

# Programmation de la voiture robotique fabriquée à la main pour qu'elle suive les instructions reçues par l'application créée



Présentation des 5 grandes idées en intelligence artificielle  
utilisant l'Internet des objets dans l'éducation STEM

T2.4 Conception de projets IoT et développement de ressources

# Projets IoT AI4STEM

## Projet: Voiture robotique IoT

### Copyright

© Copyright au AI4STEM Consortium  
2022-1-FR01-KA220-SCH-000085611  
Tous droits réservés.



Projets IoT AI4STEM Projet: Voiture robotique IoT © 2023 par [AI4STEM CONSORTIUM](#) est sous licence [Attribution - Utilisation non commerciale - Partage dans les mêmes conditions 4.0 International](#)

## Table des matières

1.1 Introduction .....	3
1.2 Ajout de l'extension Bluetooth .....	5
1.3 Création de la section "Configuration et connectivité.....	6
1.4 Création de la section "Actions entreprises après la réception d'un message Bluetooth". .....	7
1.5 Ajout d'indicateurs pour les commandes reçues.....	10
1.6 Remarques importantes concernant la connexion Bluetooth.....	11

## 1.1 Introduction

L'image suivante (Figure 1) présente l'ensemble du script qui doit être téléchargé sur le micro:bit afin de permettre à notre voiture robotique de suivre les instructions reçues par l'application. En ce qui concerne les besoins de ce projet, cette partie n'est pas obligatoirement enseignée dans les moindres détails. Les parties incluses dans ce script, telles que la manière dont les moteurs peuvent être activés et programmés pour se déplacer dans différentes directions (c'est-à-dire vers l'avant, vers l'arrière, etc...) sont expliquées dans le fichier "T2.4\_WarmUp\_programming\_activities\_for\_the\_robotic\_car.pdf". Par conséquent, et en fonction du niveau de vos élèves, vous pouvez soit expliquer brièvement les points clés du script, soit leur demander de télécharger le fichier .hex correspondant sur le micro:bit.

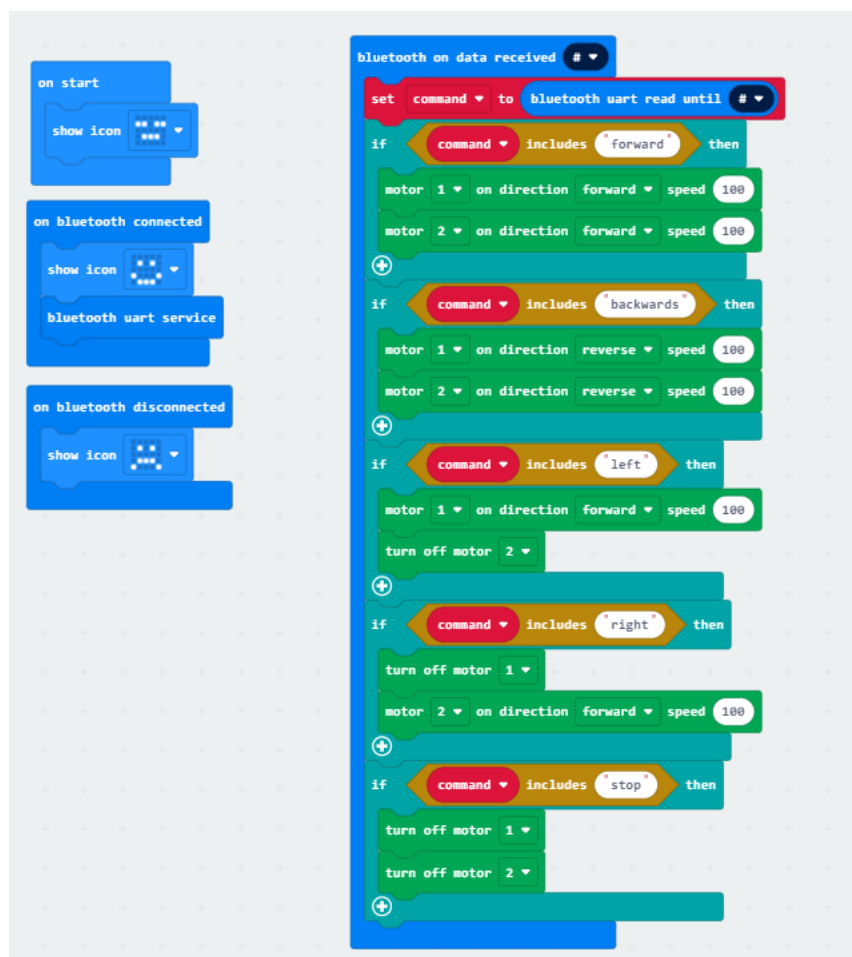


Figure 1: The entire script

Le script ci-dessus peut être grossièrement divisé en deux sections, à savoir la section "Configuration et connectivité" et la section "Actions entreprises après réception d'un message Bluetooth" (Figure 2, Figure 3).

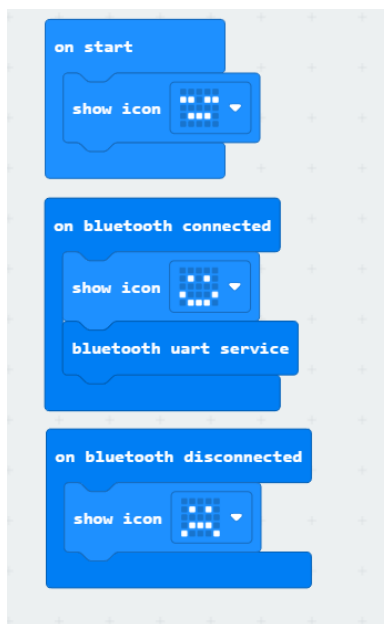


Figure 2: Configuration et connectivité

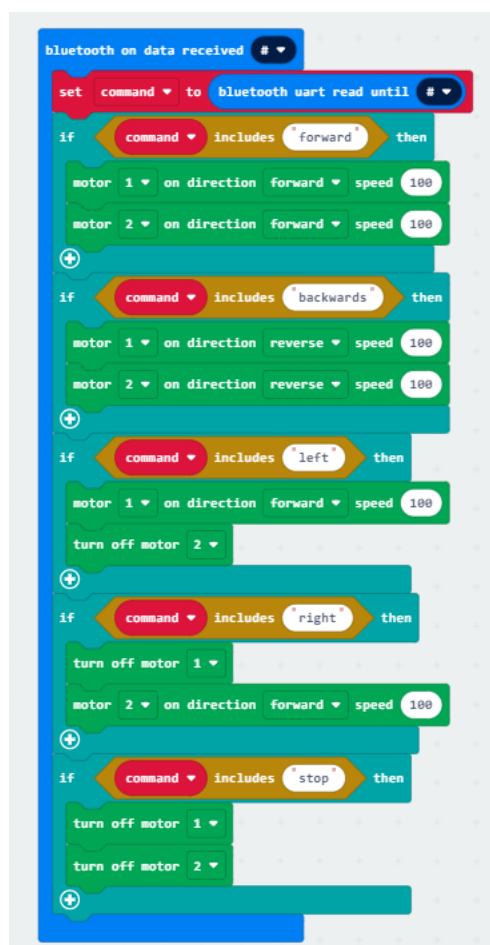


Figure 3: Actions entreprises après la réception d'un message Bluetooth

## 1.2 Ajout de l'extension Bluetooth

La première étape de la création du script consiste à ajouter l'extension Bluetooth de micro:bit à MakeCode, afin d'activer cette fonctionnalité à des fins de programmation. Cette procédure est la même que celle décrite à la page 5 du document "T2.4\_WarmUp\_programming\_activities\_for\_the\_robotic\_car.pdf", à la différence que nous recherchons "Bluetooth" dans la barre de recherche des extensions (Figure 4).

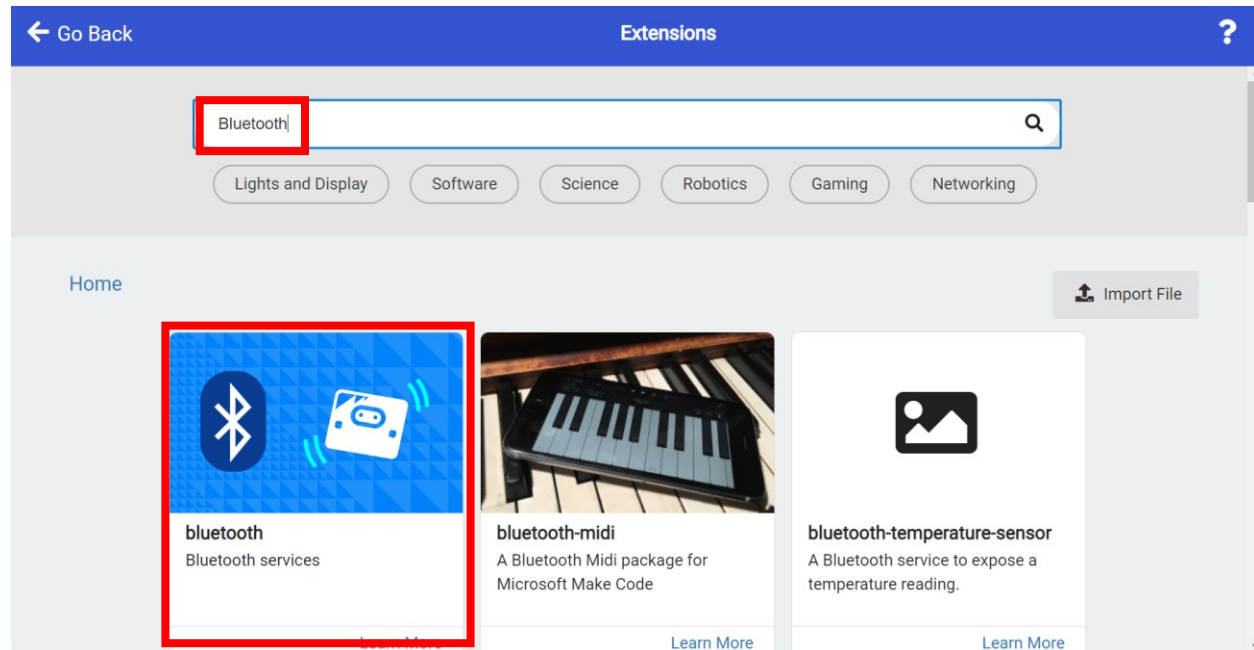


Figure 4: Finding the Bluetooth extension

Après avoir ajouté l'extension, un nouveau menu avec un bloc de commandes peut être trouvé dans l'environnement MakeCode (Figure 5).

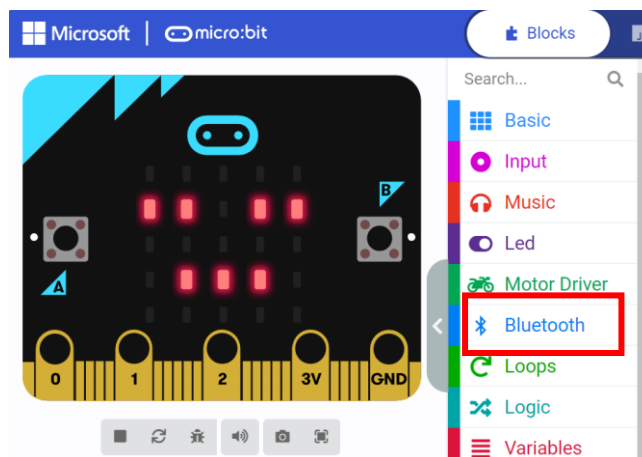


Figure 5: The Bluetooth menu.

**Note :** Avant d'ajouter le nouveau menu, un/e avertissement/notification s'affiche (Figure 6), informant que l'extension radio est incompatible avec Bluetooth et doit être supprimée. Cliquez sur le bouton **"Remove extension(s) and add bluetooth"** pour confirmer votre choix.

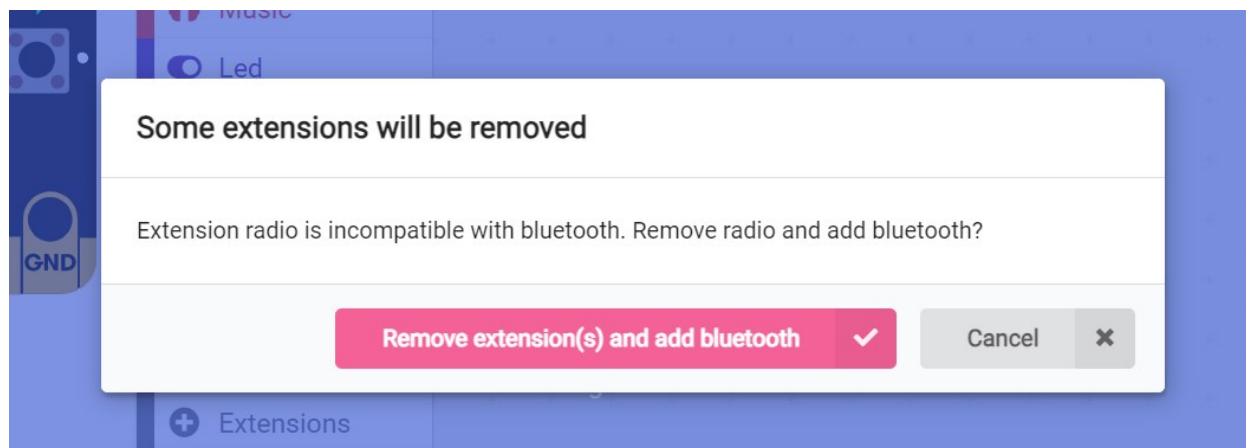


Figure 6: Avertissement concernant les extensions qui seront supprimées

### 1.3 Création de la section "Configuration et connectivité"

Cette section (Figure 7) se compose de trois blocs principaux : un bloc **de base** appelé **"on start"** et deux blocs **Bluetooth** appelés **"on bluetooth connected"** et **"on bluetooth disconnected"**. Le premier spécifie ce qui se passe lorsque le Micro:bit s'allume, le second ce qui se passe après l'établissement de la connexion Bluetooth et le troisième les actions entreprises lorsque la connexion Bluetooth est terminée.

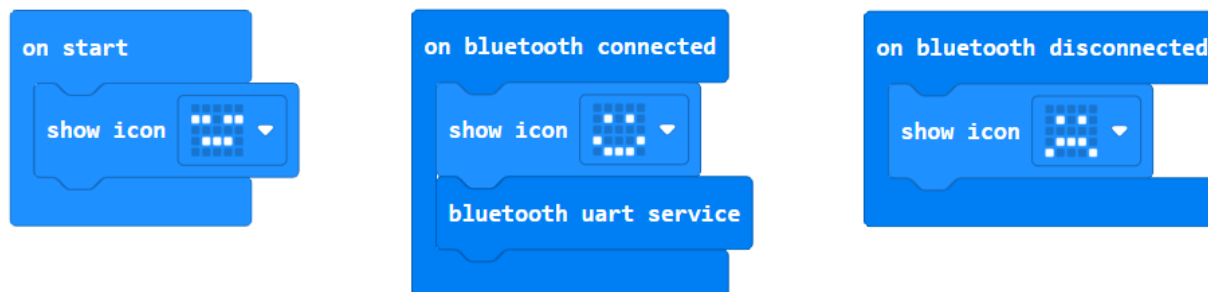


Figure 7 : La section "configuration et connectivité"

Dans les détails :

- **On start** (lorsque le micro:bit s'allume), **affiche** une **icône** de sommeil. Cette icône sera affichée sur l'écran LED du micro:bit et est utilisée pour vérifier que le script a été téléchargé avec succès sur la carte. Vous pouvez choisir l'icône de sommeil ou n'importe quelle autre icône dans le menu flottant.
- **On bluetooth connected** (lorsqu'une connexion Bluetooth a été établie), **affiche** une **icône** "visage heureux" et démarre le **service bluetooth uart**. Le service bluetooth uart permet au micro:bit d'envoyer et de recevoir des messages via Bluetooth. En outre, l'icône "visage heureux" est utilisée comme indicateur d'une connexion Bluetooth **réussie**.
- **On bluetooth disconnected** (lorsque la connexion Bluetooth a été interrompue), l'**icône** "visage triste" **s'affiche**. L'icône "visage triste" est utilisée comme indicateur d'une connexion Bluetooth **interrompue**.

Après avoir effectué toutes les étapes susmentionnées, votre micro:bit vous informera de l'état de la connectivité de tout service Bluetooth et sera en mesure de recevoir des messages d'autres appareils Bluetooth.

## 1.4 Création de la section "Actions entreprises après la réception d'un message Bluetooth".

Cette section décrit ce qui se passe lorsqu'un message est reçu via Bluetooth. Pour que le micro:bit puisse reconnaître un message, celui-ci doit être placé entre deux symboles hashtag (#) (par exemple #message#). Le message est ensuite communiqué et, en fonction de son contenu, la commande correspondante est exécutée. Par exemple, et comme le montre le tableau suivant, si le message reçu contient le mot forward, placé entre 2 hashtags (c'est-à-dire #forward#), la voiture robotisée recevra l'instruction de se déplacer vers l'avant.



Message Reçu	Message à proprement parler	Si le message contient le mot :	Exécutez alors l'action suivante :
#message#	message	forward	faire avancer la voiture
		backwards	faire reculer la voiture
		left	déplacer la voiture vers la gauche
		right	déplacer la voiture vers la droite
		stop	Arrêter la voiture
		<b>aucune de ces réponses</b>	-

La figure 8 présente la partie du script qui permet la communication susmentionnée.

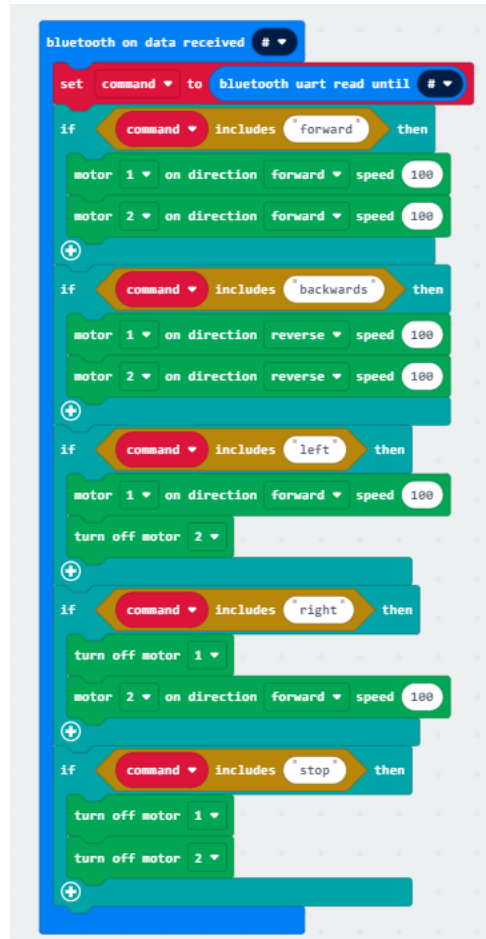


Figure 7

Dans les détails :

On **bluetooth data received** (lors de la réception d'un message Bluetooth), **lire** jusqu'à **bluetooth uart** jusqu'à voir # et alors le **définir** comme une variable appelée **command**. Procédez ensuite aux vérifications suivantes :

- Si **command** **inclut** le mot "forward", **alors démarrer** les deux **moteurs** et aller dans le sens de **la marche avant** avec une **vitesse de 100**
- Si **command** **inclut** le mot "backwards", **alors démarrer** les deux **moteurs** et aller dans le sens de **la marche arrière** avec une **vitesse de 100**
- Si **command** **inclut** le mot "left", **alors démarrer** le **1er moteur** pour aller dans le sens de **la marche avant** avec une **vitesse de 100**, et **arrêter** le **2nd moteur**

- Si **command** **inclut** le mot "right", **alors** **démarrer** le **2nd moteur** pour aller dans le sens de **la marche avant** avec une **vitesse de 100**, et **arrêter** le **1er moteur**
- Si **command** **inclut** le mot "stop", **alors** **arrêter** les deux **moteurs**.

**Note:** Pour créer la variable "command", cliquez sur "Variables", puis sur le bouton "Créer une variable" (Figure 9). Dans le menu contextuel, tapez le nom "command" dans la section "nouveau nom de variable", et appuyez sur Ok. Le bloc de commandes "**set command to ...**" apparaît dans le sous-menu Variables.

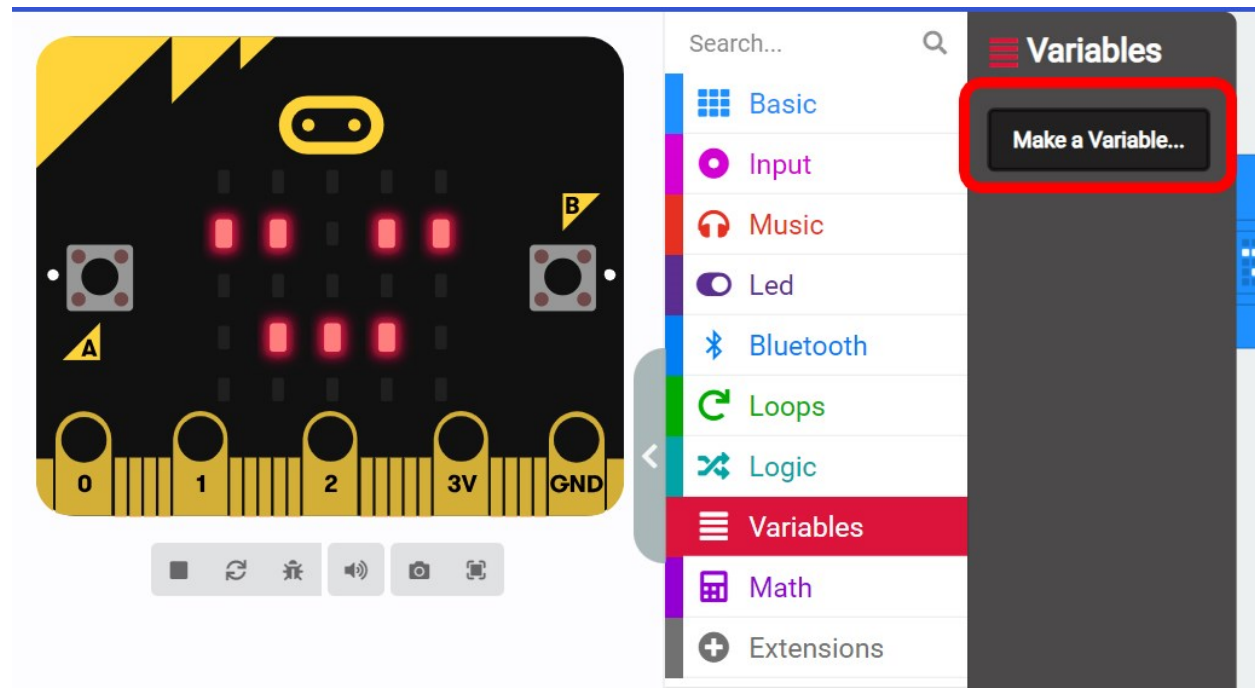


Figure 8: Creating a Variable

## 1.5 Ajout d'indicateurs pour les commandes reçues

Cette étape n'est pas obligatoire, mais elle peut aider les élèves à savoir si le script téléchargé fonctionne correctement ou non.

Dans l'instruction "If...then", ajoutez une commande "show leds" et cochez les cases correspondantes, afin de créer une flèche indiquant la direction que la voiture robotisée doit suivre, si la commande correspondante est reçue (par exemple, créez une flèche vers le haut si la commande reçue est "forward").

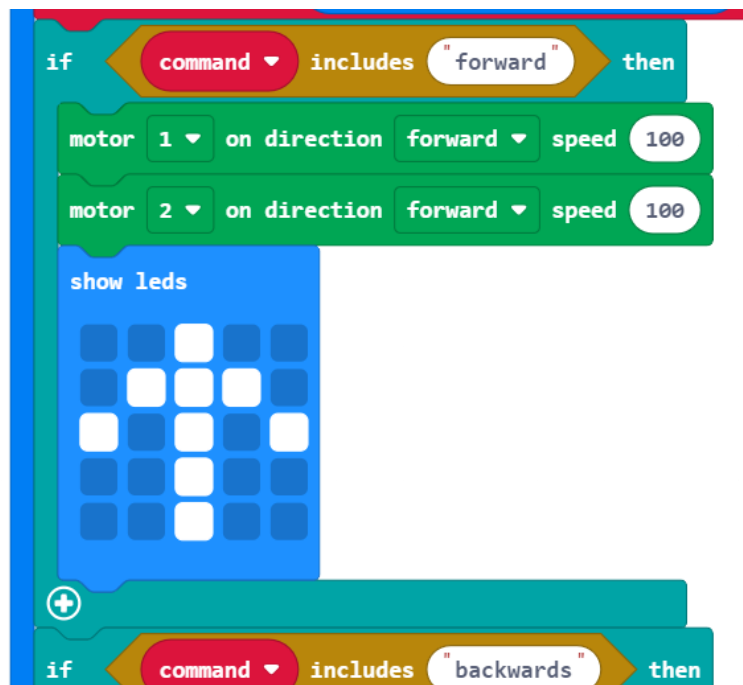


Figure 9: Ajout d'une commande "show leds" représentant une flèche "avant", pour vérifier que le micro:bit a reçu la commande correcte correspondant au mouvement vers l'avant (de même pour les autres commandes).

L'ajout de la commande "show leds" dans MakeCode pour afficher une icône spécifique lorsqu'un message Bluetooth est reçu sur le micro:bit est une amélioration précieuse du projet. Ce bloc sert d'indicateur visuel, permettant de confirmer plus facilement que le micro:bit a bien reçu et interprété la commande Bluetooth. Les signaux visuels sont particulièrement utiles pour déboguer et assurer le bon fonctionnement du script en temps réel.

**Note:** Le bloc "show leds" est situé dans la catégorie Basic.

**Conseil :** Personnalisez le motif des LED pour représenter l'icône ou le symbole de votre choix en allumant ou en éteignant les différentes LED de la grille.

En mettant en œuvre cette étape, non seulement l'expérience de l'utilisateur est améliorée, mais le processus de dépannage est également simplifié car un retour d'information immédiat sur la réception de la commande est fourni. En outre, le projet devient plus convivial et plus attrayant sur le plan visuel.

## 1.6 Remarques importantes concernant la connexion Bluetooth

**A)** Avant de télécharger le script final sur micro:bit, assurez-vous que l'option "No Pairing Required : Anyone can connect via Bluetooth" est sélectionnée dans le menu Paramètres du projet. Pour vérifier ce paramètre, cliquez sur le symbole de roue dentée et, dans le menu flottant, cliquez sur "Project settings" (Figure 11). Activez ensuite l'option "No Pairing Required : Anyone can connect via Bluetooth".

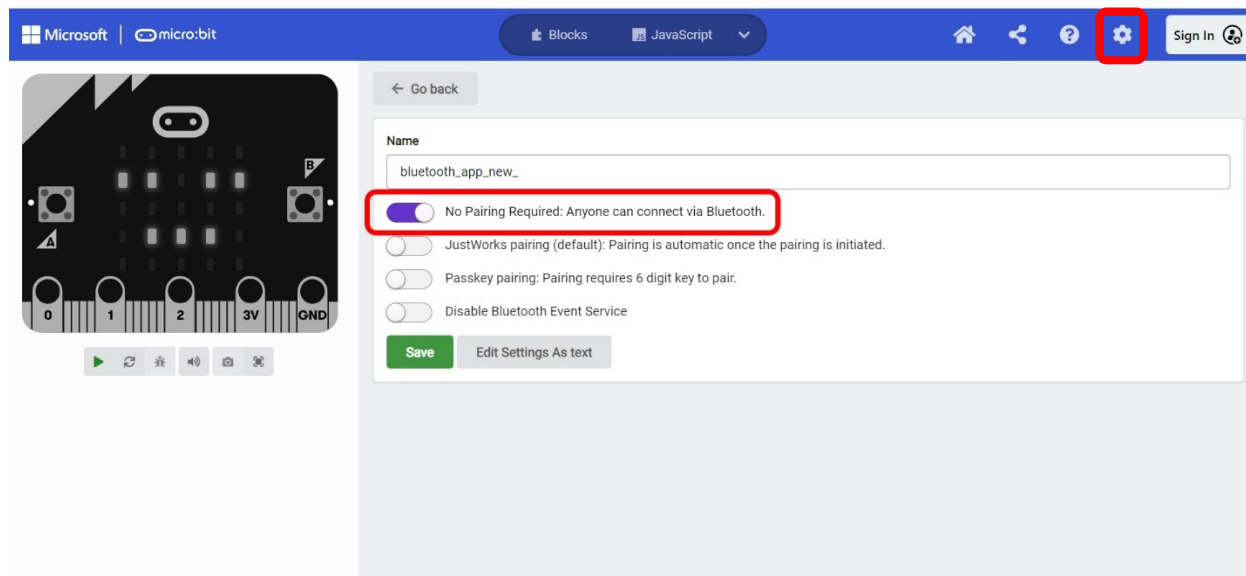


Figure 10: Checking the Project settings menu

**B)** Avant d'essayer de connecter le micro:bit à l'application créée, assurez-vous que votre appareil intelligent reconnaît effectivement le micro:bit. Vous pouvez vérifier cela en ouvrant le menu Bluetooth sur votre appareil intelligent, et en vérifiant si la carte micro:bit apparaît ou non dans les connexions disponibles.

Si vous ne trouvez pas votre micro:bit dans les connexions disponibles, essayez de renommer votre projet en ajoutant le mot "microbit" au début du nom (par exemple "microbit-robotic\_car"), et téléchargez à nouveau le projet sur votre micro:bit.