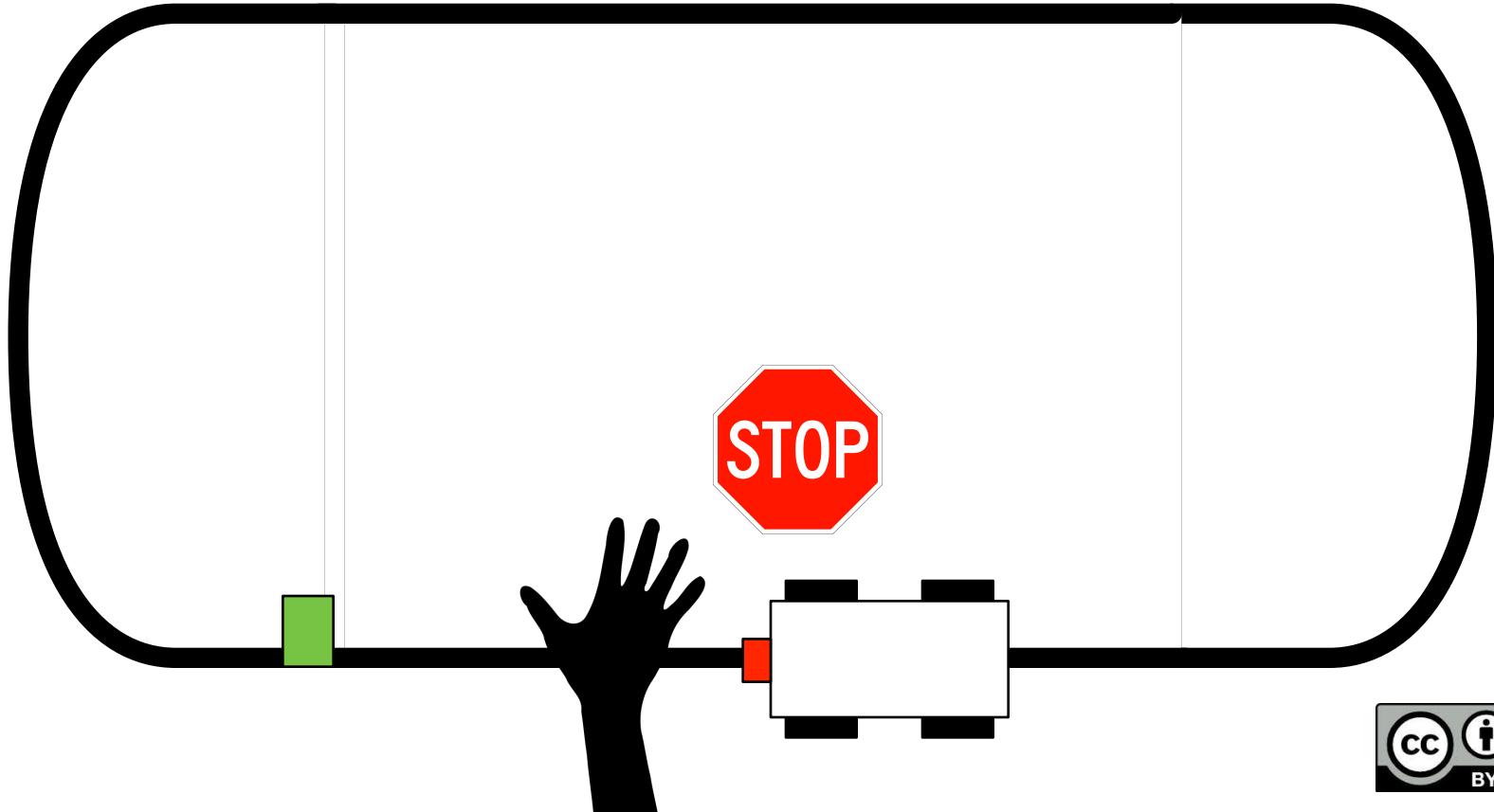
# Mission 4: Une solution



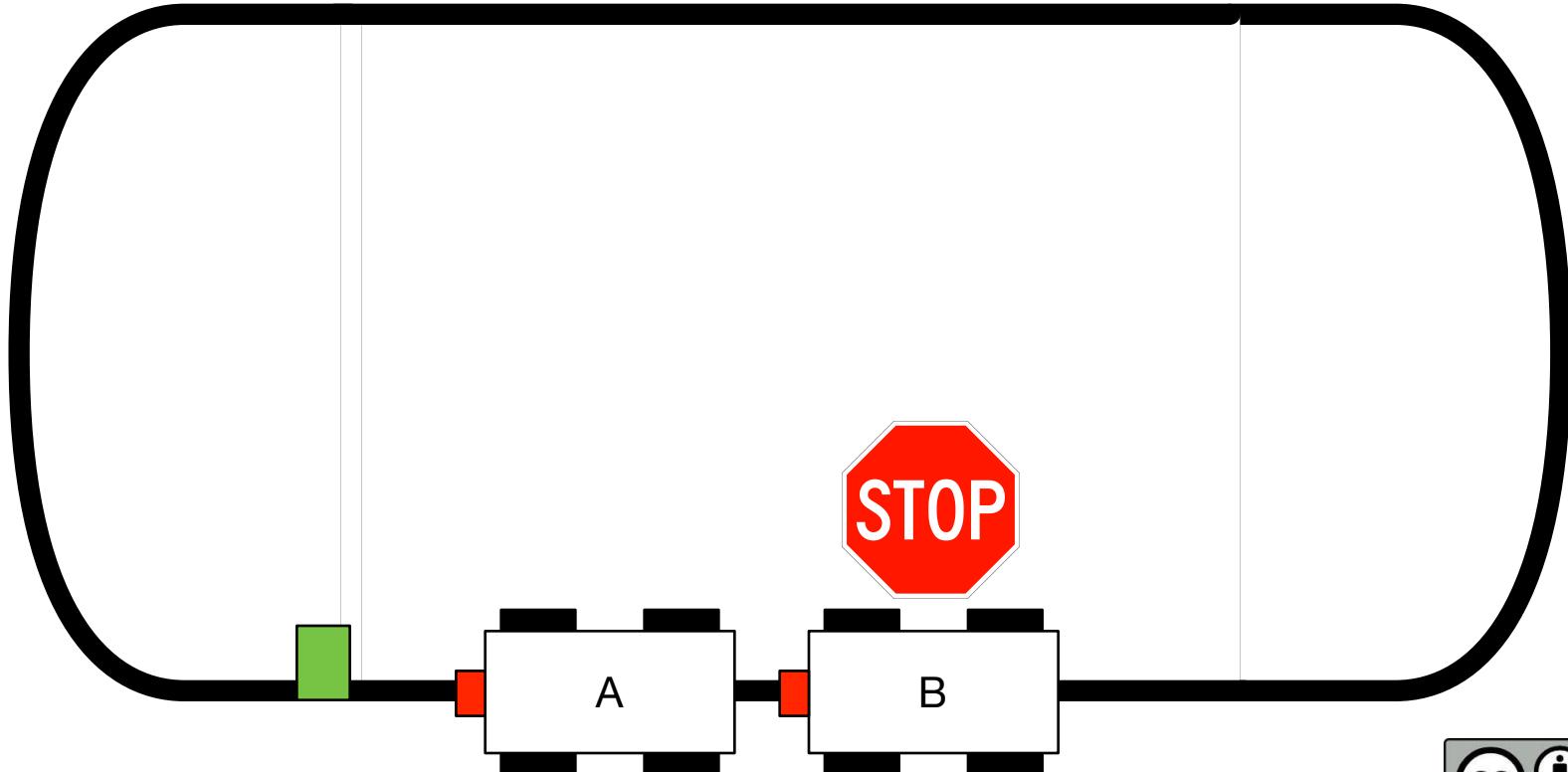
# Mission 4: Une solution (détail)



# Mission 5: Éviter les collisions



# Mission 5: Éviter les collisions



# Mission 5: Instructions

Crée un nouveau programme et nomme-le “mission05”.

Reprends le code de la “mission04”.

Lorsque le véhicule détecte un obstacle, il s’arrête.

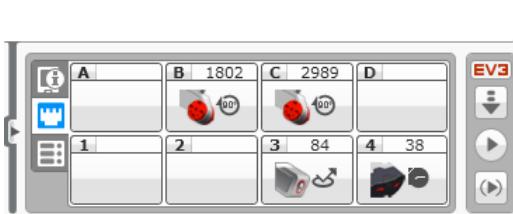
Lorsque l’obstacle disparaît, le véhicule reprend sa route.

Règle la valeur du seuil du détecteur infrarouge à 20 %.



# Mission 5: Seuil de proximité

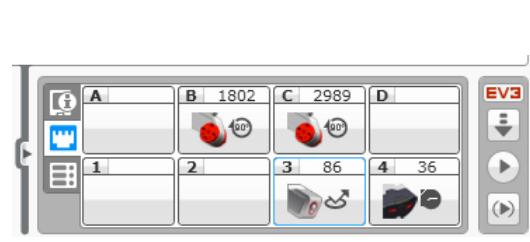
Définis le seuil du capteur de proximité:



Affichage des ports



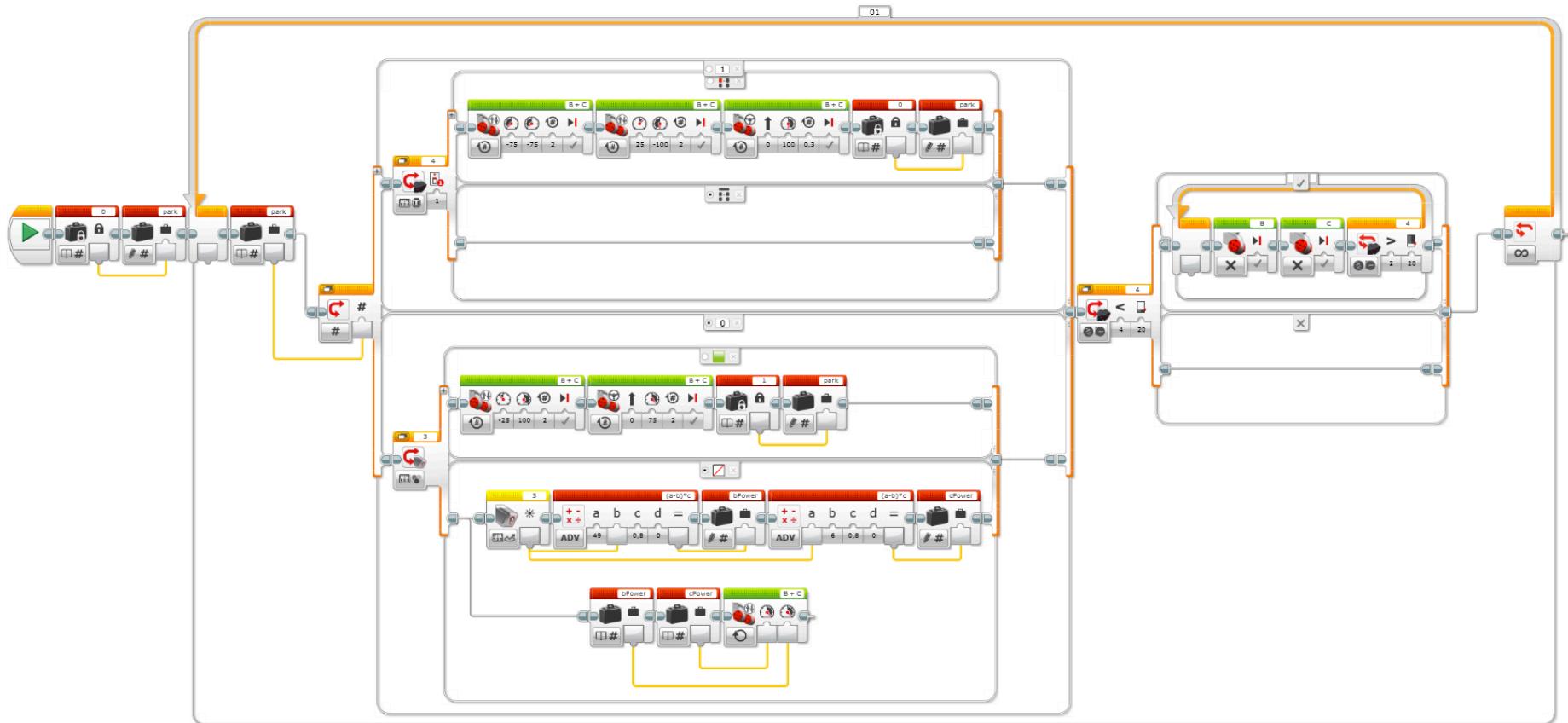
Clique sur le capteur infrarouge et choisis "Proximité"



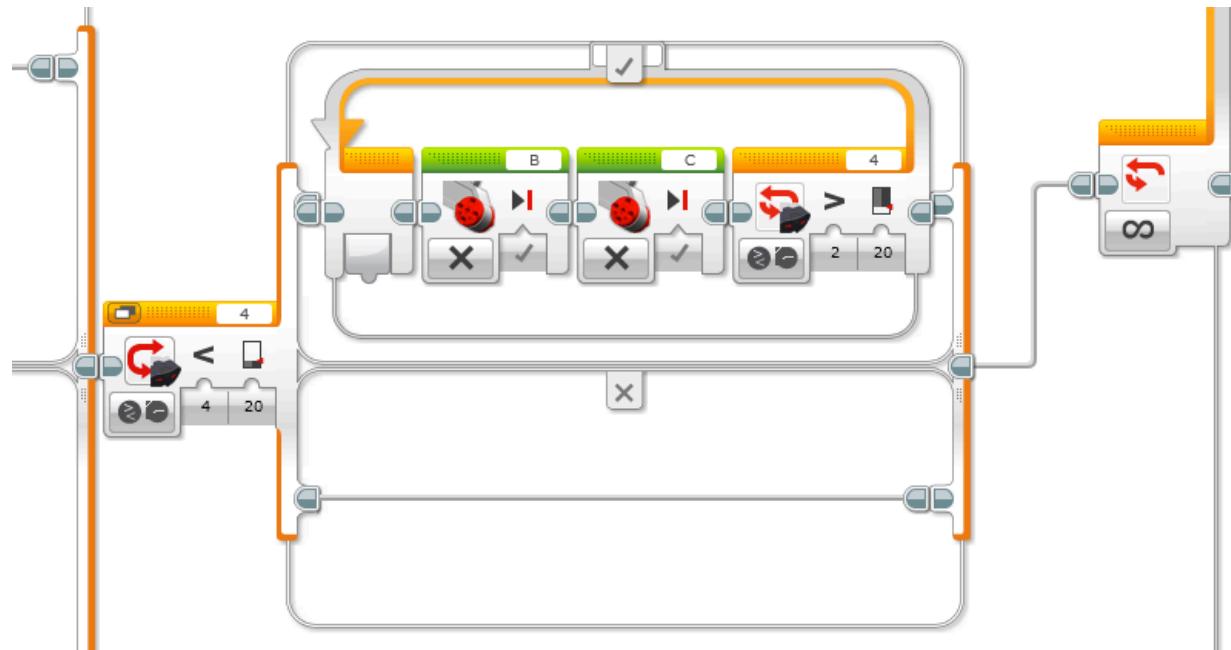
Mesure les valeurs en plaçant un obstacle devant le capteur de proximité



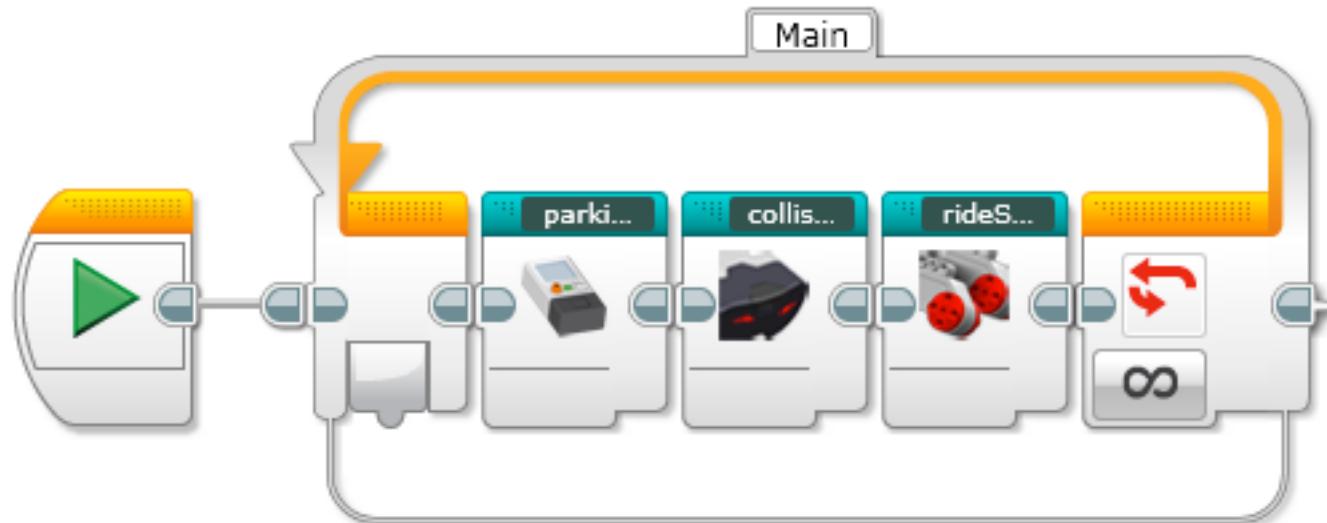
# Mission 5: Une solution



# Mission 5: Une solution (détail)



# Mission 6: Simplifie ton programme



# Mission 6: Instructions

Ton programme fonctionne, bravo !!!

Mais tu avoueras qu'il devient de plus en plus illisible :-(

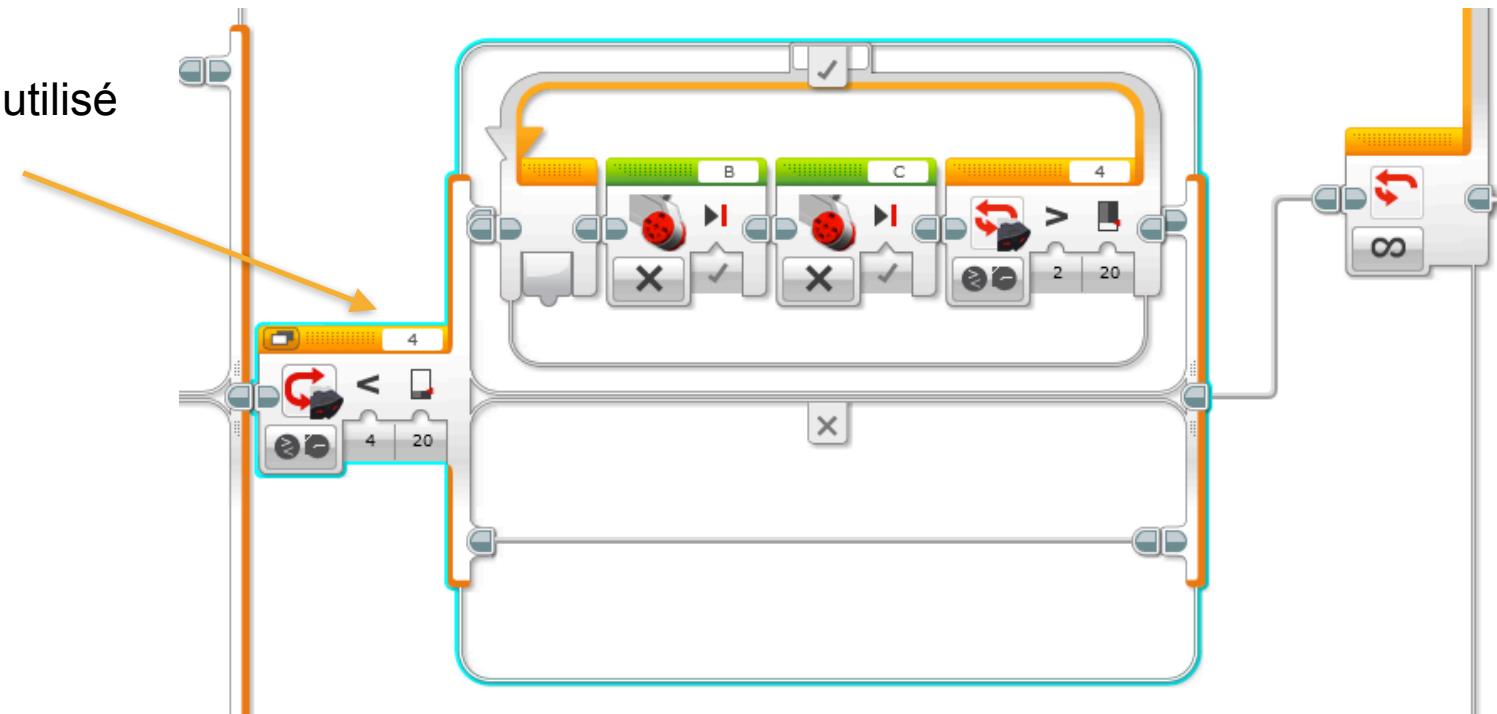
Tu peux améliorer cela en créant tes propres blocs.

Nous allons montrer comment créer un bloc pour stocker la logique mise en place dans la mission 5.



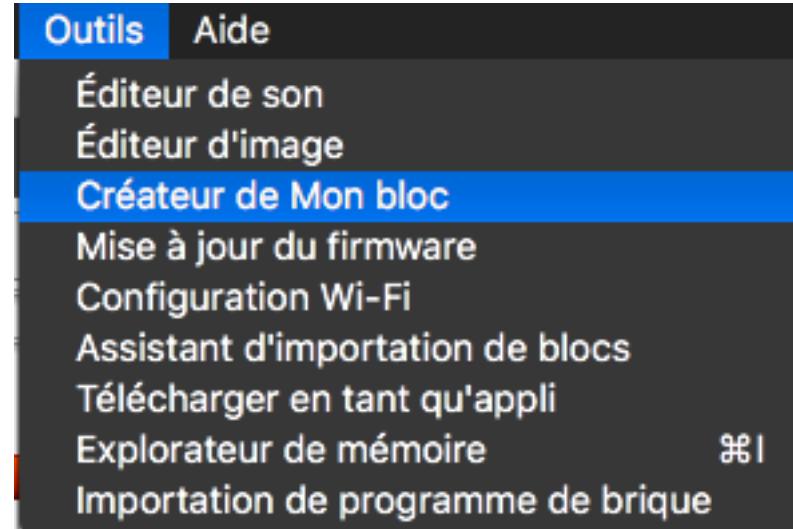
# Mission 6: Sélection du bloc

Sélectionne le bloc utilisé  
pour détecter les  
obstacles



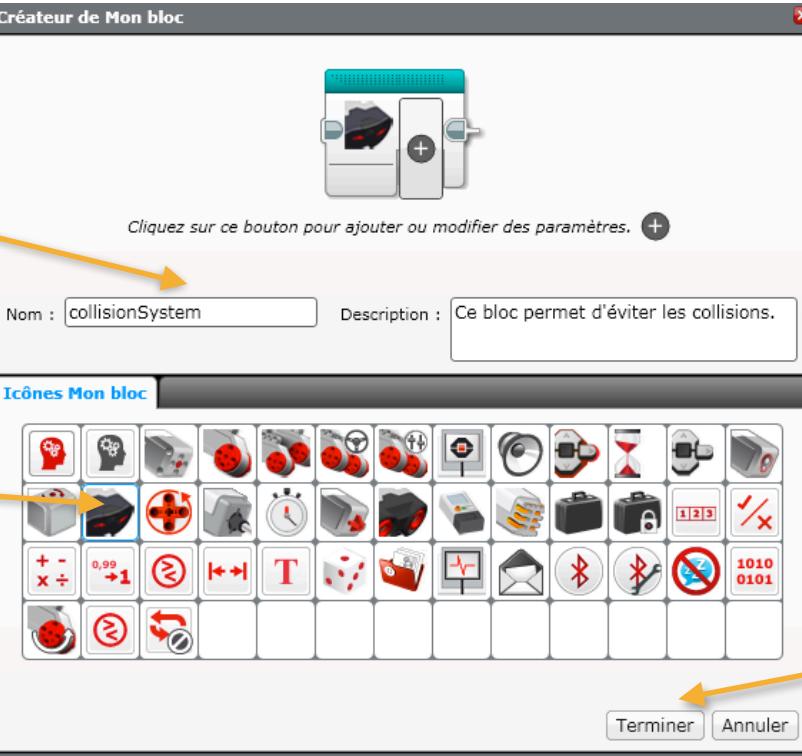
# Mission 6: Sélection du bloc

Dans le menu “Outils”,  
sélectionne l’option  
“Créateur de mon bloc”



# Mission 6: Sélection du bloc

Nomme ton bloc



Donne une description

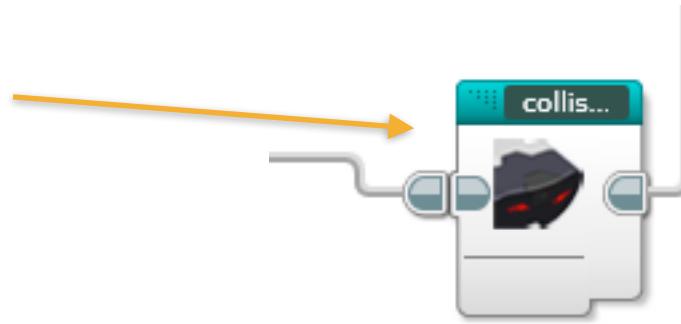
Choisis une icône

Sauver ton bloc



# Mission 6: Utiliser le bloc

Le code est remplacé par le bloc



Ton bloc est disponible dans  
la section “Mes blocs”



# Mission 7: Simplification

Crée un nouveau projet “smart-car” et nomme le programme “main”.

Reprends le code de la “**Mission 5**”.

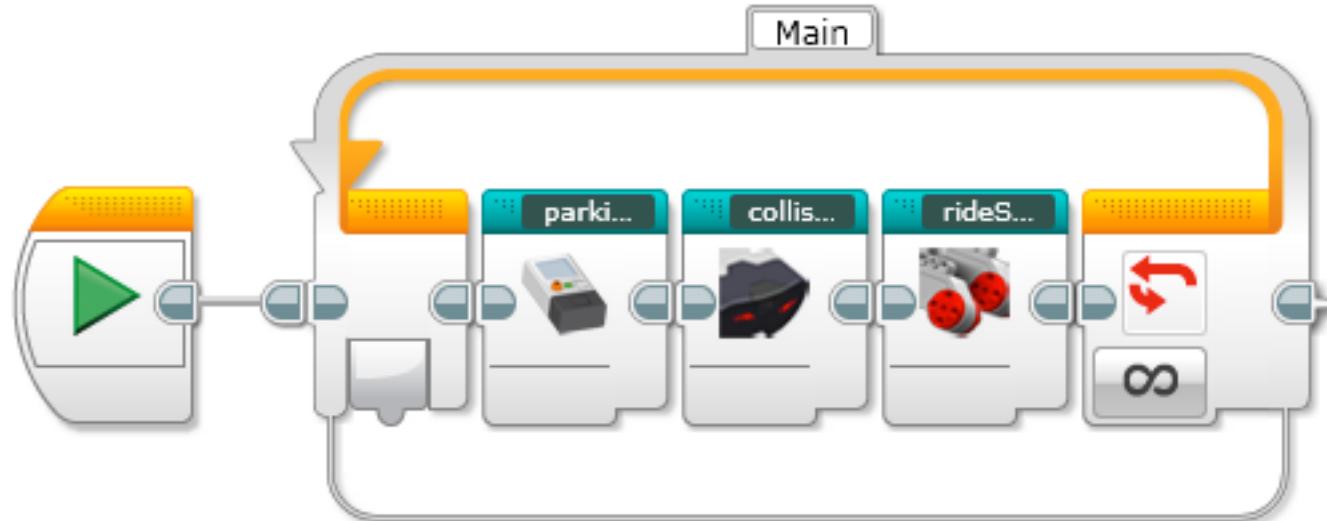
Utilise les principes de la “**Mission 6**” pour simplifier ton programme.

Dans ton programme principal (“main”), tu dois obtenir 3 blocs avec la logique suivante:

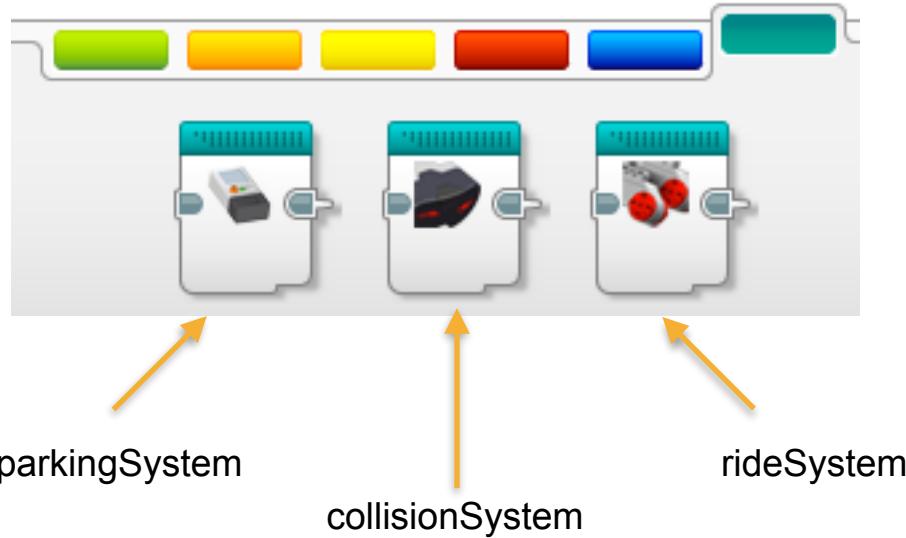
- **rideSystem**: faire rouler le véhicule sur la piste
- **collisionSystem**: détecter les obstacles
- **parkSystem**: détecte l’entrée du parking et le quitte lorsque le bouton 1 de la télécommande est appuyé



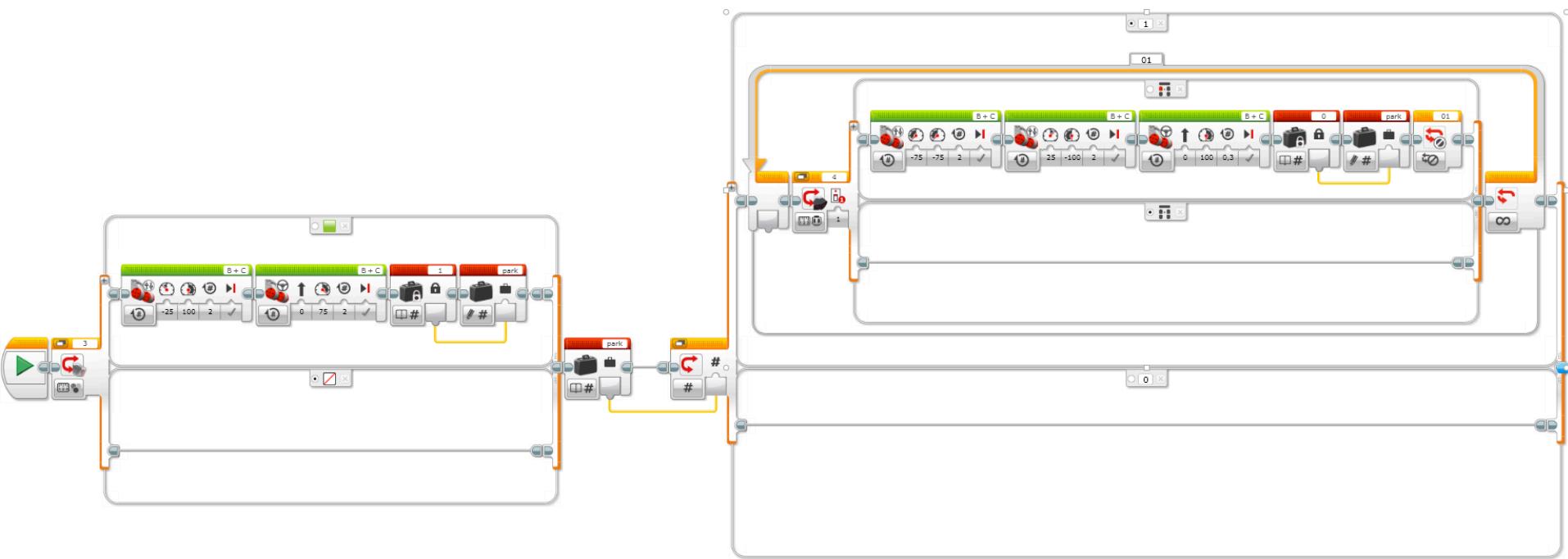
# Mission 7: Programme principal



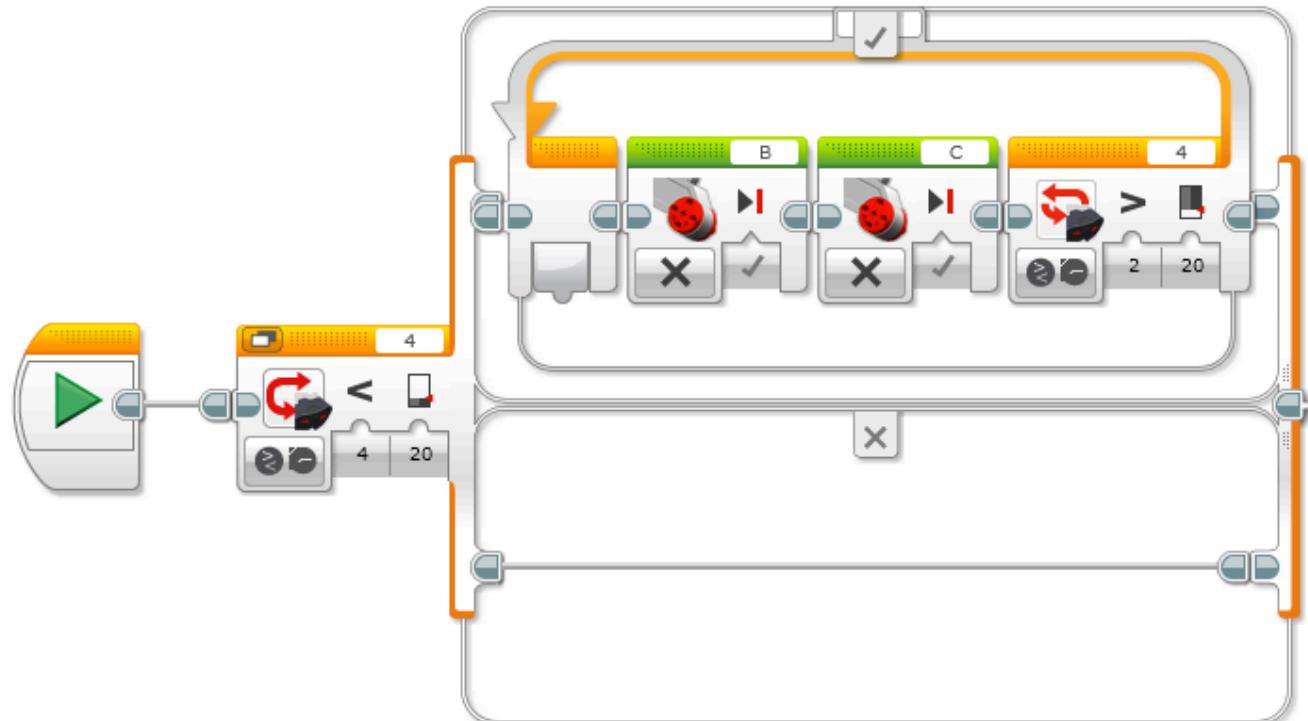
# Mission 7: Mes blocs



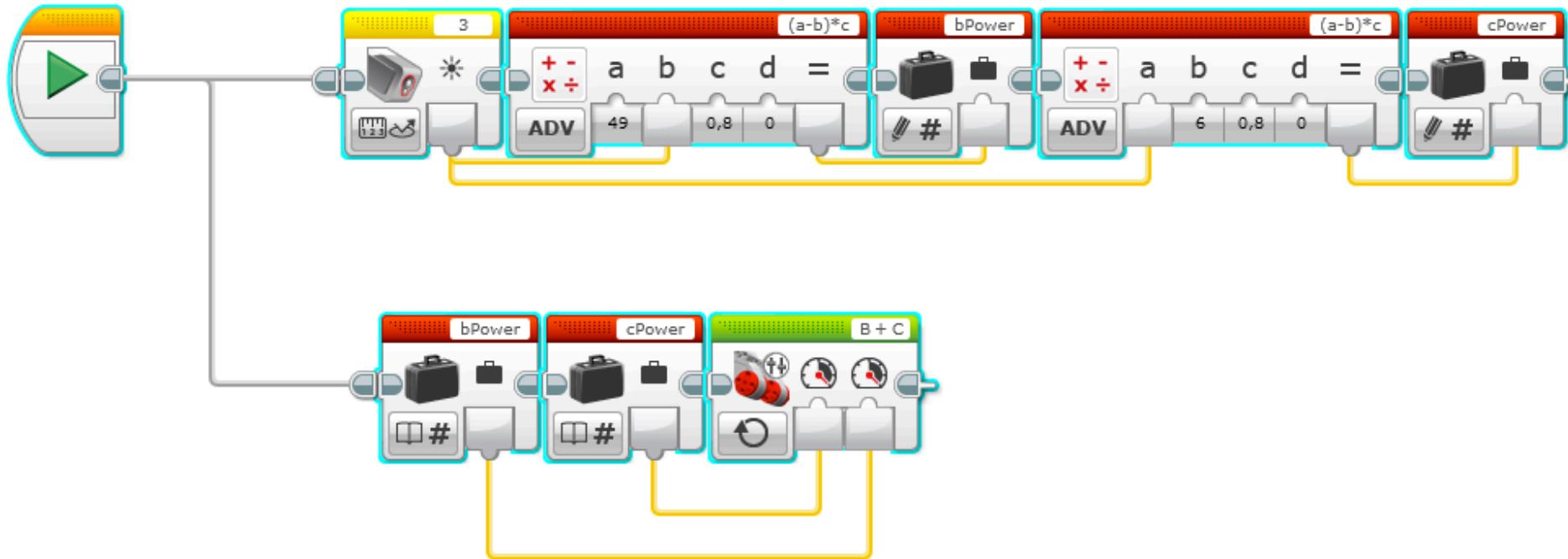
# Mission 7: parkSystem



# Mission 7: collisionSystem



# Mission 7: rideSystem



# Mission 7: Exécute le programme

Connecte le robot à l'ordinateur avec le câble USB

Télécharge le programme dans la brique du robot



Débranche le robot de l'ordinateur, pose-le sur le Test Pad et lance le programme (Icône Dossier > smart-car > main)



# Félicitations !!!

Tu as utilisé les fonctionnalités du Lego Mindstorms EV3 pour simuler le fonctionnement d'une voiture autonome.

N'hésite pas à enrichir ta voiture intelligente ou à créer tes propres projets.

La programmation avec Lego Mindstorms va t'aider à développer ta logique et ta créativité.



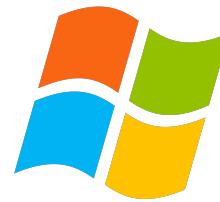
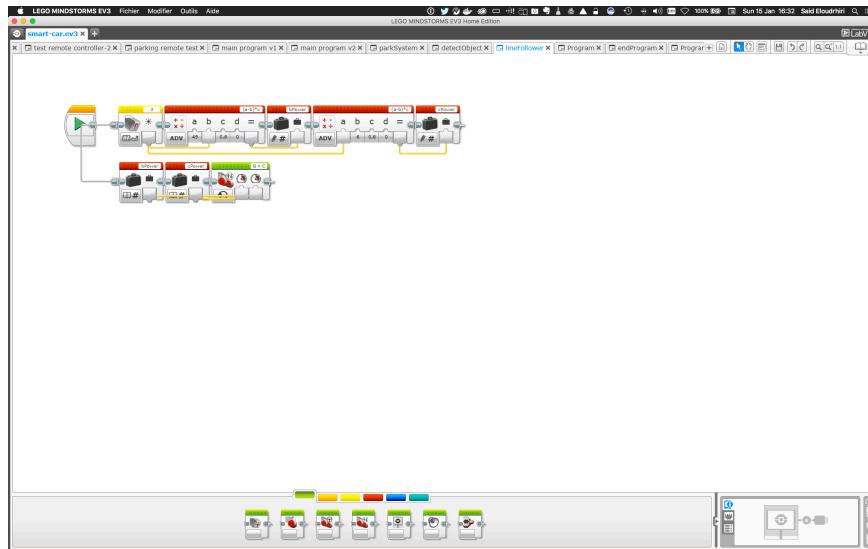
# Annexes: Aller plus loin ...

Dans les annexes suivantes, tu trouveras des informations additionnelles pour compléter tes connaissances qui te donneront l'envie d'aller encore plus loin.



# Annexe 1: Outils

## L'application Lego Mindstorms EV3



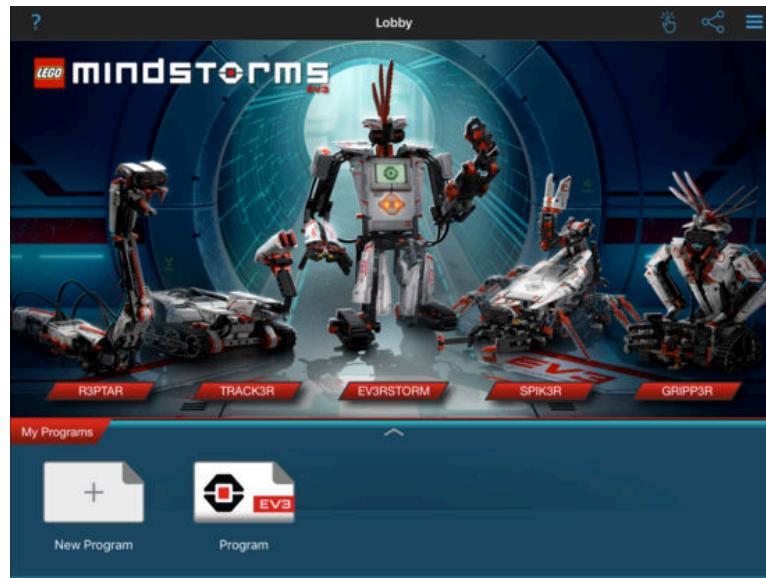
<https://goo.gl/NkQXJ6>



# Annexe 1: Outils

## L'application mobile Lego Mindstorms EV3

Attention  
De nombreux blocs  
d'instruction sont manquants  
dans la version mobile

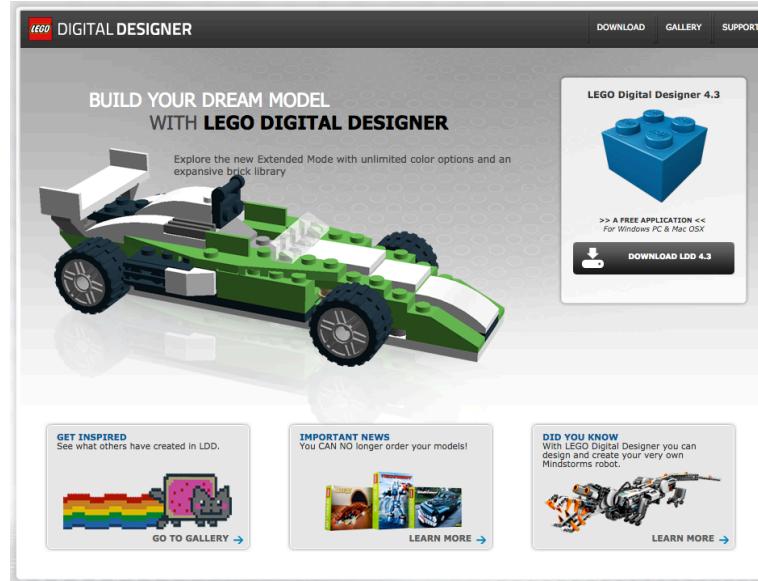


<https://goo.gl/IG3Z76>



# Annexe 1: Outils

## Lego Designer: construis ton lego

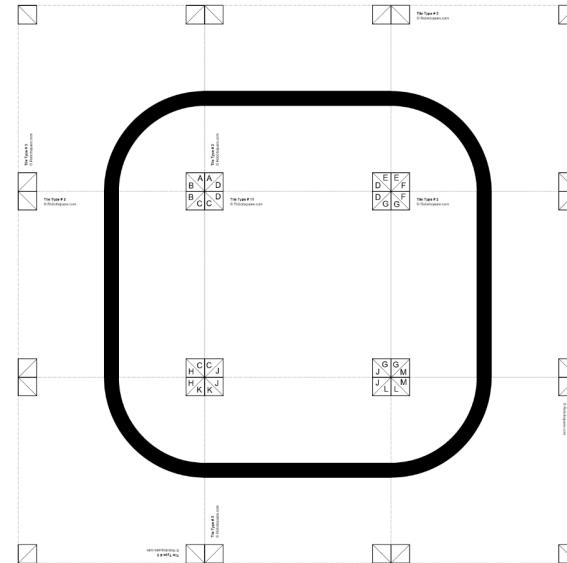


<http://ldd.lego.com>



# Annexe 1: Outils

## Imprime ton Test Pad



<https://goo.gl/rZyncZ>



# Annexe 2: Mindstorms

Les différents modèles de Mindstorms



**RCX**  
**1998**



**NXT**  
**2006**



**EV3**  
**2013**



# Annexe 2: Mindstorms

<b>CPU</b>	ARM 9 300 MHz	<b>Mobile</b>	Compatible avec iOS et Android
<b>Mémoire</b>	16 MB Flash 64 MB RAM Extensible via une carte micro-SD (max 32 GB)	<b>Ports</b>	4 ports d'entrée et 4 ports de sortie
<b>Système d'exploitation</b>	Linux	<b>Affichage</b>	LCD Matrix, monochrome 178 x 128 Pixels
<b>Communication</b>	Bluetooth 2.1	<b>Son</b>	Haut-parleur
<b>USB</b>	Interconnexion de plusieurs EV3. Support Wifi via dongle Espace de stockage		



# Annexe 2: Mindstorms

Mindstorms EV3 Home



+/- 400 euros

**MINDSTORMS**  
EV3  
Home Edition: #31313



© robotsquare.com  
Images © The LEGO Group

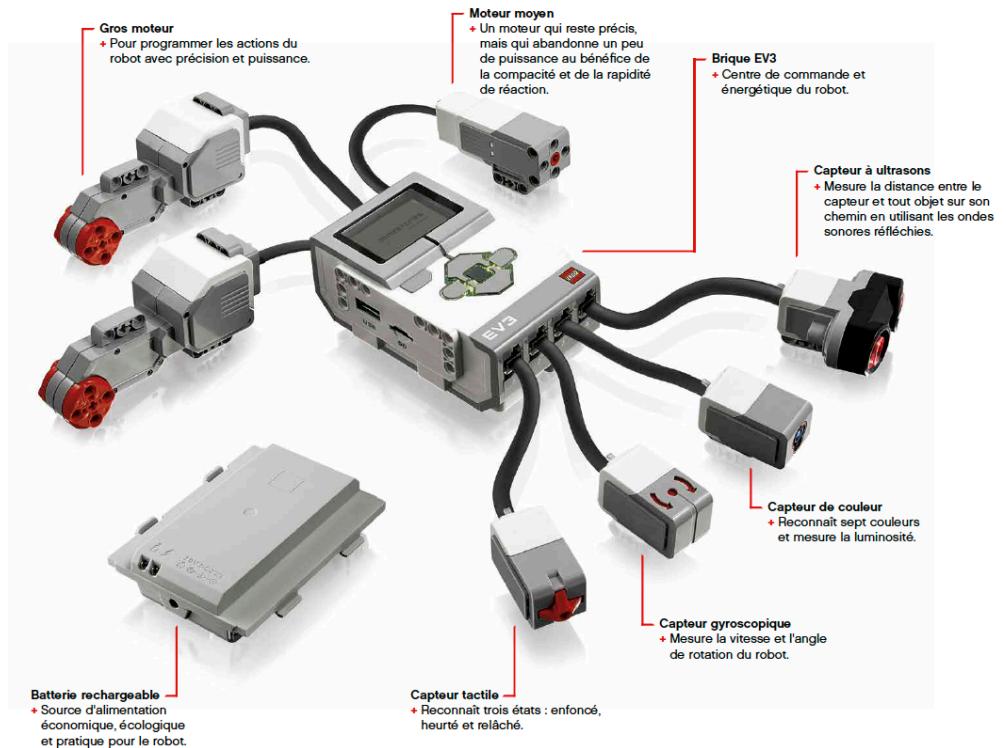


# Annexe 2: Mindstorms

## Mindstorms EV3 Education



+/- 450 euros



BY NC SA

# Annexe 2: Mindstorms

	Education	Home
Programming	EV3-G	EV3-G
Datalogging Software	Yes	No
Contents Editor	Yes	No

	Education	Home
Rechargeable Battery	Yes	No
Intelligent Brick	1	1
IR Beacon (Infrared Remote Control)	0	1
Infrared Sensor	0	1
Ultrasonic Sensor	1	0
Touch Sensor	2	1
Color Sensor	1	1
Gyro Sensor	1	0
Large Motor	2	2
Middle Motor	1	1

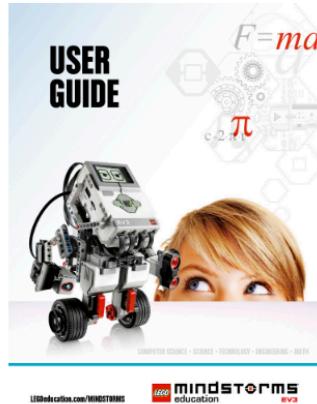


<https://goo.gl/uDNhZu>



# Annexe 2: Mindstorms

Desktop User Guide (English US):



App User Guide (English US):



Other languages (PDFs):

[Danish](#) - [German](#) - [Spanish](#) - [French](#) - [Italian](#) - [Japanese](#) - [Korean](#) - [Dutch](#) -  
[Norwegian](#) - [Portuguese](#) - [Russian](#) - [Swedish](#) - [Chinese](#) - [Arabic](#) - [English UK](#)

Other languages (PDFs):

[German](#) - [English UK](#) - [Spanish](#) - [French](#) - [Japanese](#) - [Dutch](#) - [Portuguese](#) -  
[Chinese](#)



<https://goo.gl/zvknu2>



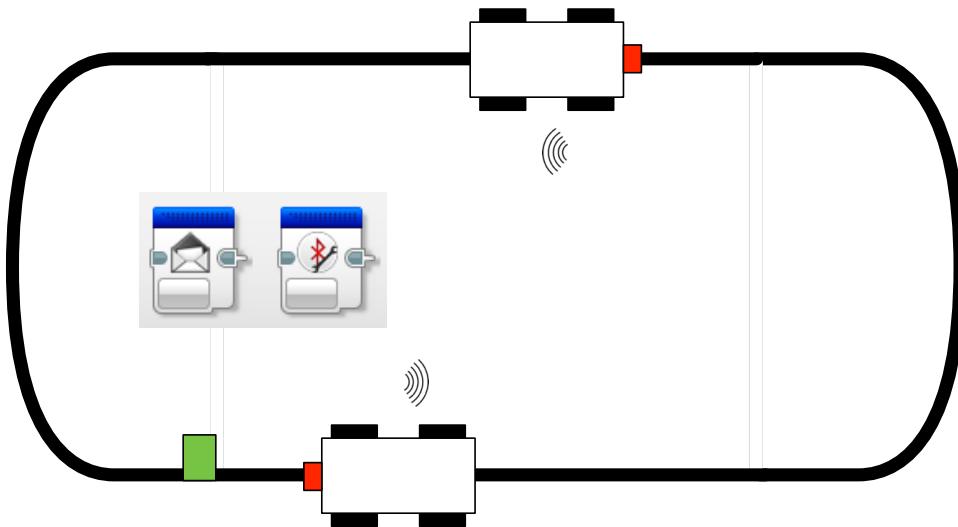
# Annexe 3: Communication entre EV3

Il est possible de faire communiquer deux mindstorms entre eux en utilisant bluetooth.

Des blocs existent pour:

1. connecter les mindstorms entre eux
2. s'envoyer des messages

Un exemple est donné ici:  
<https://www.youtube.com/watch?v=VkJtYK0r0c>



# Annexe 4: Importer des briques logicielles

Tu peux étendre la liste des capteurs avec de nouvelles briques logicielles que tu peux acheter dans le commerce où récupérer d'une ancienne version de Mindstorms (NXT).

Si tu as un Mindstorms NXT, sache que tu peux utiliser ses capteurs sous EV3.

Pour cela, ouvre la page suivante: <https://goo.gl/eDuyvb>

Dans la section “BLOCS LOGICIELS EV3”, tu trouveras les blocs dont certains sont compatibles avec NXT:

- Capteur sonore
- Capteur à ultrasons

Nous allons montrer les différentes étapes pour importer ces blocs dans le logiciel EV3.



# Annexe 4: Importer des briques logicielles

1. Récupère les blocs logiciels sur le site de Mindstorms à l'adresse: <https://goo.gl/eDuyvb>
2. Les blocs logiciels se trouvent sur la page (recherche "TÉLÉCHARGER UN BLOC LOGICIEL EV3 (PC/MAC)")
3. Les blocs logiciels des capteurs sonores et à ultrasons sont compatibles NXT
4. Lance le programme Lego Mindstorms EV3 sur ton PC ou Mac
5. Dans le menu "Outils", sélectionne "Importation de programme de brique"
6. Importe les blocs que tu as téléchargés et redémarre le programme Lego Mindstorms EV3
7. Les nouveaux blocs sont disponibles dans la section "Capteur"
8. Pour les tester, connecte-les à ton EV3 et analyse leur comportement dans la zone "Affichage des ports" (disponible en bas et à droite du logiciel EV3)



# Annexe 5: D'autres systèmes et langages

- LabView
- RobotC
- EV3dev
- LeJOS
- OpenRoberta
- EV3Basic
- Scratch
- Enchanting
- ROBOLAB

<http://www.legoengineering.com/alternative-programming-languages/>



# Annexe 6: Quelques liens

Site officiel **Lego Mindstorms EV3**: <https://goo.gl/W8DIKx>

**Emma** (Blue Angel Robotics) vous présente ses projets Lego Mindstorms sur sa chaîne YouTube: <https://goo.gl/tA24Gr> (en anglais)

**Builderdude35** explique ses différents projets Lego Mindstorms à l'aide de très nombreuses vidéos: <https://goo.gl/bB50jY> (en anglais)

Le site du **Wajug**: <http://www.wajug.be/>

Le site du **Devoxx4Kids**: <http://www.devoxx4kids.org/>

