



Cofinancé par  
l'Union européenne

Financé par l'Union européenne. Les points de vue et avis exprimés n'engagent toutefois que leur(s) auteur(s) et ne reflètent pas nécessairement ceux de l'Union européenne ou de l'Agence exécutive européenne pour l'éducation et la culture (EACEA). Ni l'Union européenne ni l'EACEA ne sauraient en être tenues pour responsables..

# Voiture robotique IoT



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

06.10.2023 | EDUMOTIVA  
NUMÉRO DE PROJET: 2022-1-FR01-KA220-SCH-000085611

# 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.Introduction au projet.....	4
1.1 La portée du projet .....	4
1.2 Les groupes cibles .....	5
1.3 Objet de ce document .....	5
2. Glossaire de l'Unité .....	5
3. Présentation de la « voiture robotique DIY qui peut être contrôlée et pilotée à l'aide de commandes vocales ».....	5
3.1 Description .....	5
3.2 Objectifs et résultats d'apprentissage .....	7
3.3 Durée estimée de l'Unité .....	8
3.4 Activité 1 – Présentation de la grande idée de la perception grâce à l'IoT : .....	9
3.4.1 Description .....	9
3.4.2 Matériel.....	9
3.4.3 Configuration .....	9
3.4.4 Expérience 1.....	17
3.5 Activité 2 : Introduire l'idée de représentation et de raisonnement.....	19
3.5.1 Description .....	19
3.5.2 Création d'un arbre de décision.....	19
3.5.3 Conception et programmation de l'application.....	20
3.5.4 Expérience 2.....	29
3.6 Activité 3 : Introduire l'idée d'apprendre en entraînant un modèle de reconnaissance des commandes vocales.....	31
3.6.1 Description .....	31
3.6.2 Utiliser Personal Audio Classifier pour entraîner un.....	31
3.6.3 Expérience 3.....	36
3.7 Activité 4 : Présentation de l'idée d'interaction naturelle en intégrant un modèle entraîné dans une application d'IA .....	37
3.7.1 Description .....	37
3.7.2 Intégration du modèle entraîné à l'application d'IA.....	37
3.7.3 Expérience 4.....	48
3.8 Activité 5 : Introduire l'idée d'impact sociétal .....	49
3.8.1 Description .....	49
3.9 Matériel et ressources .....	50

3.10 Le matériel pour la voiture robotique .....	50
--	----

## 1.Introduction au projet

Le présent projet se concentre sur la création d'une voiture robotique DIY pouvant être contrôlée par des commandes vocales, tout en étant capable de collecter un certain nombre de données (telles que la température, le niveau de luminosité, la distance, l'accélération, etc.) pouvant conduire à certaines décisions pour optimiser sa performance. Cette intervention d'apprentissage basée sur des projets aidera les enseignants et les étudiants à s'initier aux domaines de l'IA et de l'IoT, à travers le prisme des 5 grandes idées (à savoir la perception, la représentation et le raisonnement, l'apprentissage, l'interaction naturelle et l'impact sociétal), ainsi que ainsi qu'à la lumière de la robotique en mettant en œuvre une série de pratiques et d'activités pratiques et informatiques. En ce qui concerne l'IA, ils seront initiés à la reconnaissance vocale et à la manière dont ce service peut être utilisé pour naviguer dans un artefact robotique. À cette fin, ils seront initiés aux processus de conception et de programmation d'une application basée sur l'IA à l'aide du logiciel MIT App Inventor, ainsi qu'à la formation d'un modèle pour classer les informations entrantes à l'aide de l'environnement Personal Audio Classifier. Concernant la robotique, ils apprendront à construire une voiture robotique en utilisant le microcontrôleur BBC micro:bit et plusieurs composants électroniques compatibles, ainsi qu'à programmer cet artefact robotique à l'aide du logiciel Makecode basé sur des blocs. Le projet sera divisé en 5 activités. Chacune de ces activités s'articulera autour de l'une des 5 grandes idées. Au cours de ces activités, les étudiants seront invités à se concentrer sur différentes parties du processus de mise en œuvre et à s'engager dans différents aspects de l'IA et de l'IoT. Les activités comprendront des lignes directrices pour les enseignants et plusieurs suggestions de tâches pour les étudiants afin d'assurer une introduction et une mise en œuvre fluides de tous les concepts susmentionnés et de leurs aspects inhérents, conduisant finalement à l'acquisition et au développement de plusieurs compétences du 21e siècle telles que la créativité, la pensée critique, la résolution de problèmes. résolution et collaboration.

### 1.1 La portée du projet

A travers ce projet, les étudiants seront initiés au domaine de l'IA et des 5 Grandes Idées, ainsi qu'au domaine de l'IoT, à la lumière de la robotique. Notamment, à travers les 5 activités et les tâches associées, qui tournent autour de la construction et de la programmation de la voiture robotisée, ainsi que de la conception et de la programmation de l'application IA, les étudiants comprendront mieux les 5 Grandes Idées et approfondiront certains des mécanismes fondamentaux de l'IA et de l'IoT. Plus précisément, dans la 1ère activité, les étudiants seront initiés à l'IoT en apprenant à programmer leur voiture robotique pour collecter et surveiller un certain nombre de données environnementales via un service de plateforme d'analyse IoT (c'est-à-dire ThingSpeak). À travers les tâches qui y sont contenues, ils seront initiés à la notion de perception. Dans la 2e activité, les élèves seront initiés à l'idée de représentation et de raisonnement en apprenant comment la voiture robotique peut « penser » et comment les données peuvent être représentées de plusieurs manières. À cette fin, ils apprendront à créer des arbres de décision et des organigrammes, et à utiliser ces informations pour créer une application qui permettra de naviguer dans la voiture robotisée à l'aide de commandes vocales. Par conséquent, ils apprendront à utiliser le service Speech Recognition AI et évalueront les résultats de sa mise en œuvre. Dans la 3ème activité, les étudiants seront initiés à l'idée d'apprentissage et, en utilisant un outil d'apprentissage automatique (ML) (le Personal Audio Classifier), ils se familiariseront avec les méthodes que la voiture robotique peut apprendre à partir des données et via le ML. . Dans la 4ème Activité, ils seront initiés à l'idée d'Interaction Naturelle, en intégrant le modèle formé (produit dans la 3ème Activité) dans l'application créée et en évaluant les résultats à la lumière d'une mise en œuvre physique, prenant ainsi

conscience des limites des systèmes d'IA en termes d'interaction naturelle. Enfin, dans la 5ème activité, ils seront initiés à l'idée d'impact sociétal, en réfléchissant à l'ensemble de leur expérience et en prenant conscience des avantages, des inconvénients et des risques liés à l'utilisation de l'IA et de l'IoT dans notre vie quotidienne. Le but ultime de ce projet est d'accroître la confiance des étudiants dans tous les concepts et aspects susmentionnés et de créer des expériences d'apprentissage significatives qui les aideront à développer un certain nombre de compétences du 21e siècle telles que la créativité, la pensée critique, la résolution de problèmes et la collaboration.

## 1.2 Les groupes cibles

Le projet s'adresse aux étudiants âgés de 12 à 16 ans. Les étudiants doivent avoir une certaine expérience des environnements de programmation basés sur des blocs.

## 1.3 Objet de ce document

L'objectif de ce document est de fournir aux enseignants des idées concrètes et des activités d'apprentissage sur la manière dont les concepts de l'IA et de l'IoT peuvent être introduits et enseignés de manière significative aux étudiants, à travers le prisme de la robotique et une série de tâches pratiques.

## 2. Glossaire de l'Unité

Mot	Définition
<b>ThingSpeak</b>	Une plateforme de services IoT pour surveiller les données collectées par le module WiFi ESP8266
<b>MIT App Inventor</b>	Logiciel de création d'applications
<b>Speech Recognizer</b>	Un service d'IA dans MIT App Inventor qui reconnaît la parole et la transforme en texte
<b>React App or Personal Audio Classifier</b>	Un outil de classification audio et d'entraînement de modèles d'apprentissage automatique (ML) compatible avec MIT App Inventor

## 3. Présentation de la « voiture robotique DIY qui peut être contrôlée et pilotée à l'aide de commandes vocales »

### 3.1 Description

Ce projet présentera aux apprenants les 5 grandes idées de l'IA et de l'IoT dans l'enseignement STEM à travers la création d'une voiture robotique DIY (Figure 1), programmée pour être pilotée à l'aide de commandes vocales. Plus précisément, les apprenants seront encouragés à concevoir et à assembler la voiture robotique en utilisant divers composants électroniques et matériaux simples, et à la programmer à l'aide d'environnements de programmation par blocs. Ils apprendront à collecter des données, en temps

réel, et à réfléchir aux résultats obtenus et à la manière dont ils peuvent être utilisés pour prendre des décisions spécifiques concernant les performances de la voiture robotisée. De plus, ils seront encouragés à concevoir et programmer une application qui utilise le service Speech Recognition AI pour permettre la navigation verbale de la voiture robotique. De plus, ils apprendront à créer un modèle entraîné pour optimiser les performances de l'application.

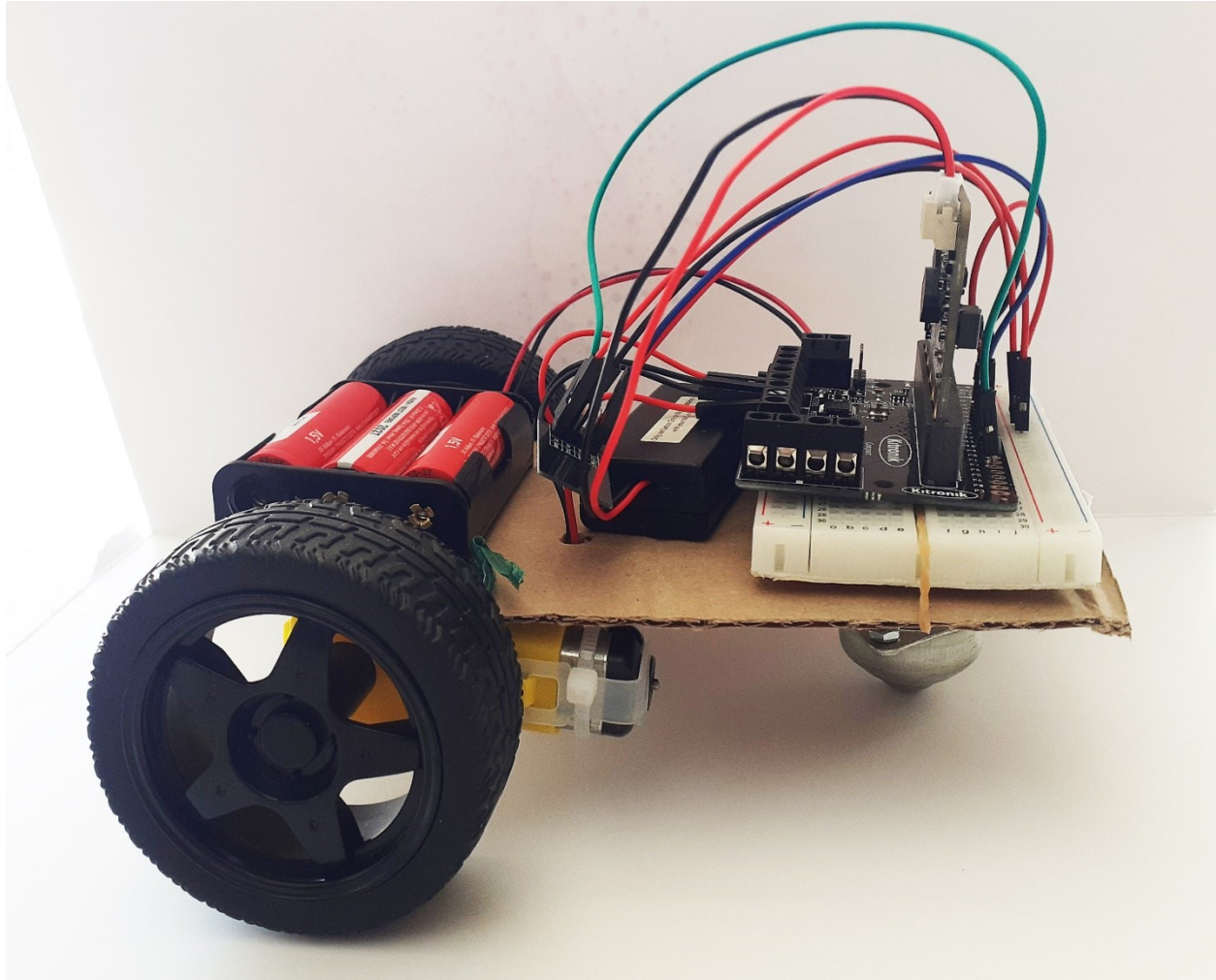


Figure 1: The DIY robotic car

Pour procéder aux activités contenues dans ce document, vous devez d'abord suivre les instructions incluses dans le fichier « T2.4\_Creating\_the\_robotic\_car.pdf », tout en encourageant vos élèves à réaliser quelques activités de programmation d'échauffement, similaires à celles contenues dans le fichier "T2.4\_WarmUp\_programming\_activities\_for\_the\_robotic\_car.pdf". Il est fortement recommandé de diviser vos élèves en équipes et de les encourager à discuter des différents aspects de chaque activité. Pour faciliter ce processus, vous trouverez dans ce document des liens vers du matériel qui peut vous aider à présenter chaque étape du projet en douceur, ainsi que quelques suggestions de questions qui peuvent vous aider à entamer un dialogue avec vos élèves sur les différentes parties et aspects du projet.



## 3.2 Objectifs et résultats d'apprentissage

À la fin de cette unité, les apprenants devraient être capables de :

- Identifier la présence et l'utilisation de la robotique IA dans la vie quotidienne
- Discuter et comprendre le rôle de l'IA en robotique
- Expliquer et discuter des différents aspects de l'intégration de l'IA dans un projet robotique grâce à la reconnaissance vocale et aux commandes vocales.
- Comprendre le fonctionnement des services d'IA tels que la reconnaissance vocale
- Comprendre comment les arbres de décision ou les organigrammes peuvent montrer des chemins logiques possibles et également conduire à des décisions sur les opérateurs qui peuvent être utilisées à un stade ultérieur.
- Discutez du rôle de l'apprentissage automatique (ML) dans la robotique IA et de la manière dont le ML peut être utilisé pour entraîner un artefact robotique à percevoir son environnement.
- Expliquer les concepts de base de la classification audio
- Identifier et discuter des avantages et des risques liés à la mise en œuvre de commandes vocales en conduite
- Identifier et discuter des avantages et des risques de la classification audio
- Expliquer les constructions/concepts de programmation de base liés à la mise en œuvre de méthodes de synthèse vocale.
- Comprendre les concepts clés qui sous-tendent l'IoT et les implications de la surveillance des données
- Réfléchir à l'impact des décisions basées sur les données dans la vie quotidienne
- Comprenez que les services tels que la reconnaissance vocale sont sujets aux erreurs.

Les étudiants apprendront également à :

- Construire un artefact robotique et créer des circuits dans le cadre d'une construction robotique
- Surveiller les données à l'aide d'un service de plateforme d'analyse IoT
- Créer des arbres de décision et des organigrammes pour représenter un type d'information
- Utiliser des commandes de programmation associées à des méthodes d'IA pour traiter un comportement spécifique d'un artefact robotique.
- Programmer un robot pour qu'il soit instruit à l'aide de commandes vocales
- Exprimer leurs idées à travers la programmation



- Utilisez l'outil Personal Audio Classifier ML pour classer différents sons.
- Évaluer les résultats produits par un outil ML
- Apporter des améliorations à un modèle formé basé sur l'évaluation
- Étudier les facteurs dans les données de formation qui peuvent conduire à des biais

### 3.3 Durée estimée de l'Unité

Il s'agit d'un projet assez vaste, nécessitant plusieurs heures pour aborder correctement tous les aspects inclus. La durée suivante est indicative et peut varier en fonction de l'âge et du niveau de vos élèves.

Activité 1 : 4 à 6 heures

Activité 2 : 8 – 15 heures

Activité 3 : 2 à 4 heures

Activité 4 : 2 à 4 heures

Activité 5 : 1 – 2 heures

## 3.4 Activité 1 – Présentation de la grande idée de la perception grâce à l'IoT :

### 3.4.1 Description

Cette activité est une intervention d'apprentissage d'échauffement pour présenter aux étudiants l'idée de perception à la lumière de l'IoT et de la robotique. Ils exploreront comment leur voiture robotique peut détecter et percevoir son environnement en collectant et en stockant des données, qui pourront ensuite être utilisées pour prendre des décisions concernant les performances de la voiture. En particulier, ils exploreront comment les capteurs intégrés à la carte BBC micro:bit, ou les capteurs qui peuvent être connectés à la carte (par exemple un capteur à ultrasons) peuvent collecter des données et être surveillés à l'aide d'un service de plateforme d'analyse IoT. Le service utilisé pour les besoins de cette activité est ThingSpeak, une plateforme qui permet l'agrégation, la visualisation et l'analyse de flux de données en direct dans le cloud.

### 3.4.2 Matériel

En plus du micro:bit BBC, le matériel requis pour cette activité est le module Wi-Fi ESP8266 et éventuellement quelques autres capteurs comme un capteur à ultrasons (de préférence le HC-SR04).

### 3.4.3 Configuration

Avant de commencer le câblage et le codage, vous devez créer un canal pour surveiller les données que vous recevez. Pour ce faire, rendez-vous sur le site ThingSpeak (<https://thingspeak.com/>) et connectez-vous à votre compte, en saisissant votre Email (1.), ou créez un compte en cliquant sur « Créer un » (2.) (Figure 2).

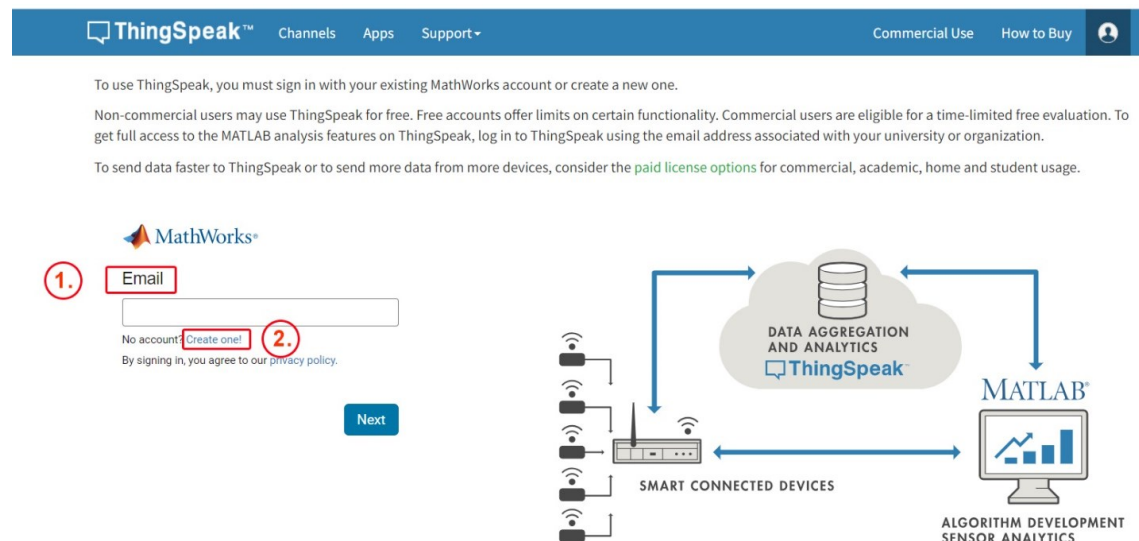


Figure 2 : Menu pour se connecter ou créer un nouveau compte sur ThingSpeak

Ensuite, allez dans le menu Canaux et cliquez sur le bouton « **Nouveau canal** » sous « Mes canaux » pour créer un nouveau canal pour enregistrer et afficher les données de vos capteurs (Figure 3).

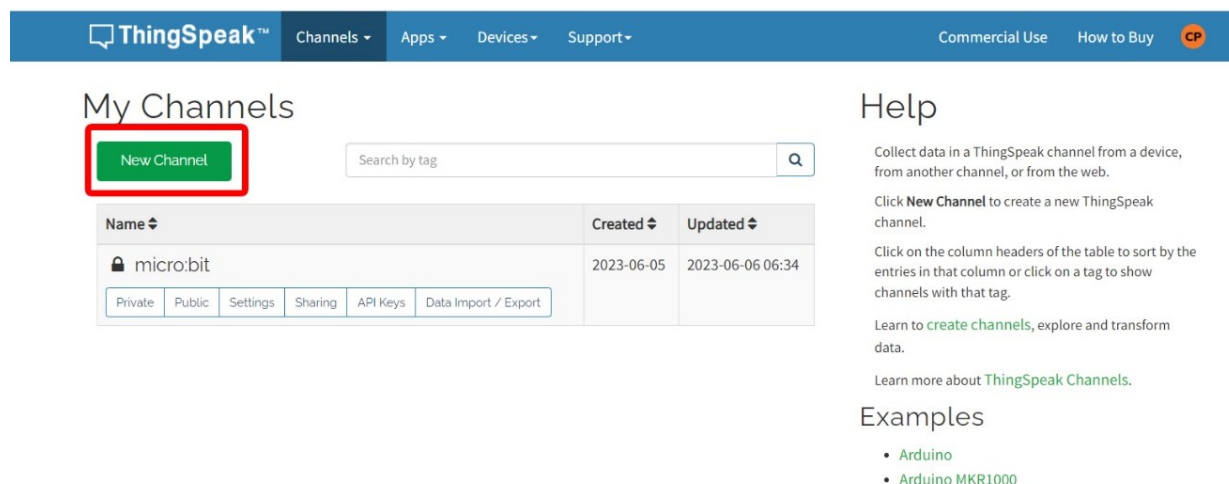


Figure 3 : Le bouton Nouvelle chaîne, situé dans le menu Chaînes

L'étape suivante consiste à créer votre nouvelle chaîne en insérant le nom (1.) de votre chaîne (par exemple, microbit) et les paramètres que vous souhaitez surveiller dans les champs fournis (2.) (Figure 4). Par exemple, dans la figure 3, l'utilisateur a inséré la température dans le champ 1 et la lumière dans le champ 2. Vous pouvez activer plus de champs en cliquant sur la case à cocher (3.), à côté d'un champ. Vous pouvez éventuellement rédiger une description pour vous aider à vous rappeler les paramètres que votre chaîne surveille.

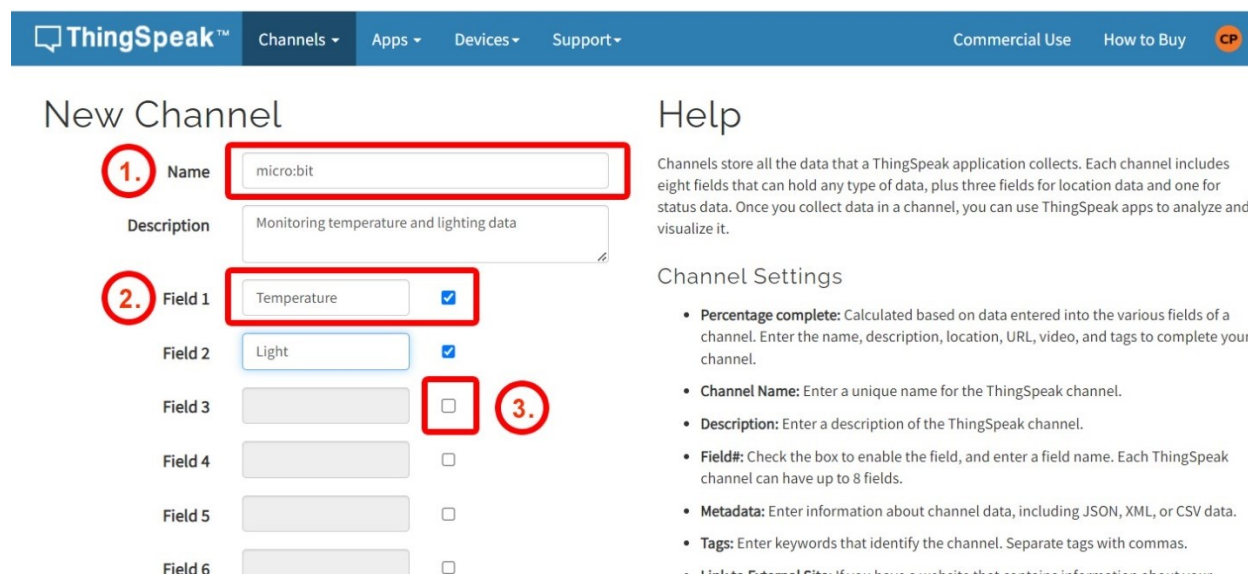


Figure 2: Creating a New Channel

Une fois que vous avez déclaré tous les paramètres que vous souhaitez surveiller, faites défiler la page Nouveau canal et cliquez sur le bouton Enregistrer le canal (Figure 5) pour enregistrer toutes les informations insérées.

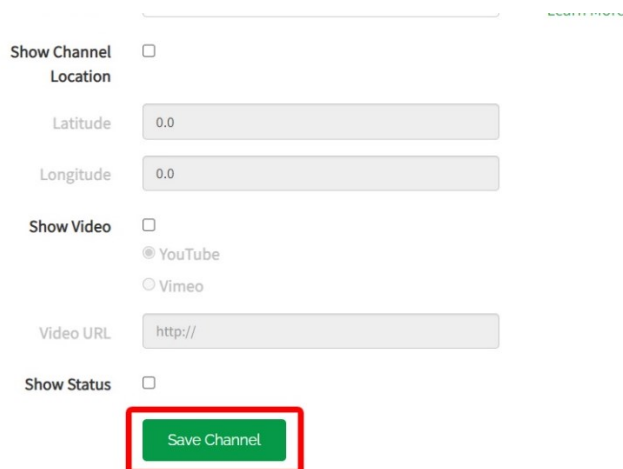


Figure 3: Scrolling down the New Channel page to click on the Save Channel button

Après avoir appuyé sur Enregistrer la chaîne, vous serez automatiquement redirigé vers votre nouvelle chaîne. Cliquez sur le menu Clés API (1.) et notez la clé (2.) qui apparaît dans le champ Écrire la clé API (Figure 6).

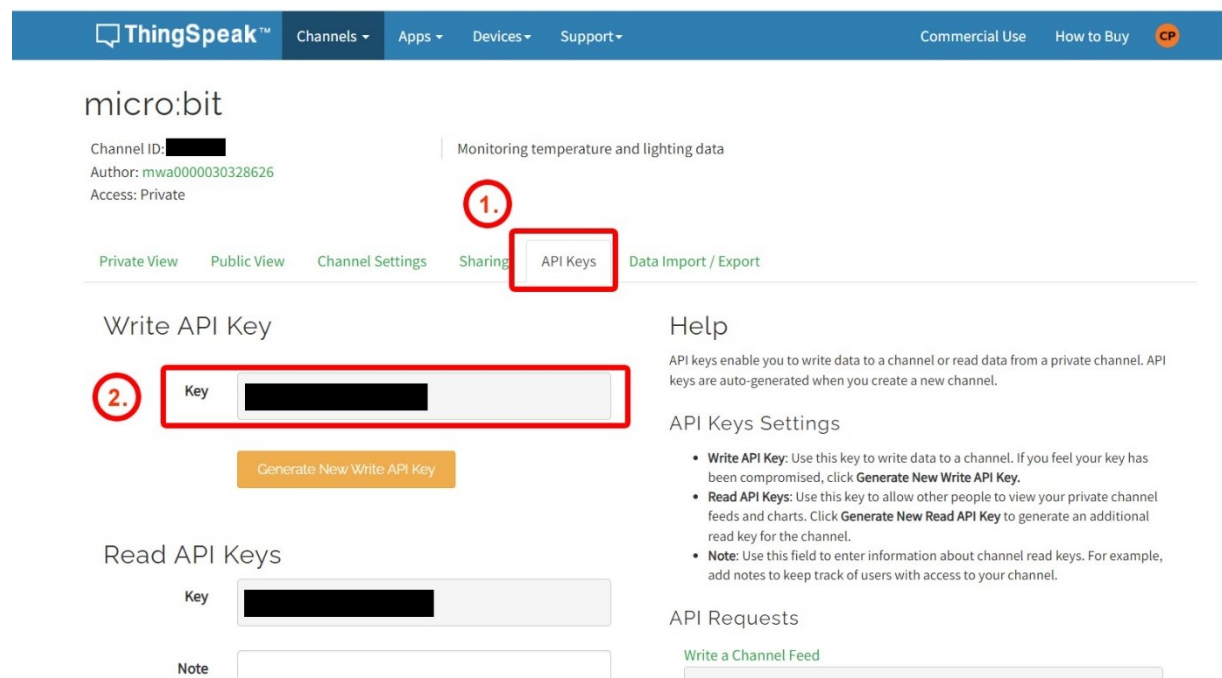


Figure 4: Finding the API Key

Cette clé sera ensuite insérée dans l'un des blocs de commande que vous utiliserez dans l'environnement de programmation Makecode.

Maintenant que vous avez enregistré votre chaîne sur le site ThingSpeak, vous pouvez procéder à la phase de câblage et de programmation du module Wi-Fi ESP8266.

### 3.4.3.1 Câblage

La figure 7 présente comment connecter le module Wi-Fi ESP8266 au microcontrôleur BBC micro:bit. Le module ESP8266 possède 8 broches différentes. Vous devez en connecter 5 à la carte micro:bit. Il faut notamment connecter les broches 3V3 et EN (1) à l'alimentation (3V), et la broche GND (4) à la broche GND du micro:bit. Enfin, connectez la broche TX (2) et la broche RX (3) aux broches P1 et P0 respectivement. Une fois le circuit terminé, connectez le microbit à votre ordinateur à l'aide d'un câble USB.

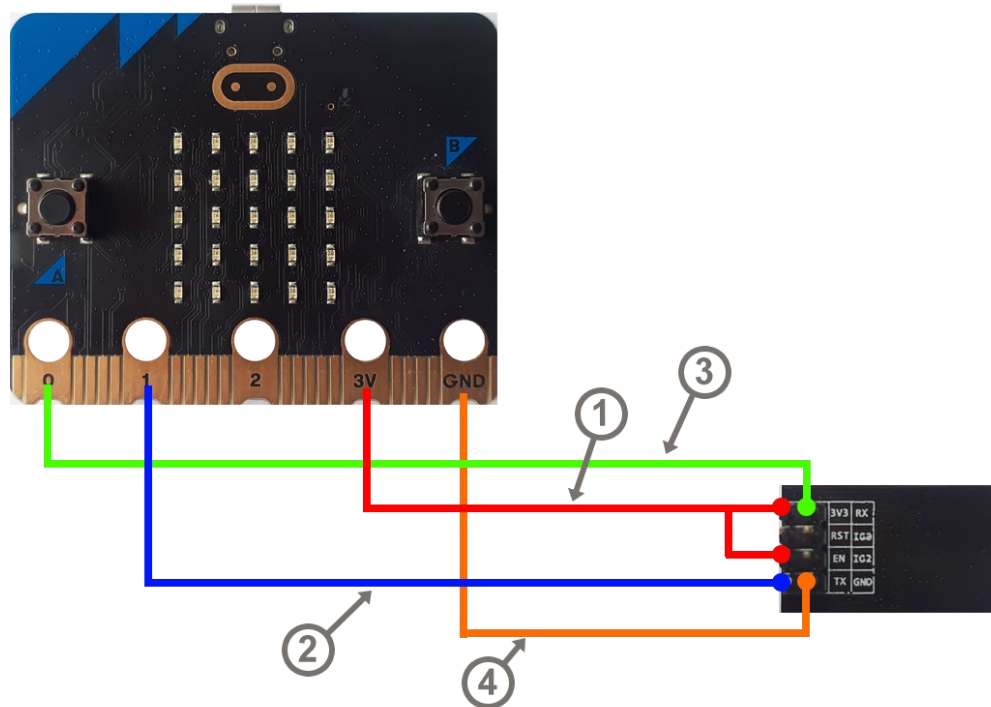


Figure 5: Connecting ESP8266 WiFi module to the BBC micro:bit microcontroller

### 3.4.3.2 Code

L'étape suivante consiste à ouvrir l'environnement de programmation basé sur des blocs Microsoft Makecode (<https://makecode.microbit.org/#>) et à créer un nouveau projet. Sur la page d'accueil de Makecode, cliquez sur l'onglet Nouveau projet (1.) et dans le menu contextuel, entrez un nom pour votre projet (par exemple ThingSpeak\_WarmUp). Cliquez ensuite sur le bouton Créer (2.) (Figure 8).

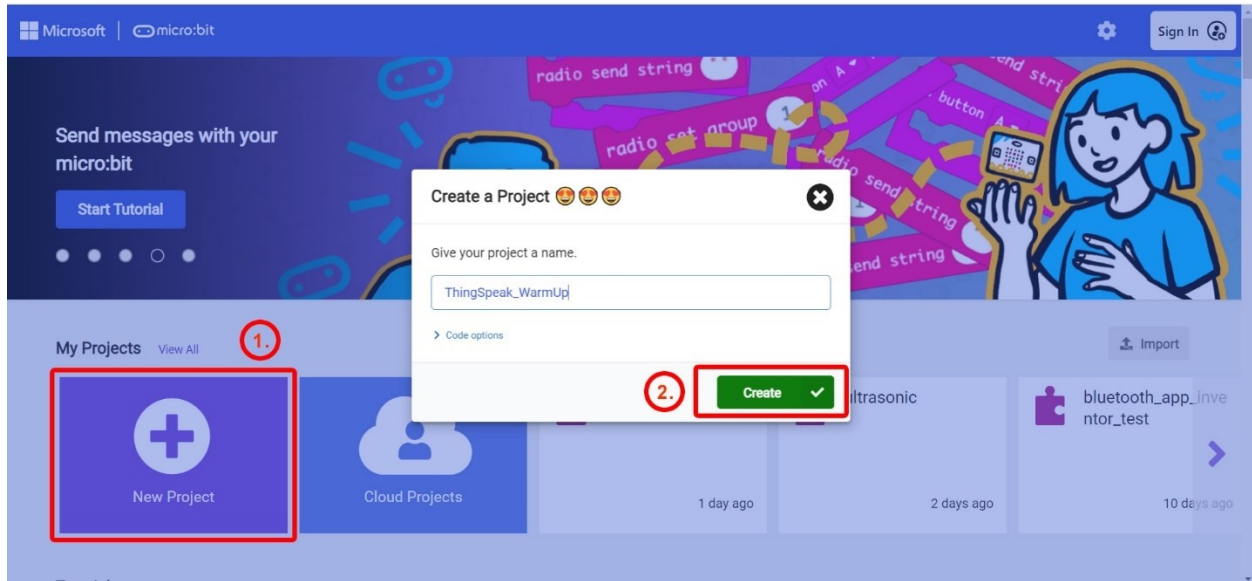


Figure 6: Creating a new project in the MakeCode programming environment

Vous serez automatiquement redirigé vers la zone où vous pourrez assembler votre code. Pour inclure les blocs de programmation du module Wi-Fi ESP8266, cliquez sur le menu +Extensions (Figure 9).

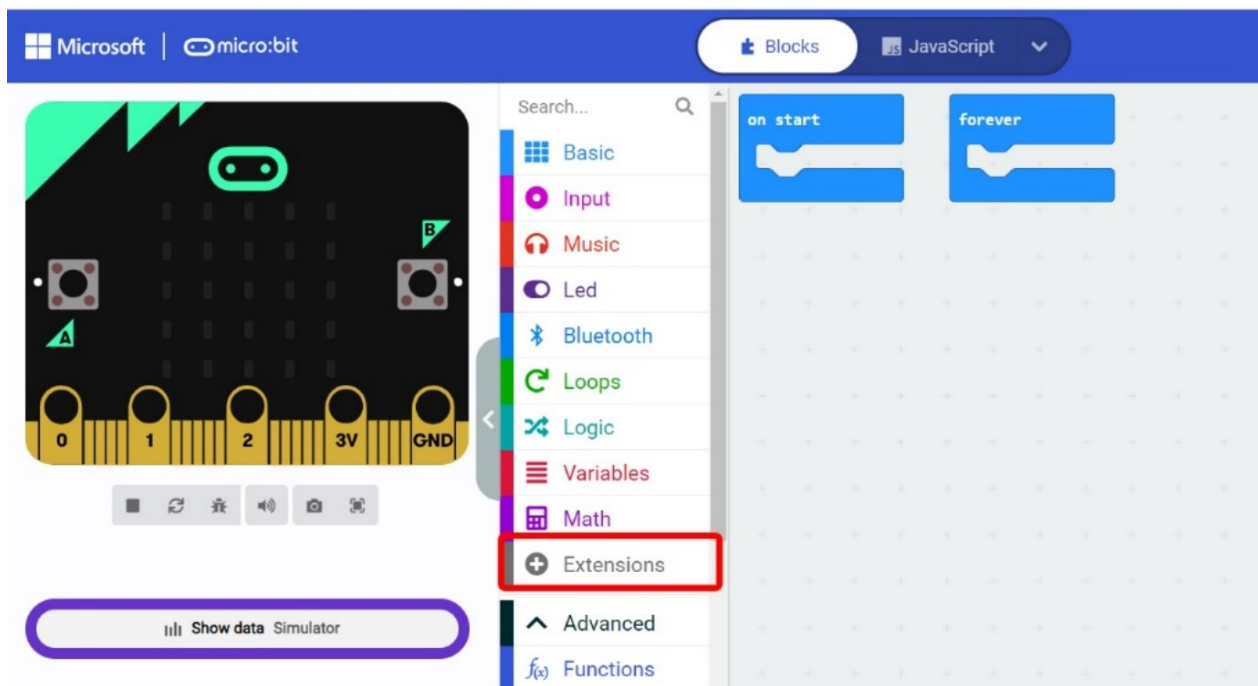


Figure 7: The +Extensions menu

Tapez « ThingSpeak » dans le champ « Rechercher ou saisir l'URL du projet » (1.) et sélectionnez l'extension correspondante (2.) (Figure 10).



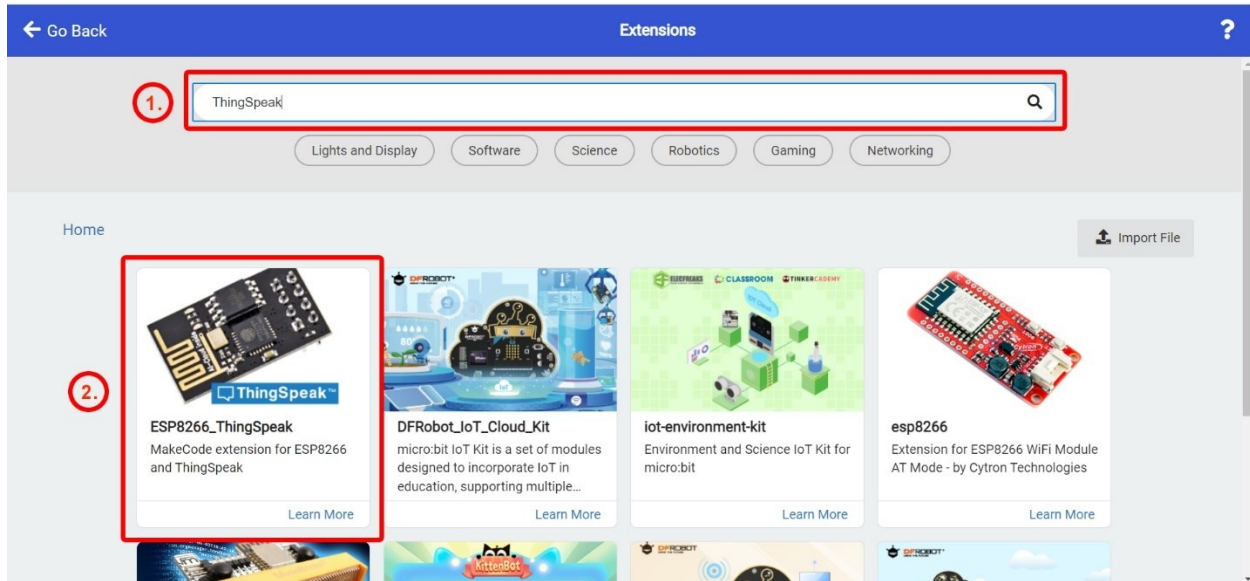


Figure 8: Finding and selecting the ESP8266\_ThingSpeak extension

Un nouveau menu de blocs (c'est-à-dire ESP8266 ThingSpeak) apparaîtra dans le menu de commandes (Figure 11).

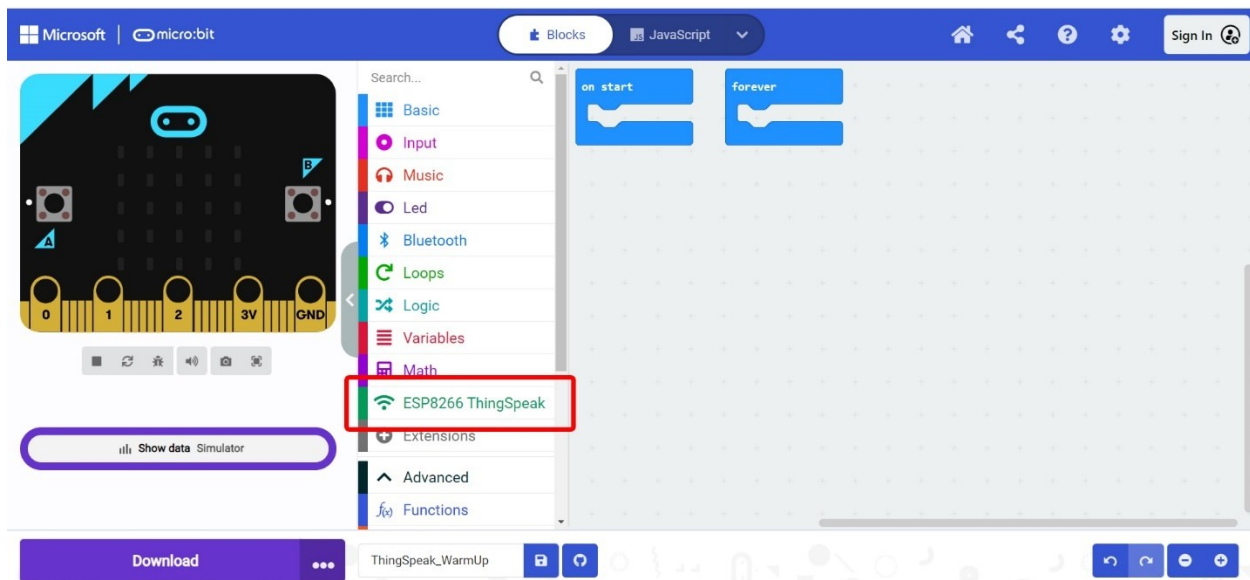


Figure 9: The new menu with block of commands for programming the ESP8266 WiFi module

Tout d'abord, vous devez initialiser le module Wi-Fi en déclarant les broches auxquelles il est connecté et en insérant les informations du routeur Wi-Fi auquel vous allez vous connecter. Pour ce faire, cliquez sur le menu ESP8266 ThingSpeak et sélectionnez la commande d'**initialisation du bloc ESP8266** dans le menu flottant (Figure 12).



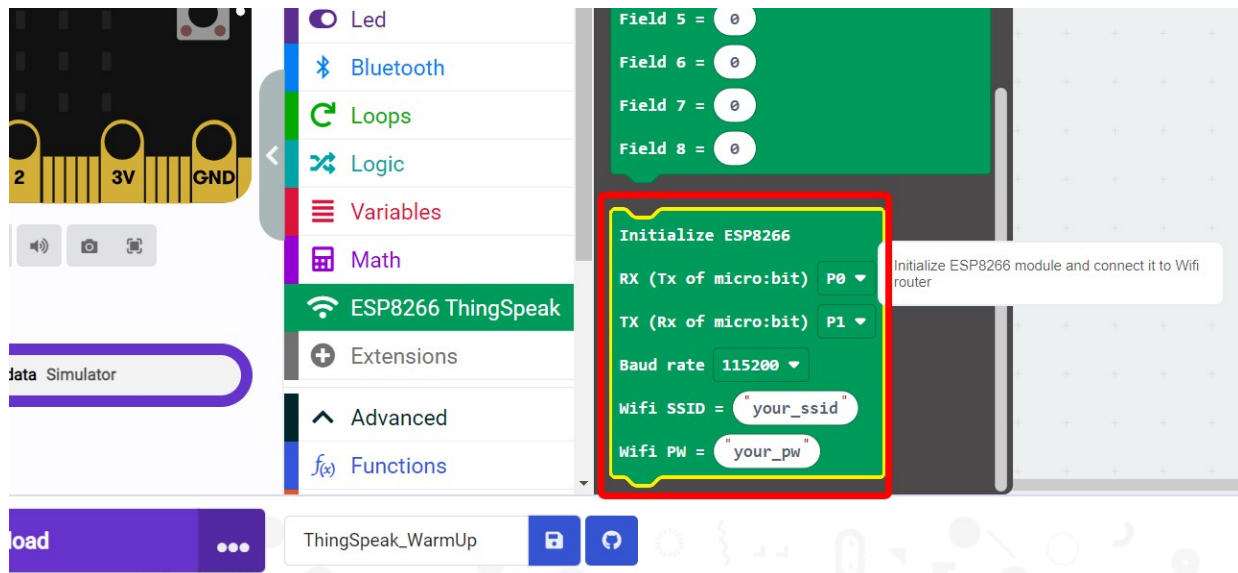


Figure 10: Finding and selecting the Initialize ESP8266 block command

Faites glisser et accrochez cette commande sur le bloc « **au démarrage** ». Dans les champs RX et TX (1.) précisez les ports auxquels vous avez connecté les broches correspondantes du module (à savoir P0 et P1) (Figure 13). Ensuite, dans les champs wifi SSID et wifiPW (2.) saisissez le nom de votre routeur Wi-Fi et son mot de passe. Cela connectera le module Wi-Fi à votre routeur Wi-Fi.

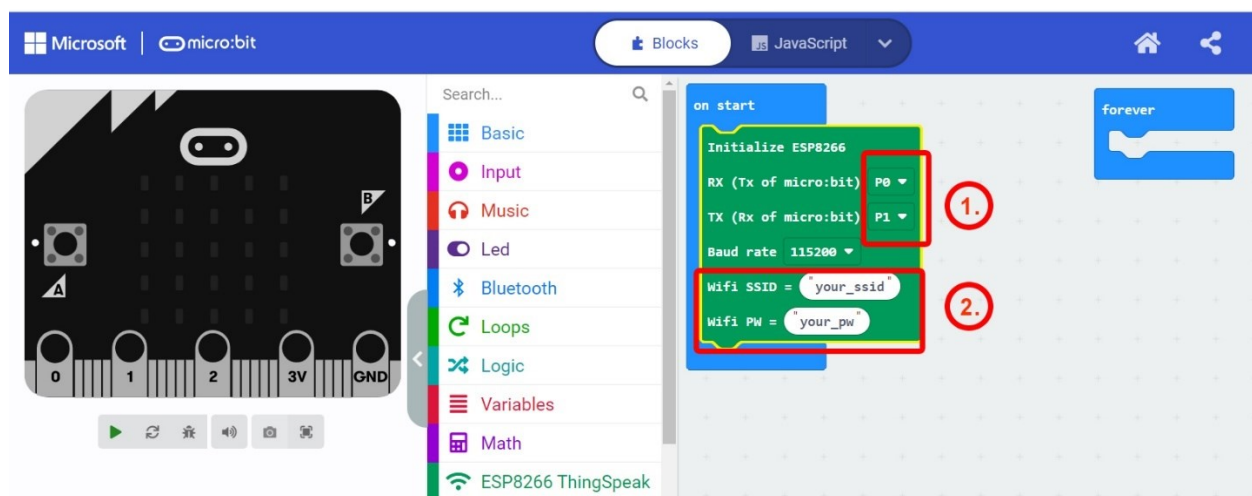


Figure 11: Declaring the pins that ESP8266 is connected and inserting information about wifi connection

L'étape suivante consiste à permettre au module Wi-Fi de télécharger les données qu'il reçoit du microbit vers la plateforme ThingSpeak. Pour ce faire, faites glisser et accrochez la commande de bloc « télécharger des données vers ThingSpeak » du menu ESP8266 TingSpeak vers le bloc « pour toujours ». Dans le champ Write API key (1.) (Figure 14), saisissez la clé Write API que vous avez reçue après la création de votre chaîne sur la plateforme ThingSpeak (Figure 6). Ensuite, pour déclarer les données qui seront surveillées par la plateforme ThingSpeak (c'est-à-dire la température et la lumière), allez dans le menu « Entrée », et faites glisser et déposez la « température » (3.) et le « niveau de lumière » (2. .) saisissez des commandes dans le champ 1 et le champ 2, respectivement (Figure 14).

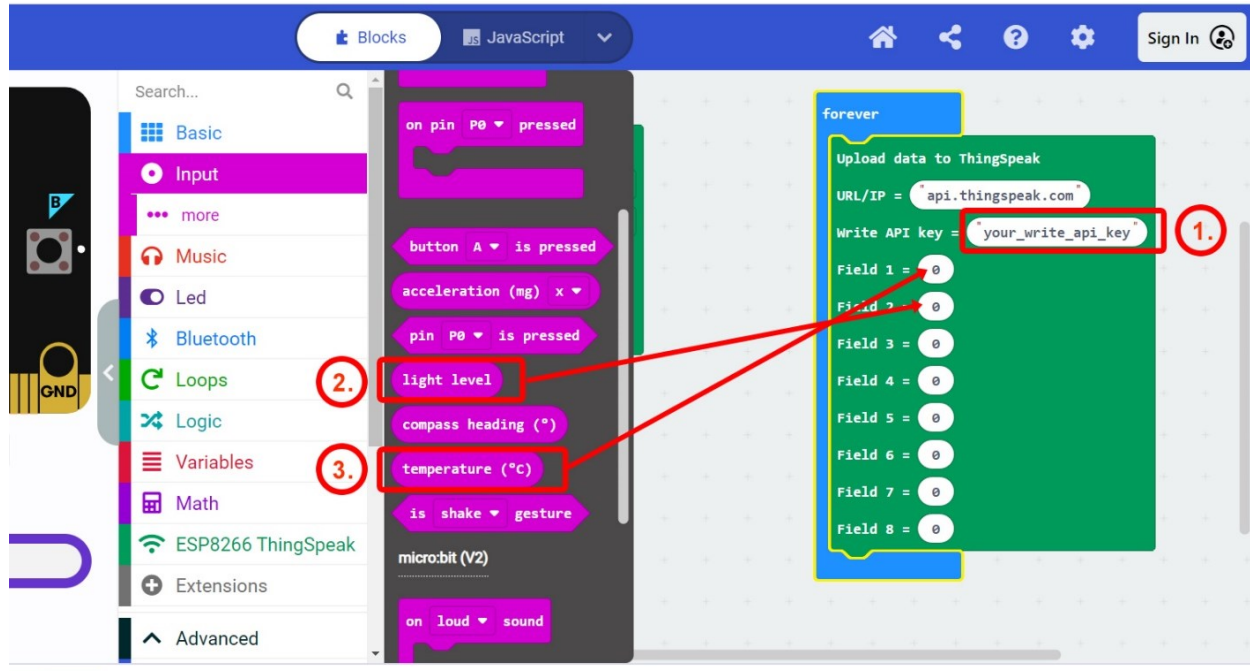


Figure 12: Inserting the API key, and adding the parameters that will be monitored

Après cela, vous pouvez télécharger le script sur votre micro:bit en cliquant sur le bouton de téléchargement.

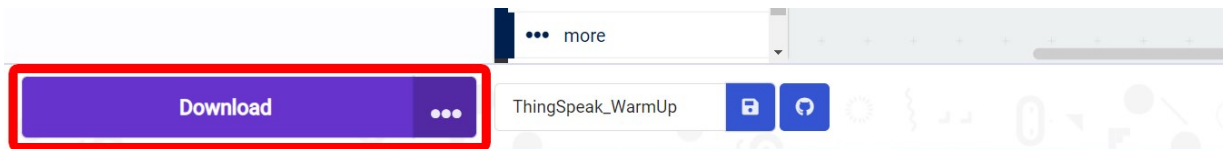


Figure 13: The Download button

**Remarque :** Gardez à l'esprit que vous devrez peut-être ajouter une commande de pause de 15 secondes dans la commande de boucle permanente, pour permettre un transfert fluide des données vers la plateforme ThingSpeak.

En option, vous pouvez ajouter le script illustré dans la figure 16 à votre code pour afficher certaines des données reçues sur l'écran LED de votre microbit. Ces commandes (c'est-à-dire « série sur les données reçues... » « chaîne de lecture série ») se trouvent dans le menu des commandes série.

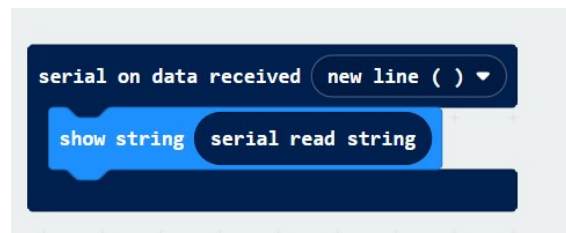


Figure 14: Displaying the received data on microbit's LED screen

Après avoir téléchargé le code, accédez à la plateforme ThingSpeak et utilisez l'onglet Vue privée pour voir comment les données reçues par microbit évoluent au fil du temps (Figure 17).

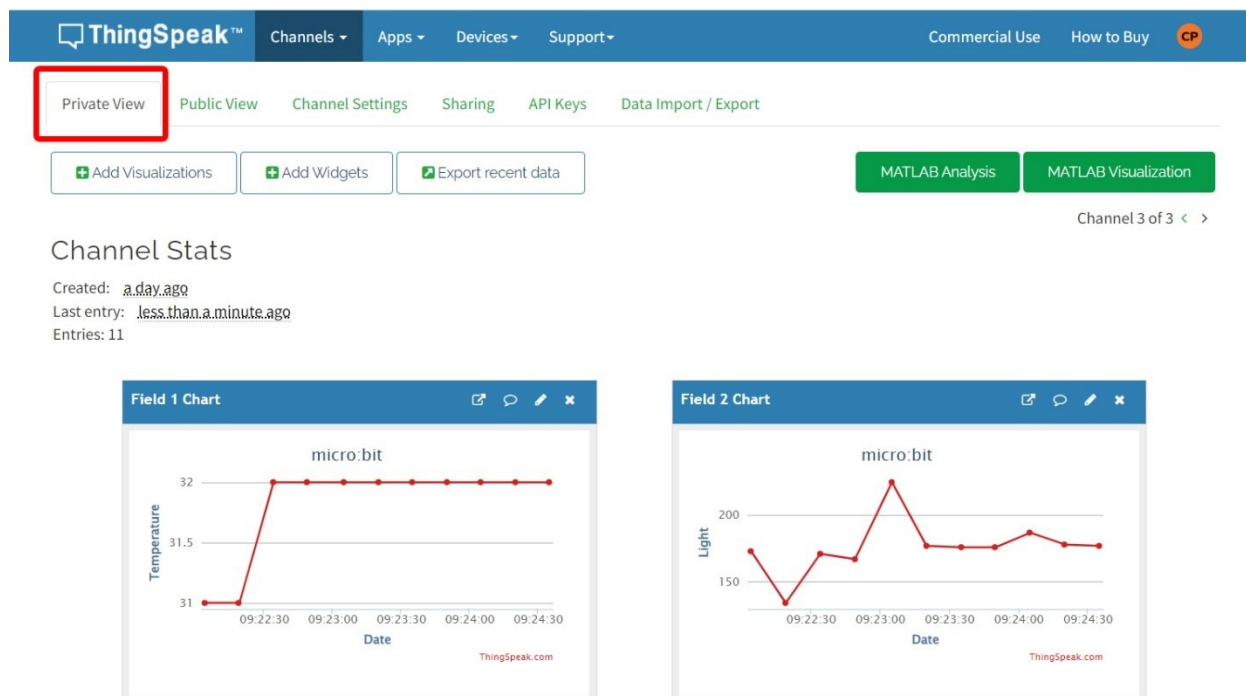


Figure 15: Checking how the received data changing over time

### 3.4.4 Expérience 1

Répétez le processus susmentionné avec vos élèves. Avant cela, créez des comptes sur la plateforme ThingSpeak et partagez-les avec vos élèves. Divisez-les en équipes de 2 ou 3 et donnez à chaque équipe une carte micro:bit et un module Wi-Fi ESP8266. Encouragez vos élèves à regarder le module Wi-Fi pour explorer les broches intégrées, puis demandez à chaque équipe de connecter le module à la carte micro:bit. Pour faciliter cette démarche, vous pouvez leur remettre le plan du circuit contenu dans le fichier « Circuit\_card\_Arctivity1.pdf ».

Parcourez toutes les équipes pour vous assurer qu'elles ont toutes réussi à créer le circuit. Passez ensuite au processus de programmation. Donnez à chaque équipe l'un des comptes que vous avez créés sur la plateforme ThingSpeak et demandez-leur de se connecter. Encouragez-les à consulter les différents onglets (c'est-à-dire Vue privée, Clés API, etc.). Donnez ensuite à chaque équipe une copie du fichier « Half\_baked\_Activity1.pdf » et demandez-leur de créer le script pour programmer le module Wi-Fi pour surveiller la température et le niveau de lumière enregistrés par le micro:bit. Demandez-leur de télécharger le script sur leur micro:bit et conseillez-leur d'observer les données surveillées sur la plateforme ThingSpeak. Encouragez-les à utiliser une lampe de poche ou/et à rapprocher le circuit de surfaces plus froides ou plus chaudes pour observer comment les données surveillées changent dynamiquement.

Discutez ensuite avec eux de certaines des questions suivantes :

- Quelles informations sont extraites ?
- Ces informations affectent-elles les performances de la voiture robotisée ?

- Ces informations peuvent-elles conduire à des décisions basées sur des données pour optimiser les performances de la voiture robotisée ?
- Pensez à d'autres paramètres qui pourraient être enregistrés. De quels types de capteurs devez-vous vous connecter à la voiture robotisée pour enregistrer ces paramètres ? Tous ces paramètres peuvent-ils conduire à des décisions significatives basées sur des données pour les performances de la voiture robotisée ?
- Pensez aux services d'IA qui pourraient bénéficier de ces données. Comment ces services peuvent-ils utiliser ces paramètres pour extraire les informations derrière un ensemble de données, et ainsi optimiser les performances de la voiture robotisée ?

Grâce à cette activité et à la discussion qui suit, les élèves comprendront :

- Comment les capteurs peuvent être utilisés non seulement pour détecter un environnement, mais également pour prendre des décisions fondées sur des données.
- Que la perception d'un artefact robotique est basée sur des données collectées
- L'importance de surveiller et d'évaluer les données pour prendre des décisions optimales pour les performances d'un véhicule autonome
- Les données surveillées peuvent être utilisées par les services d'IA pour optimiser leurs performances.

## 3.5 Activité 2 : Introduire l'idée de représentation et de raisonnement

### 3.5.1 Description

Cette activité vise à initier les étudiants à la 2ème Grande Idée, à savoir l'idée de Représentation et de Raisonnement, et à les familiariser avec la manière dont un artefact robotique peut « apprendre » à partir de données. Lors de cette activité, les élèves doivent garder à l'esprit que l'IA et les agents intelligents peuvent « penser » : a) en construisant des représentations du monde sous forme de structures de données, et b) en utilisant des algorithmes pour les aider à donner du sens à ces données. structures (raisonnement). Par conséquent, les étudiants apprendront comment les structures de données peuvent être représentées et comment des algorithmes peuvent être utilisés pour extraire des informations spécifiques des représentations. À cette fin, ils seront encouragés à créer des arbres de décision. Grâce à ce processus, ils comprendront les critères de sélection d'un algorithme qui mènera à la meilleure solution possible.

Les étudiants apprendront notamment à programmer une application permettant à l'utilisateur de naviguer dans la voiture robotisée à l'aide de commandes vocales. Pour ce faire, ils utiliseront le service Speech Recognition AI et seront initiés à l'environnement App Inventor. Avant de développer l'application, ils créeront un arbre de décision pour représenter la décision que leur voiture robotique devrait prendre lorsqu'une commande vocale est reçue. Ils décideront ensuite du chemin logique à suivre lors de la programmation de leur application.

### 3.5.2 Création d'un arbre de décision

Pour commencer à créer un arbre de décision, vous devez réfléchir à la manière dont le service d'IA sera mis en œuvre et aux actions possibles qui peuvent être activées. L'objectif principal est de créer une application avec un bouton qui enregistre les commandes vocales lorsqu'on appuie dessus. Sur cette base, vous pouvez démarrer votre arbre en supposant que la commande reçue contient le mot « forward ». Si tel est le cas, la voiture robotique doit recevoir l'ordre d'avancer. S'il ne contient pas « forward », alors il existe au moins deux chemins logiques différents : a) La commande n'est pas reconnue du tout, donc la voiture robotique ne bougera pas (ou elle continuera à exécuter la commande précédente), b) la commande reçue contient un autre mot-clé directionnel, donc la voiture robotique recevra des instructions en conséquence. La figure 18 montre un exemple indicatif d'un tel arbre de décision. L'arbre n'est pas complet. Plus de branches peuvent être ajoutées en fonction des différentes commandes reçues.

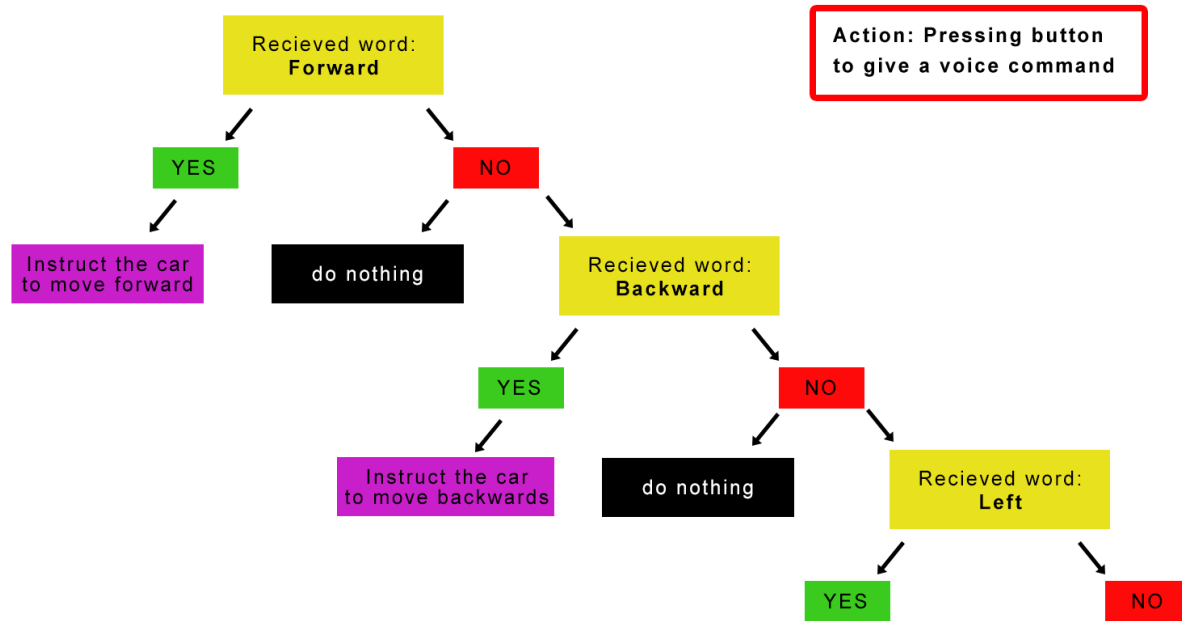


Figure 16: An indicative decision tree depicting the possible actions that can be activated when Press to Speak button is pressed

Sur la base de l'arbre de décision, les étudiants peuvent décider quels chemins logiques doivent être utilisés lors de la programmation. L'arbre de décision illustré à la figure 18 suggère qu'une série de commandes « si... alors » doit être utilisée pour programmer avec succès l'application.

### 3.5.3 Conception et programmation de l'application

Après avoir analysé les actions possibles qui peuvent être activées, l'étape suivante consiste à créer l'application qui utilisera le service Speech Reconnaissance AI pour permettre à l'utilisateur de contrôler la voiture robotique à l'aide de commandes vocales. Avant de commencer à créer l'application, il est fortement recommandé de réaliser avec vos élèves l'activité d'échauffement contenue dans le fichier « 2.4\_App\_Inventor\_Warm\_Up.pdf », et de continuer à travailler sur le même projet « .aia ». Vous pouvez également ouvrir le fichier de projet « Remote\_control\_Microbit.aia » et travailler dessus. Dans ce cas, il est fortement conseillé de sauvegarder le projet sous un nouveau nom.

Remarque : Si vous utilisez le fichier .aia susmentionné, au lieu de faire l'activité d'échauffement d'App Inventor, vous devez également penser à conseiller à vos élèves de télécharger le script décrit dans le fichier « T2.4\_Programming\_the\_robotic\_car.pdf » sur la voiture robotique. .

#### Procédé de design

Tout d'abord, vous devez ajouter un nouveau bouton qui activera le service Speech Recognition AI. Faites glisser et déposez un élément Button sous les boutons de navigation (1), renommez-le en btn\_speak (2), tout en changeant son texte en « Appuyez pour parler » (3) (Figure 19).



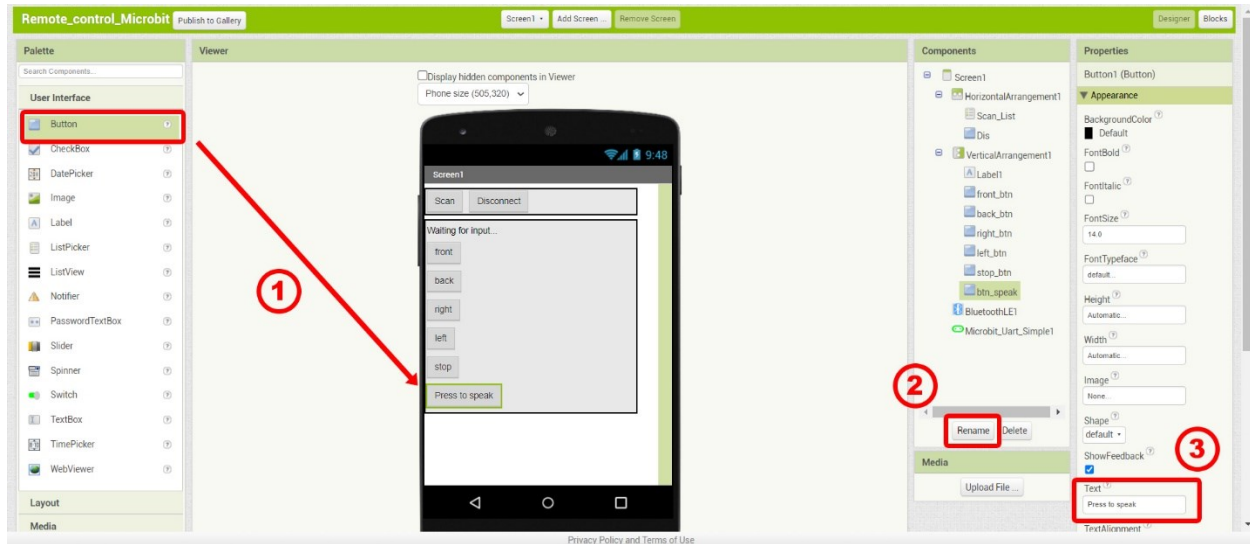


Figure 17: Adding a new button (1) and renaming to btn\_speak (2), while changing the text to Press to speak (3)

Lorsque vous appuyez sur le bouton « Appuyez pour parler », l'application commencera à détecter votre voix. Ensuite, par défaut, il utilisera le service d'IA de synthèse vocale pour convertir les commandes vocales reçues en texte.

Pour utiliser ce service, vous devez ajouter le composant « SpeechRecognizer ». Trouvez-le dans l'onglet « Média », puis faites-le glisser et déposez-le sur l'écran (Figure 20).

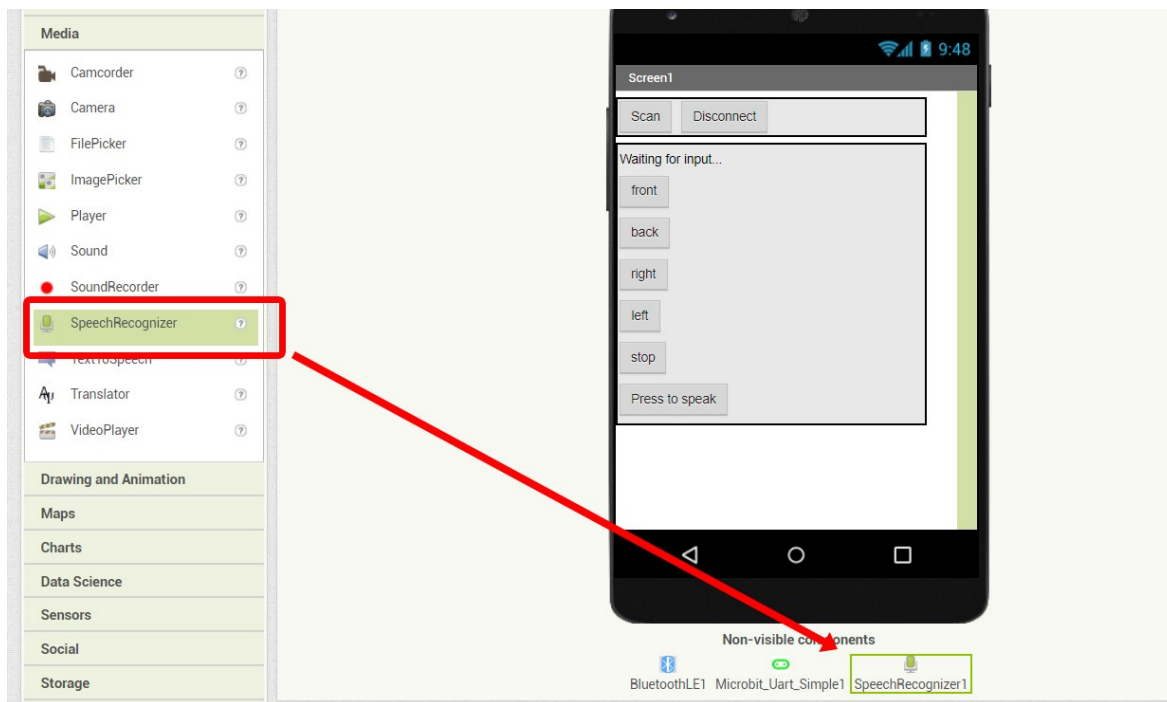


Figure 18: Dragging and dropping the SpeechRecognizer component to the design area

**Remarque :** Semblable aux composants BluetoothLE et Microbit\_Uart\_Simple, SpeechRecognizer est un composant non visible et n'apparaît donc pas à l'écran, mais dans la zone située sous l'aperçu de l'écran.



## Processus de programmation

Cette solution de programmation demandera à votre application de reconnaître une commande vocale, chaque fois que vous appuierez sur le bouton « Appuyer pour parler ».

Allez dans le menu Bloc et sélectionnez le composant « btn\_speak ». Dans le menu flottant, sélectionnez le gestionnaire d'événements « when btn\_speak.Click do » et faites-le glisser dans la zone d'assemblage du script. Ensuite, cliquez sur le composant Speech Recognizer et dans le menu flottant, sélectionnez la commande « appeler SpeechRecognizer.GetText » et placez-la dans l'événement du gestionnaire.



Ce bloc programme le bouton btn\_speak pour « obtenir le texte » détecté par SpeechRecognizer, à chaque fois que le bouton « Appuyer pour parler » est enfoncé. Parallèlement, le service d'IA de synthèse vocale est activé et la voix de l'utilisateur est convertie en texte, permettant à la voiture robotique d'effectuer le mouvement demandé (c'est-à-dire avancer si la commande vocale reçue est « devant », etc.)

**Remarque :** le microphone de l'appareil intelligent de l'utilisateur est utilisé pour la capture et l'enregistrement de la voix.

## Réagir aux commandes vocales

Grâce à la méthode de programmation ci-dessus – et en implémentant le service d'IA de synthèse vocale – nous avons trouvé un moyen de stocker une commande vocale sous forme de texte. Ce texte est utilisé comme entrée pour commander à la voiture robotique d'effectuer différents mouvements. Pour ce faire, les commandes vocales entrantes sont filtrées et recherchées selon des mots-clés spécifiques qui s'appliquent à notre cas, à savoir recto, verso, etc. De cette manière, l'application peut correspondre à un certain nombre de commandes vocales différentes. Le tableau suivant explique brièvement ce concept :

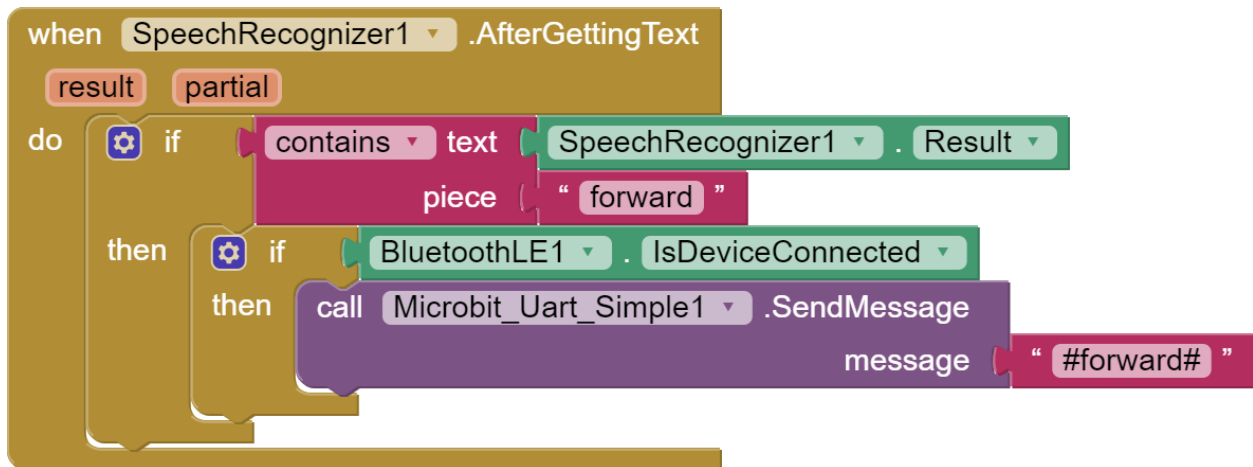
<i>Si la phrase reçue par la reconnaissance vocale contient le mot :</i>	<i>Ensuite, envoyez le message :</i>	<i>Et l'automobile :</i>
“forward”	#forward#	move forward

"backwards"	#backwards#	move backwards
"left"	#left#	turn left
"right"	#right#	turn right
"stop"	#stop#	stop

Des phrases similaires à celles-ci auront le même effet sur le mouvement de la voiture :

En avant : « Avance », « Avance, s'il te plaît », « Je veux que tu avances », etc...

Le script suivant demande à l'application d'envoyer le message #front# à la voiture robotique, lorsqu'une commande vocale enregistrée contient le mot « forward ». Lorsque le message #front# est reçu par la voiture robotique (sur la base du code déjà téléchargé sur la voiture robotique depuis l'environnement Makecode), la voiture robotique avance.



Dans ce qui suit, la fonctionnalité de chaque bloc de commandes est expliquée plus en détail :

When\_SpeechRecogniser.AfterGettingText : ce bloc reçoit le résultat du texte après que la commande vocale enregistrée a été convertie en texte. Ainsi, lorsque la commande vocale de l'utilisateur est convertie en texte par SpeechRecogniser, ce bloc exécutera les blocs de commandes suivants :

Si le SpeechRecognizer.Result, à savoir le texte renvoyé à la suite du processus SpeechRecognizer, contient\_text (), l'élément "forward" (c'est-à-dire que le texte reçu contient le mot "forward")

Ensuite, si BluetoothLE1 est connecté à l'appareil souhaité, le Microbit\_Uart\_Simple1.SendMessage est appelé et envoie le message #forward# (texte « forward ») à la voiture robotique. En conséquence, la voiture robotique avance.

Pour le mettre en conformité avec l'arbre de décision produit :

« Lorsque SpeechRecogniser a converti la commande vocale de l'utilisateur en texte, vérifiez si le résultat de SpeechRecogniser (à savoir le texte) contient le mot « forward ». Si tel est le cas, envoyez le message « front » via Microbit\_Uart\_Simple ».

Le reste des commandes vocales peut être reconnu de la même manière. Par conséquent, le script ci-dessus doit être étendu en ajoutant quatre autres conditions « si... alors », ainsi que les morceaux de texte et les messages texte correspondants, qui demanderont à notre voiture robotique d'exécuter un mouvement spécifique (c'est-à-dire, en arrière et le message « #backwards # » pour reculer, à droite et message « #right# » pour tourner à droite etc.).

L'intégralité du script est présentée dans l'image suivante (Figure 21) :

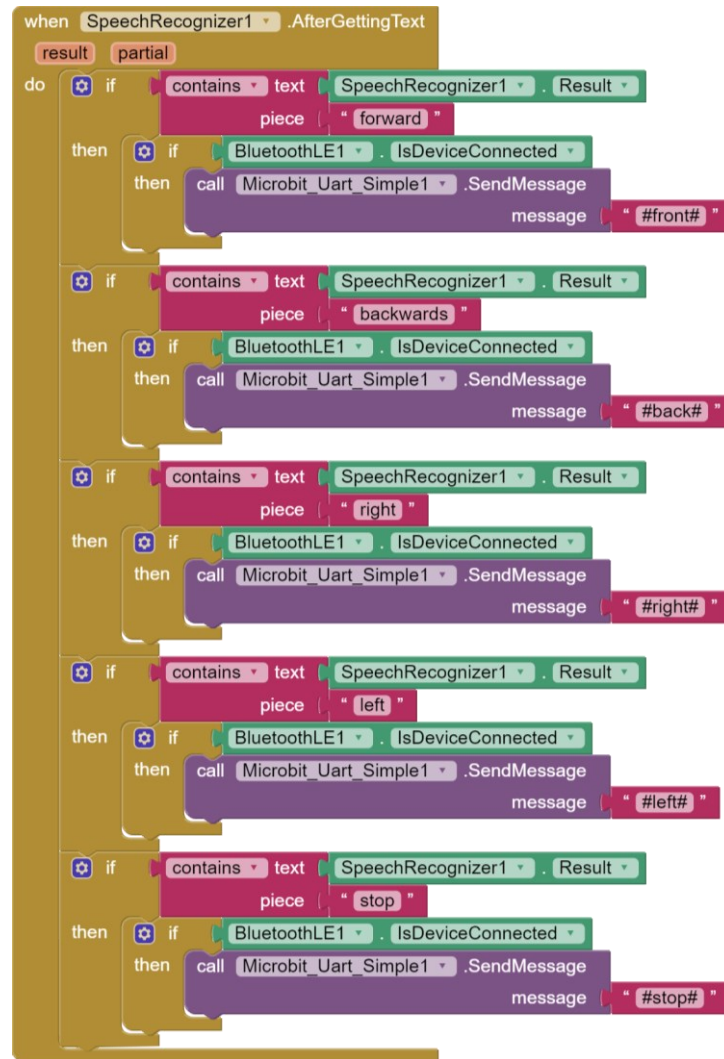


Figure 19: The entire script for SpeechRecognizer component after getting text

### Limites – vers une solution de programmation optimale

La solution de programmation ci-dessus présente un inconvénient majeur ; il oblige l'utilisateur à appuyer sur le bouton « Appuyer pour parler » à chaque fois qu'il souhaite donner une commande vocale. Cela rend l'ensemble du processus moins intuitif en termes d'habitudes de conduite et peut également réduire l'enthousiasme initial de vos élèves, car ils peuvent trouver cette solution très similaire au processus de télécommande (c'est-à-dire, toujours appuyer sur un bouton pour donner une seule commande vocale. ). Afin de pallier cet inconvénient, une solution de programmation optimale est également proposée, dans laquelle l'utilisateur n'a qu'à appuyer une fois sur le bouton « Appuyez pour parler » au début du processus, puis l'application essaiera d'entendre la voix de l'utilisateur tous les 2. secondes.

**Remarque** : Il s'agit d'une solution légèrement plus avancée, donc en fonction du niveau de vos élèves, vous pouvez sauter cette partie et terminer cette activité d'apprentissage ici.

### Ajout d'un capteur d'horloge

Dans cette solution, nous ajouterons un capteur d'horloge qui définira un compte à rebours pour l'application. Lorsque le bouton « Appuyez pour parler » est enfoncé, la minuterie est déclenchée et le

compte à rebours commence pour un intervalle de temps spécifié. Pendant cet intervalle de temps, l'application écoute une commande vocale entrante. Si l'application reconnaît une commande vocale valide, elle demandera alors à la voiture robotique d'effectuer le mouvement correspondant. Sinon, il continuera à chercher jusqu'à ce qu'il entende une commande valide. Lorsque le compte à rebours se termine, la minuterie est automatiquement réinitialisée et un nouveau compte à rebours - pour le même intervalle de temps - commence.

Par conséquent, il suffit à l'utilisateur d'appuyer une seule fois sur le bouton « Appuyer pour parler » pour initialiser l'ensemble du processus, en gardant à l'esprit que l'application ne peut recevoir qu'une seule commande vocale valide entre les intervalles de temps.

Voici comment cette solution peut être mise en œuvre.

Tout d'abord, vous devez ajouter le capteur d'horloge à l'application. Dans le menu du concepteur, accédez à l'onglet « Capteurs » et faites glisser et déposez le capteur « Horloge » sur l'écran (Figure 22).

Remarque : L'horloge est également un élément non visible, elle n'apparaît donc pas dans l'aperçu de l'écran.

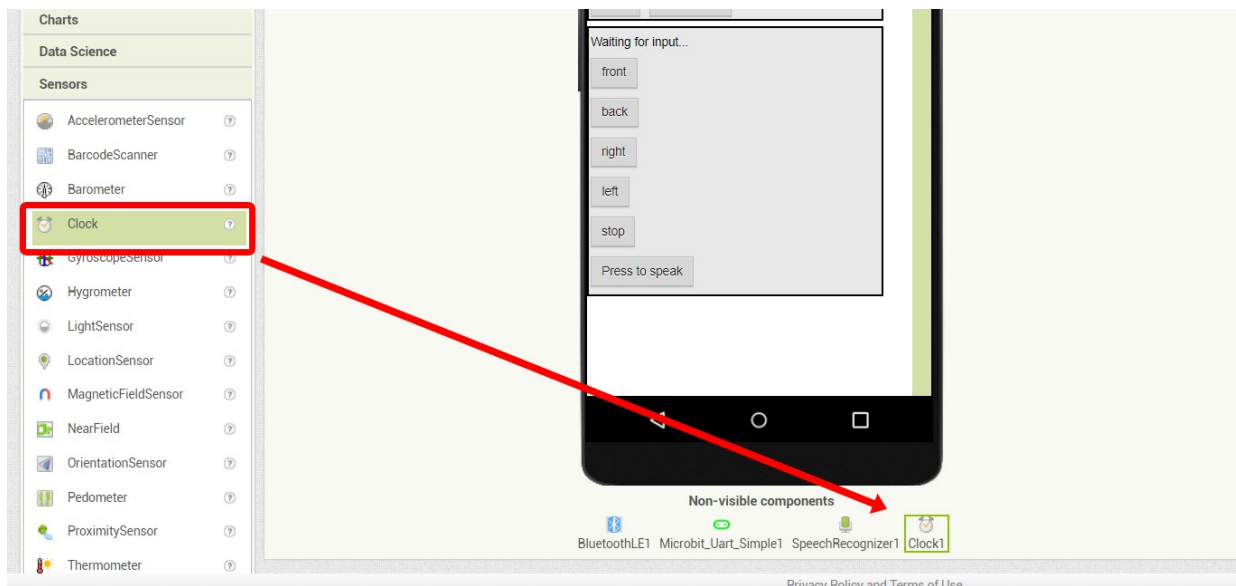


Figure 20: Adding the Clock item to the application

Sélectionnez le composant Clock, dans la liste des composants et dans le menu Propriétés, définissez les paramètres suivants : a) cochez la case sous TimerAlwaysFires pour permettre au minuteur de l'horloge de se déclencher à chaque fois que le compte à rebours se termine, b) dans le champ TimerInterval, définissez la durée du minuteur, en insérant manuellement une valeur numérique en millisecondes (c'est-à-dire 2000 dans l'exemple) (Figure 23).

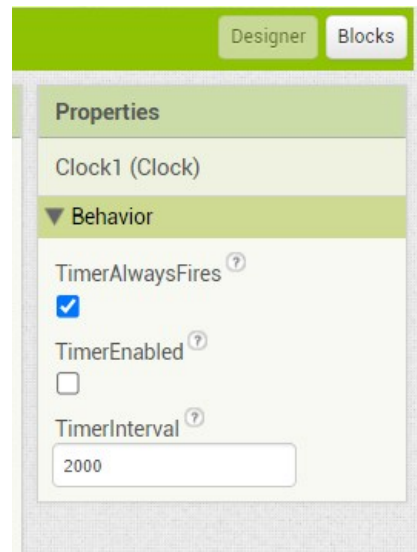


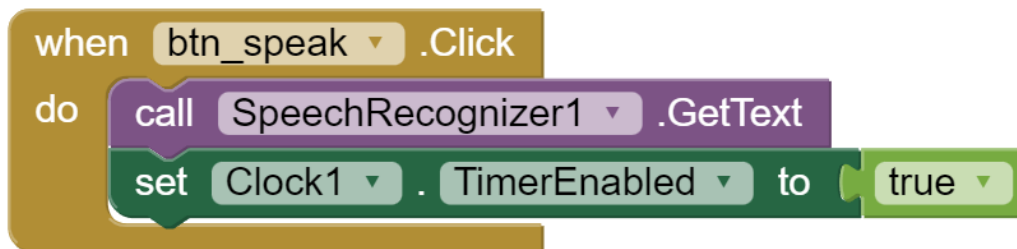
Figure 21: Setting the properties of the clock component

**Note 1 :** la durée d'un intervalle se mesure en millisecondes. 1 seconde équivaut à 1000 millisecondes. Par conséquent, 2 000 ms équivalent à 2 secondes.

**Remarque 2 :** la case sous `TimerEnabled` n'est pas cochée, car nous ne voulons pas que le Timer soit activé lors de l'initialisation de notre application. Nous voulons qu'il soit activé lorsque le bouton « Appuyez pour parler » est enfoncé.

La prochaine étape consiste à programmer ce nouveau composant.

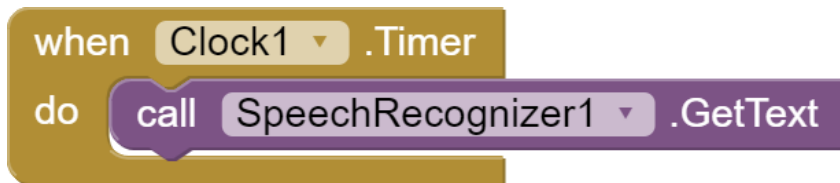
Dans le menu Blocs, ajoutez la commande « `set Clock1.TimerEnabled to` » dans le gestionnaire d'événements « `When btn_speak.Click` » et sous la commande « `call SpeechRecognizer1.GetText` ». Ensuite, indiquez une condition « vraie » sur le côté droit de la commande. Grâce à cette commande, la minuterie de l'horloge est activée après l'activation initiale de `SpeechRecognizer`.



Maintenant que nous avons déclenché/allumé la minuterie pour la première fois (après avoir appuyé sur le bouton « Parler »), l'étape suivante consiste à réactiver le `SpeechRecognizer` à chaque fois que le compte à rebours se termine. La durée de la minuterie est automatiquement renouvelée grâce à ce processus.

Pour ce faire, sélectionnez le composant Clock et dans le menu flottant, sélectionnez le gestionnaire d'événements « When Clock1.Timer ». Ensuite, placez la commande « call SpeechRecognizer1.GetText » dans le gestionnaire.

Grâce à ce script, nous demandons à notre application « d'appeler SpeechRecogniser pour obtenir le texte » de ce qu'elle entend, chaque fois que la minuterie de l'horloge se déclenche.



Explication de l'intégralité du code : Lorsque l'utilisateur clique pour la première fois sur le bouton « parler », le SpeechRecogniser est appelé, puis la minuterie de l'horloge est activée. Lorsque la minuterie est activée, elle se déclenchera toutes les 2 secondes. Chaque fois qu'il se déclenche, le SpeechRecogniser est appelé, en attente de recevoir une entrée audible/verbale (dans notre cas, une commande vocale).

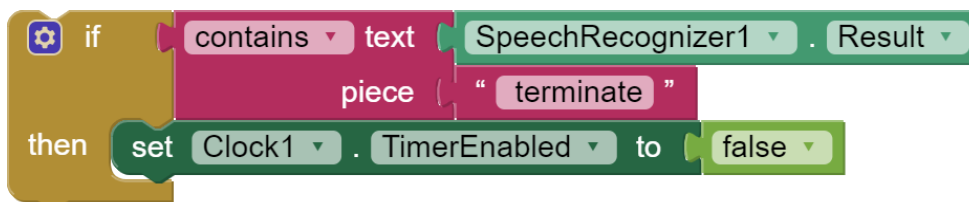
Ainsi, vous avez créé une application qui permet à l'utilisateur de cliquer une seule fois sur le bouton « Appuyez pour parler », au début, puis d'appeler automatiquement SpeechRecogniser toutes les 2 secondes pour vérifier les nouvelles commandes vocales.

Ajout d'une commande supplémentaire pour terminer l'application

En ajoutant le composant Clock, vous avez créé une application plus intuitive. Cependant, l'application fonctionnera indéfiniment. En effet, la minuterie de l'horloge se déclenche indéfiniment et toutes les 2 secondes, créant une boucle infinie dans laquelle SpeechRecogniser recherche constamment des commandes vocales valides.

Pour résoudre ce problème, vous devez ajouter des blocs de code supplémentaires dans l'application. Plus précisément, vous devez ajouter au script de la figure 20 le bloc de commandes suivant.

En particulier, ajoutez une condition supplémentaire « si... alors » à la fin du script. À l'intérieur de la partie if, placez un bloc de commandes qui permettront à SpeechRecogniser d'identifier le morceau de texte « terminer ». Dans la partie then, placez un bloc de commande qui désactivera le capteur Clock, à savoir les blocs « set Clock1.TimerEnabled to » « false ».



Grâce à ces blocs de code supplémentaires, la minuterie de l'horloge sera désactivée lorsque l'utilisateur prononce le mot terminer. Ainsi, la boucle sera terminée et SpeechRecognizer cessera de rechercher des commandes vocales valides.



**Astuce :** En fonction du niveau de vos élèves, vous pouvez les encourager à trouver leur propre solution au problème de la boucle infinie. De cette façon, ils peuvent mieux comprendre le fonctionnement du capteur d'horloge et se sentir plus en confiance dans l'ensemble du processus.

### 3.5.4 Expérience 2

Cette 2ème activité est assez étendue, il faut donc prévoir quelques heures pour aborder tous les concepts et leurs aspects inhérents.

Pour introduire la deuxième idée, en plus d'encourager vos élèves à concevoir des arbres de décision ou des organigrammes pour représenter les actions ou comportements possibles que la voiture robotique peut adopter, vous pouvez également engager un dialogue en posant certaines des questions suivantes :

- Quel type de représentations devez-vous créer pour mieux illustrer les informations codées ?
- Quels chemins ou opérateurs logiques devez-vous adopter pour aider un agent intelligent à entreprendre la meilleure action possible ?
- Quelle est la meilleure façon d'utiliser les résultats de la perception de l'IA ? (par exemple, en vous basant sur la réaction de la voiture robotique, pensez à des cas où une telle technologie aurait une valeur ajoutée)

Grâce à cette activité et à la discussion qui suit, les élèves seront capables de comprendre :

- Comment les arbres de décision et les organigrammes peuvent révéler des chemins logiques possibles, conduisant également à des décisions sur les opérateurs à utiliser
- Comprendre comment un agent intelligent peut adopter le meilleur comportement ou performance possible
- Comprendre comment les agents intelligents peuvent percevoir leur environnement et agir sur les informations entrantes

Pour la conception et la création de l'application, vous pouvez utiliser le fichier « Student\_Worksheet\_for\_Activity\_2.pdf » et guider discrètement vos étudiants tout au long du processus, en leur donnant des conseils si nécessaire. Vous pouvez également les encourager à expérimenter différentes commandes vocales (et donc des morceaux de texte) et à observer les résultats. À cette fin, vous pouvez leur conseiller de réaliser un tableau – comme celui ci-dessous – dans lequel ils enregistreront quelles commandes vocales réussissent et lesquelles ne permettent pas de déplacer la voiture robotique dans une direction donnée.

Commande vocale	Succès	Échec
Move forward	√	
Go to front		√

....		
------	--	--

Pour la solution optimisée (avec l'ajout de l'élément Horloge), vous pouvez encourager vos élèves à définir différentes durées pour l'intervalle de minuterie et à enregistrer leurs observations sur la fonctionnalité de l'application. Vous pouvez également leur proposer de créer un tableau (comme celui ci-dessous) pour organiser leurs observations. De plus, vous pouvez encourager chaque équipe à utiliser un intervalle de temps différent et à comparer les résultats dans des démonstrations parallèles.

Intervalle de minuterie	Observation
500ms	The application is working very fast
2000ms	The application is properly working
....	....

## 3.6 Activité 3 : Introduire l'idée d'apprendre en entraînant un modèle de reconnaissance des commandes vocales

### 3.6.1 Description

Dans cette activité, les étudiants seront initiés à la 3ème grande idée, à savoir l'apprentissage, en entraînant un modèle à reconnaître des commandes vocales spécifiques. Grâce à cette activité, ils comprendront le rôle des algorithmes d'apprentissage automatique (ML) et d'apprentissage automatique pour aider les ordinateurs à apprendre. Ils apprendront notamment à utiliser l'outil Personal Audio Classifier ML pour former un modèle qui classe les commandes vocales selon des critères spécifiques.

### 3.6.2 Utiliser Personal Audio Classifier pour entraîner un modèle

Dans l'activité précédente, vous apprendrez à utiliser le service de reconnaissance vocale pour enregistrer des commandes vocales et les transformer en texte, afin de naviguer dans la voiture robotique. Dans cette activité, vous apprendrez à entraîner un modèle à percevoir un certain nombre de commandes vocales et à les classer selon des critères spécifiques. Pour ce faire, vous utiliserez l'environnement de formation Personal Audio Classifier (PAC) (<https://c1.appinventor.mit.edu/>).

La figure 24 présente l'environnement de formation PAC. Comme vous pouvez le constater, le modèle est vide. Il n'y a pas de catégories, pas d'étiquettes et pas de sons classés. Vous devrez donc créer un certain nombre de catégories. Chacune de ces catégories contiendra plusieurs échantillons audio enregistrés (par exemple, différentes commandes identifiables comme recto, verso, etc.), qui seront représentées par une étiquette commune (par exemple, recto, verso, etc.).

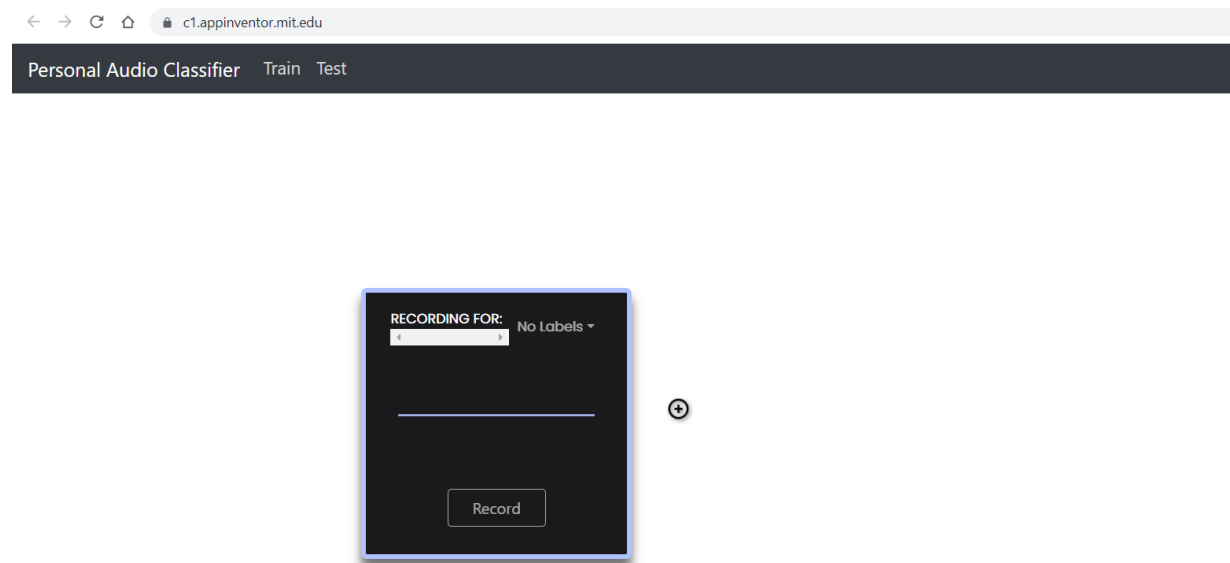


Figure 22: Personal Audio Classifier (PAC) training environment

**Remarque :** Assurez-vous que votre appareil (PC, ordinateur portable, etc.) est équipé d'un microphone/haut-parleur. Sinon, vous ne pourrez pas enregistrer de sons.

La première étape dans la création d'une catégorie de sons communs consiste à créer une nouvelle étiquette. Cliquez sur le bouton +, et dans le menu flottant « Créer une nouvelle étiquette » qui apparaît, saisissez le nom de la catégorie que vous souhaitez créer (c'est-à-dire « front » dans l'exemple illustré à la figure 25a). Appuyez sur la touche « Entrée » et une nouvelle fenêtre apparaîtra avec le titre de l'étiquette que vous avez créée (Figure 25b). Il s'agit de la première catégorie de sons que vous remplirez d'échantillons audio. Pour cela, appuyez sur le bouton Enregistrer (Figure 25b) (1) pour lancer l'enregistrement de l'audio reçu par votre microphone (pour plus d'informations, voir la section « Conseils pour l'enregistrement »). Assurez-vous que vous enregistrez pour la bonne catégorie, en vérifiant le nom de l'étiquette qui apparaît à côté de la section « Enregistrement pour » (Figure 25b) (2).

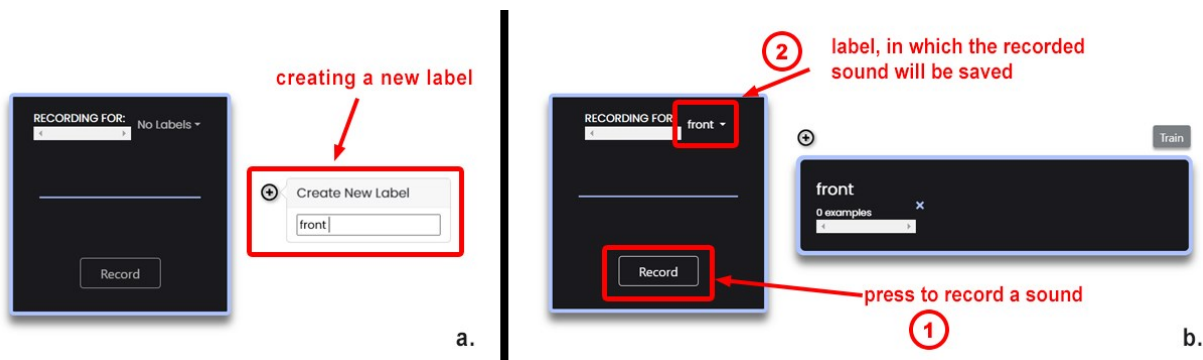


Figure 23: a. Creating a new label; b. Starting recording audio samples to fill the “front” category

Répétez le même processus pour créer toutes les différentes catégories de commandes vocales que vous souhaitez que votre application reconnaisse et que la voiture robotique exécute (par exemple, avant, arrière, droite, etc.).

**Remarque :** Il est important de se souvenir du nom exact de chaque étiquette/catégorie, afin que l'application puisse récupérer ces catégories avec succès. Encouragez vos élèves à créer un tableau dans lequel ils noteront le nom de chaque étiquette et une brève description des sons qui y sont stockés.

Étiquette/Catégorie	Résultat de la commande vocale
front	Move the robotic car forward
clap	Move the robotic car backwards
left	Make the robotic car turn on the left
...	...

### Conseils pour l'enregistrement (Figure 26) :

Lorsque vous appuyez sur le bouton Enregistrer, le processus d'enregistrement est activé pendant environ 1 seconde. Pendant ce temps, votre microphone enregistrera tous les sons pouvant être entendus. En

raison de cette limite de temps, il est recommandé d'enregistrer des commandes vocales courtes (par exemple, « avant » au lieu de « vers l'avant »).

Vous pouvez vérifier l'état de l'enregistrement en regardant la ligne bleue (1) au-dessus du bouton Enregistrer. Si la ligne droite se transforme en forme d'onde, votre microphone fonctionne correctement. Chaque son enregistré est sauvegardé en tant que nouveau fichier audio au sein de la catégorie sélectionnée (« front » dans l'exemple). Si votre microphone a réussi à enregistrer un son, celui-ci sera converti en fichier audio et une icône de spectrogramme coloré apparaîtra (2). Si votre microphone n'a capté aucun son, une boîte blanche apparaîtra, indiquant que le fichier audio enregistré est vide (3). Si vous souhaitez supprimer un fichier audio déjà enregistré, passez votre curseur sur le fichier audio correspondant. Une marque X apparaîtra dans le coin supérieur droit. Cliquez dessus et le fichier audio sera supprimé.

Pour créer un modèle entraîné plutôt fiable, vous devez inclure au moins 5 à 6 exemples/échantillons de fichiers audio dans chaque catégorie. Le nombre de fichiers audio déjà enregistrés est indiqué dans le champ sous le titre principal de chaque catégorie (4). Si vous souhaitez supprimer une catégorie entière, appuyez sur le bouton X (5) à côté du champ « Exemples ».

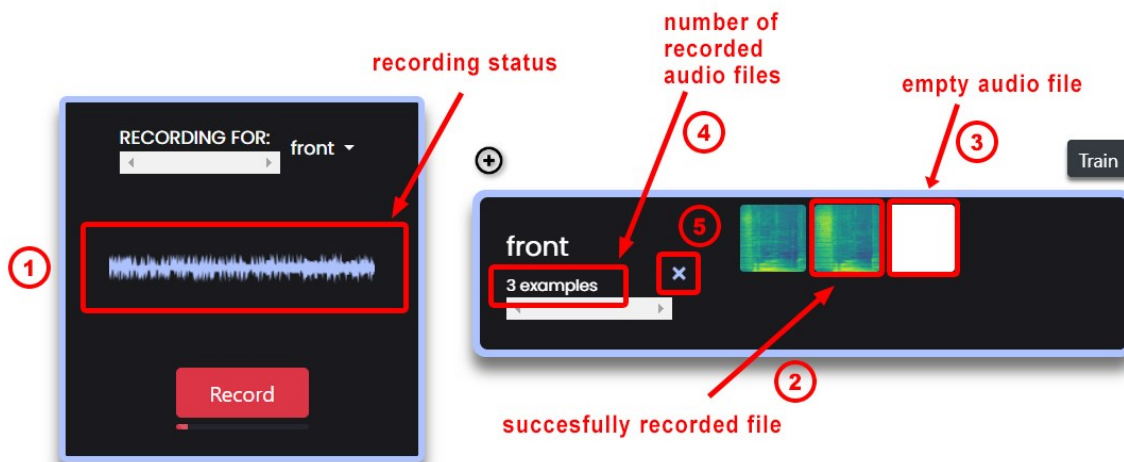


Figure 24: Information regarding the recording process and the recorded audio files

### Entraîner, tester et exporter le modèle

Une fois que toutes les étiquettes/catégories ont été créées, avec au moins 5 à 6 sons enregistrés dans chaque catégorie, vous pouvez procéder à l'entraînement du modèle (Figure 27). Pour ce faire, cliquez sur le bouton Train (1). Une nouvelle fenêtre apparaîtra vous permettant d'ajuster certains paramètres. Conservez les paramètres par défaut et cliquez sur le bouton Train Model (2) pour démarrer le processus de formation.

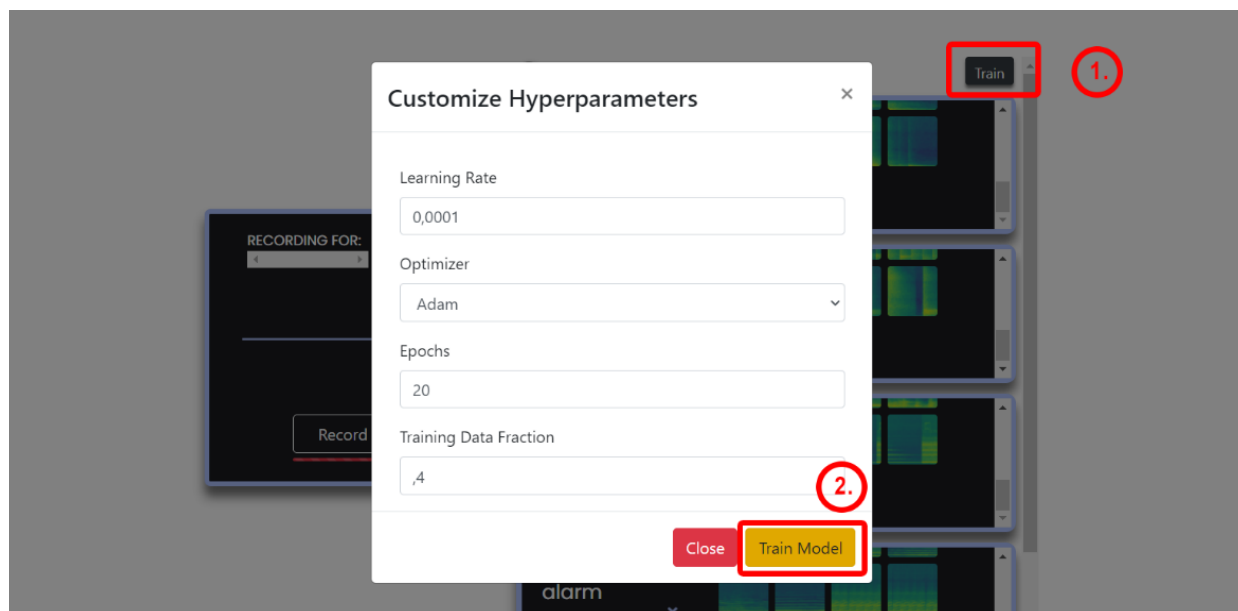


Figure 25: Training the model

Attendez quelques minutes pour que la formation soit terminée. Avant d'exporter le modèle entraîné, il est fortement conseillé de tester le modèle entraîné créé en cliquant sur le bouton Enregistrer (2), en enregistrant un son (par exemple, appuyez sur Enregistrer en disant « retour ») et de voir si le son enregistré est reconnu et classé. dans la catégorie correspondante. Vous pouvez le faire en vérifiant laquelle des catégories de la fenêtre Classification (4) devient verte. Le spectrogramme du son enregistré (3) apparaît également, confirmant que le son entrant a été enregistré avec succès.

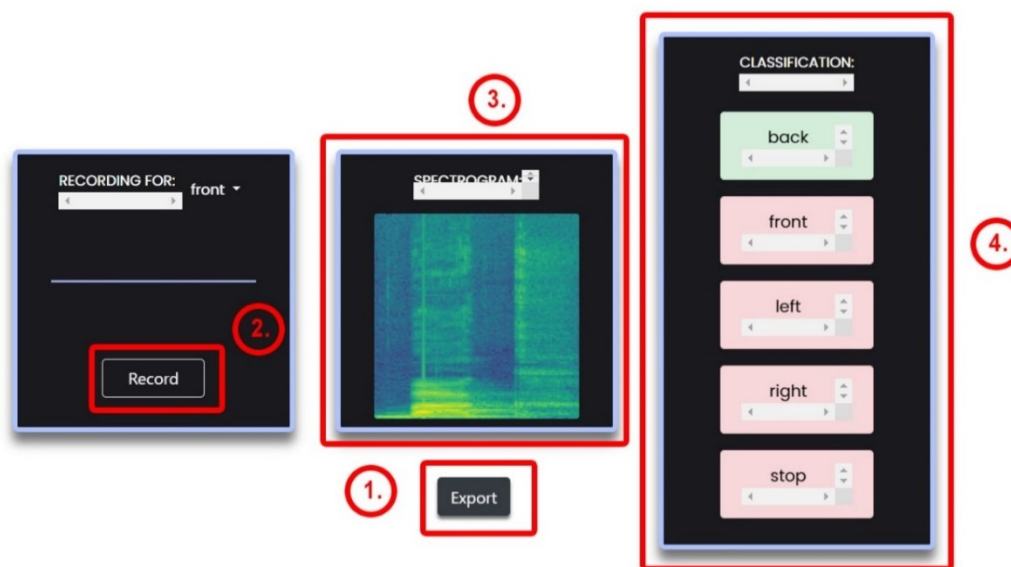


Figure 26: Testing and exporting the trained model

Vous pouvez également faire défiler la page pour voir le niveau de confiance de la prédiction. Par exemple, dans la figure 29, vous êtes informé que le niveau de confiance de la prédiction « retour » est de 38 %.



Figure 27: The confidence level of the prediction for the incoming recording

Encouragez vos élèves à effectuer plusieurs tests avec le modèle formé. Si trop d'erreurs sont trouvées, conseillez-leur de cliquer sur l'onglet « entraîner » (Figure 30) pour apporter des modifications aux données enregistrées.

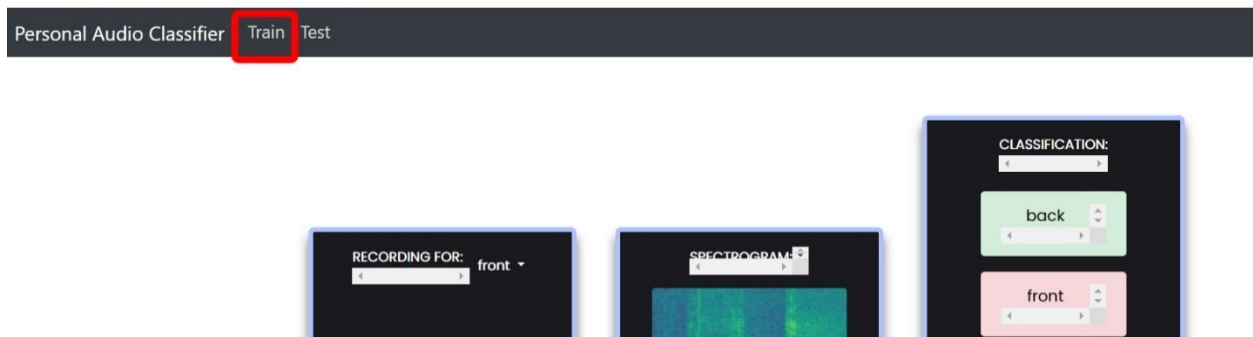


Figure 28: Click on the Train tab to return to the recording mode

Si le modèle entraîné fonctionne correctement, cliquez sur le bouton Exporter (1) (Figure 28) pour enregistrer le modèle entraîné localement sur votre ordinateur, sous forme de fichier .mdl.

**Remarque :** le fichier est automatiquement nommé model.mdl lors de son téléchargement, mais vous pouvez renommer manuellement ce fichier (après le processus de téléchargement) en quelque chose de significatif (par exemple, voice\_commands\_model.mdl, etc.).

**Remarque importante :** malheureusement, la version actuelle de l'environnement Personal Audio Classifier ne permet aucune modification sur le modèle entraîné, après avoir quitté la page. Il faut donc



prévoir suffisamment de temps pour réaliser cette activité pendant une heure d'enseignement/réalisation.

Dans la 4ème activité, vous apprendrez à intégrer ce modèle dans l'application que vous avez déjà conçue dans App Inventor et à utiliser le modèle exporté pour classer les commandes vocales entrantes afin d'ordonner à la voiture robotique de se déplacer en conséquence.

### 3.6.3 Expérience 3

La section précédente a donné quelques conseils sur la façon de présenter cette activité à votre élève. La première chose à faire est de leur expliquer quel est le but de cette activité (c'est-à-dire apprendre à utiliser un outil ML pour former un modèle qui pourra être utilisé par la voiture robotisée à un stade ultérieur). Vous pouvez les encourager à réfléchir à différents scénarios dans lesquels la classification audio pourrait aider la voiture robotique à apprendre de son environnement, mais comme point de départ, encouragez-les à réfléchir à la manière dont ils peuvent entraîner un modèle à classer différentes commandes vocales.

À cette fin, et avant d'ouvrir l'outil Personal Audio Classifier, encouragez-les à noter les commandes vocales qu'ils souhaitent enregistrer et quel est le résultat attendu de chacune de ces commandes vocales (par exemple, ce qu'ils attendent de la voiture robotique). , si la commande vocale est « correcte »). Pour faciliter ce processus, vous pouvez leur conseiller de créer un tableau comme celui ci-dessous et de noter leurs idées.

Commande vocale	Résultat

Demandez-leur ensuite d'ouvrir l'outil Personal Audio Classifier et d'expliquer brièvement comment l'utiliser. Encouragez-les à créer un certain nombre d'étiquettes et à enregistrer des échantillons audio dans chaque catégorie. Voici quelques conseils que vous pouvez leur donner :

- de créer autant d'étiquettes/catégories que de mouvements que la voiture robotisée effectuera
- expérimentez différents accents ou essayez de mettre l'accent sur différentes parties d'un mot enregistré
- notez le nom exact qu'ils donnent à chaque étiquette/catégorie. Ceci est important pour la prochaine activité (c'est-à-dire la 4ème activité)

Une fois qu'ils ont créé toutes les catégories, demandez-leur de former et de tester le modèle. Lors des tests, conseillez-leur de vérifier également le niveau de confiance de chaque prédiction. Vous pouvez les encourager à noter sur un tableau comme celui-ci si la commande vocale enregistrée a été correctement perçue et classée et quel est le niveau de confiance de chaque prédiction.

Son	Classé avec succès	niveau de confiance (%)
	YES / NO	
	YES / NO	
	YES / NO	
	YES / NO	

Lorsqu'ils sont satisfaits du résultat, demandez-leur d'exporter le modèle, le sauvegardant ainsi localement sur leur ordinateur.

Au cours de cette activité, vous pourrez discuter avec eux des différents aspects de ce processus. Pour engager un dialogue avec eux, vous pouvez poser certaines des questions suivantes :

- Que peut faire un outil de ML ?
- Quels paramètres devez-vous prendre en compte lors de la formation de votre modèle ?
- Est-il important d'évaluer un modèle entraîné avant de l'utiliser dans une application d'IA ?
- Comment un modèle formé biaisé, classifiant les commandes vocales, peut-il affecter les voitures sans conducteur ?
- Comment pouvons-nous éviter les modèles formés biaisés ?

Grâce à cette activité, les étudiants apprendront :

- Comment utiliser l'outil Personal Audio Classifier ML, ou similaire à cet outil ML.
- Comment entraîner un modèle basé sur une classification planifiée
- Comment tester et évaluer un modèle entraîné
- Comment les données biaisées peuvent affecter la précision d'un modèle entraîné

## 3.7 Activité 4 : Présentation de l'idée d'interaction naturelle en intégrant un modèle entraîné dans une application d'IA

### 3.7.1 Description

Dans cette 4ème Activité, les étudiants apprendront à intégrer le modèle entraîné, produit dans le cadre de la 3ème Activité, dans l'application créée lors de la 2ème Activité, afin d'observer comment les performances de la voiture robotique peuvent être affectées lorsqu'un Le modèle formé est intégré à l'application. De cette façon, ils prendront conscience de la façon dont les systèmes d'IA peuvent être sujets à des erreurs en raison des limites de l'IA à interagir de manière naturelle.

### 3.7.2 Intégration du modèle entraîné à l'application d'IA

Pour les besoins de cette activité, vous pouvez soit continuer à travailler sur l'application créée lors de l'activité 2, soit utiliser le fichier « Robotic\_car\_SpeechRecognizer.aia ». Alternativement, vous pouvez envisager de créer un nouveau projet .aia, mais il est fortement recommandé de continuer à travailler sur

le fichier précédent, car il serait plus facile pour les étudiants de comparer les différences de performances de la voiture robotique lorsqu'ils utilisent la reconnaissance vocale. et lors de l'utilisation du classificateur audio personnel.

### Ajout de quelques composants supplémentaires

Vous devez d'abord ajouter des composants supplémentaires à l'écran de l'application (dans le menu du concepteur). Plus précisément, vous devez ajouter a) un composant WebView, b) l'extension Personal Audio Classifier et c) une étiquette supplémentaire où les résultats de la classification seront affichés.

**a)** Le composant WebView permet aux applications d'héberger une URL et de rediriger l'utilisateur vers une page Web spécifique. Aux fins de cette activité, ce composant est ajouté pour permettre au classificateur audio personnel de charger la base de données intégrée dans le modèle entraîné.

Pour trouver ce composant, accédez au sous-menu Interface utilisateur, puis faites-le glisser et déposez-le sur l'écran, sous la disposition Disposition horizontale (Figure 31).

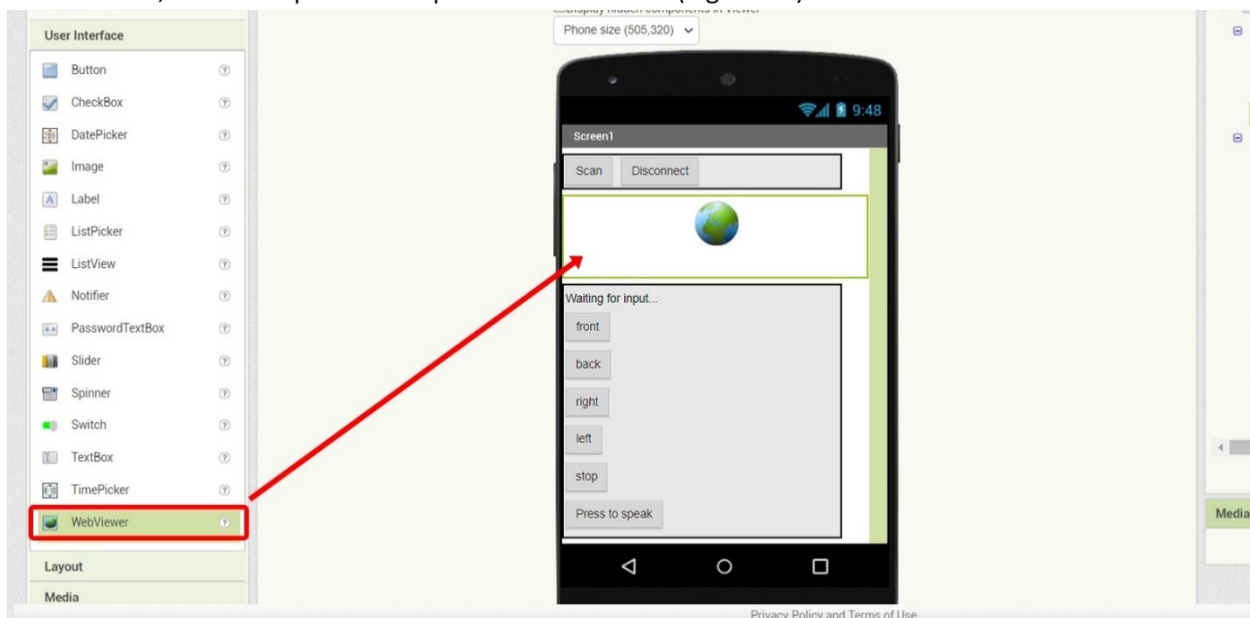


Figure 29: Adding the WebView component

**b)** L'étape suivante consiste à ajouter le composant Personal Audio Classifier, qui permet à l'application d'utiliser le modèle formé précédemment créé par l'environnement de formation PAC. Pour ce faire, vous devez d'abord télécharger cette extension localement sur votre ordinateur. Accédez à ce lien : <https://mit-cml.github.io/extensions/> et enregistrez le fichier .aix correspondant sur votre disque local (Figure 32). Ajoutez ensuite cette extension à l'environnement App Inventor de la même manière que vous avez ajouté les extensions pour Bluetooth et micro:bit.

MIT APP INVENTOR

Home Directory Documentation

Supported:

Name	Description	Author	Version	Download .aix File	Source Code
BluetoothLE	Adds as Bluetooth Low Energy functionality to your applications. See <a href="#">BluetoothLE Documentation and Resources</a> for more information.	MIT App Inventor	20230728	<a href="#">BluetoothLE.aix</a>	<a href="#">Via GitHub</a>
FaceMeshExtension	Estimate face landmarks with this extension.	MIT App Inventor	20210414	<a href="#">Facemesh.aix</a>	<a href="#">Via GitHub</a>
LookExtension	Adds object recognition using a neural network compiled into the extension.	MIT App Inventor	20181124	<a href="#">LookExtension.aix</a>	<a href="#">Via GitHub</a>
Microbit	Communicate with micro:bit devices using Bluetooth low energy (needs BluetoothLE extension above).	MIT App Inventor	20200518	<a href="#">Microbit.aix</a>	<a href="#">Via GitHub</a>
PersonalAudioClassifier	Use your own neural network classifier to recognize sounds with this extension.	MIT App Inventor	20200904	<a href="#">PersonalAudioClassifier.aix</a>	<a href="#">Via GitHub</a>
PersonalImageClassifier	Use your own neural network classifier to recognize images with this extension.	MIT App Inventor	20210315	<a href="#">PersonalImageClassifier.aix</a>	<a href="#">Via GitHub</a>
PosenetExtension	Estimate pose with this extension.	MIT App Inventor	20200226	<a href="#">Posenet.aix</a>	<a href="#">Via GitHub</a>
TeachableMachine	Use vision models trained in TeachableMachine with your device's camera.	MIT App Inventor	1	<a href="#">TeachableMachine.aix</a>	<a href="#">Via GitHub</a>

Figure 30: Download the Personal Audio Classifier extension by clicking on [PersonalAudioClassifier.aix](#)

Après avoir importé l'extension PersonalAudioClassifier, faites un glisser-déposer sur l'écran de l'application conçue. PersonalAudioClassifier est également un composant non visible, il apparaîtra donc dans la section des composants non visibles.

Sélectionnez ensuite le composant PersonalAudioClassifier dans le menu Composants pour modifier ses propriétés (Figure 33). Plus précisément, cliquez sur le champ « Aucun... » sous les propriétés Modèle (1) et WebViewer (2) pour télécharger en conséquence le fichier de modèle formé .mdl (que vous avez précédemment créé sur l'environnement de formation PAC) et pour sélectionner le composant WebViewer1. de la liste flottante (3).

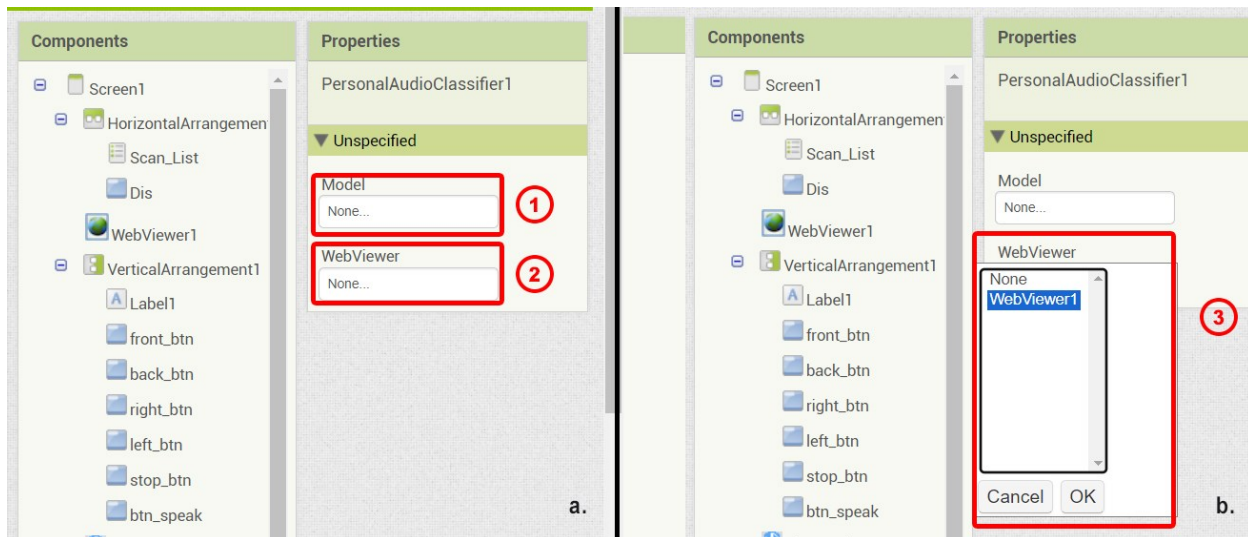


Figure 31: a. Modifying the properties of PersonalAudioClassifier; b. Upload the file of the trained model and select the WebViewer1 component

c) La dernière étape consiste à ajouter une étiquette supplémentaire, qui affichera les résultats du classement, chaque fois qu'un nouveau son est enregistré par l'application.

Faites glisser et déposez un composant d'étiquette sur l'écran, sous le composant WebView et dans le menu Propriétés, modifiez le texte de l'étiquette en « En attente de classification » ou quelque chose de similaire. Vous pouvez également renommer le composant label – à partir du menu Composants – en quelque chose de significatif tel que Classification\_label (Figure 34).

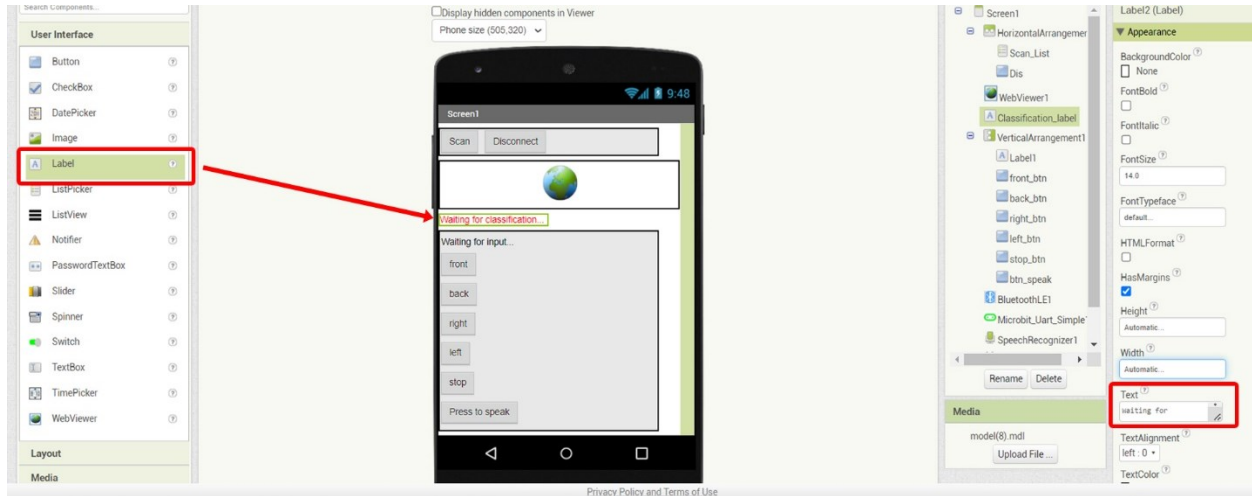


Figure 32: Adding a second label for displaying the classification results

Après cette étape, vous pouvez programmer les nouveaux composants.

### Programmation du composant PersonalAudioClassifier

L'étape suivante consiste à programmer le composant PersonalAudioClassifier et plus particulièrement ses fonctionnalités une fois la classification audio d'un son entrant terminée.

Pour cette étape de programmation, la commande de bloc d'événement « when PersonalAudioClassifier1 .GotClassification...result...do » (2) située dans le menu flottant du composant PersonalAudioClassifier (1), doit être implémentée (Figure 35). Faites glisser et déposez cette commande de bloc d'événements sur la zone où le code est assemblé.

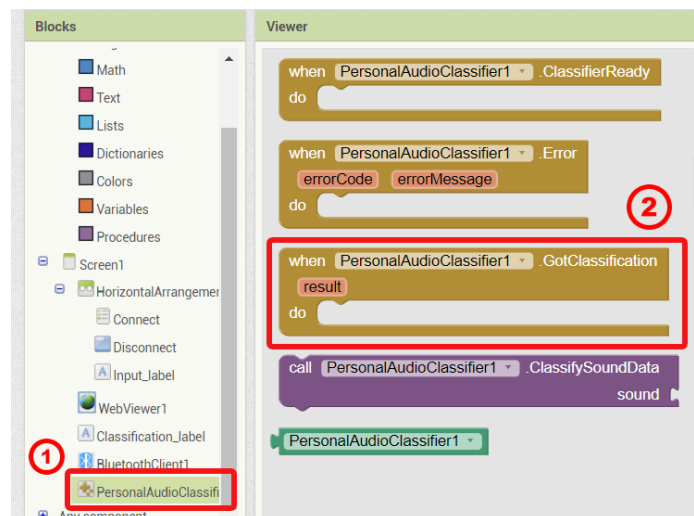


Figure 33: Finding the when PersonalAudioClassifier1 .GotClassification...result ..do.. event block command

Le composant Personal Audio Classifier ajoute un bouton d'enregistrement à l'application. Ce bouton n'est pas visible dans la zone de conception. Il ne sera visible que sur l'appareil intelligent (une fois la conception et la programmation de l'application terminées, et une fois que l'application aura été construite et installée sur un appareil intelligent) (Figure 36). La commande de blocage d'événements ci-dessus détermine ce que l'application doit faire avec les résultats de l'enregistrement, une fois la commande vocale entrante classifiée (c'est-à-dire demander au chat robotique de se déplacer en conséquence).

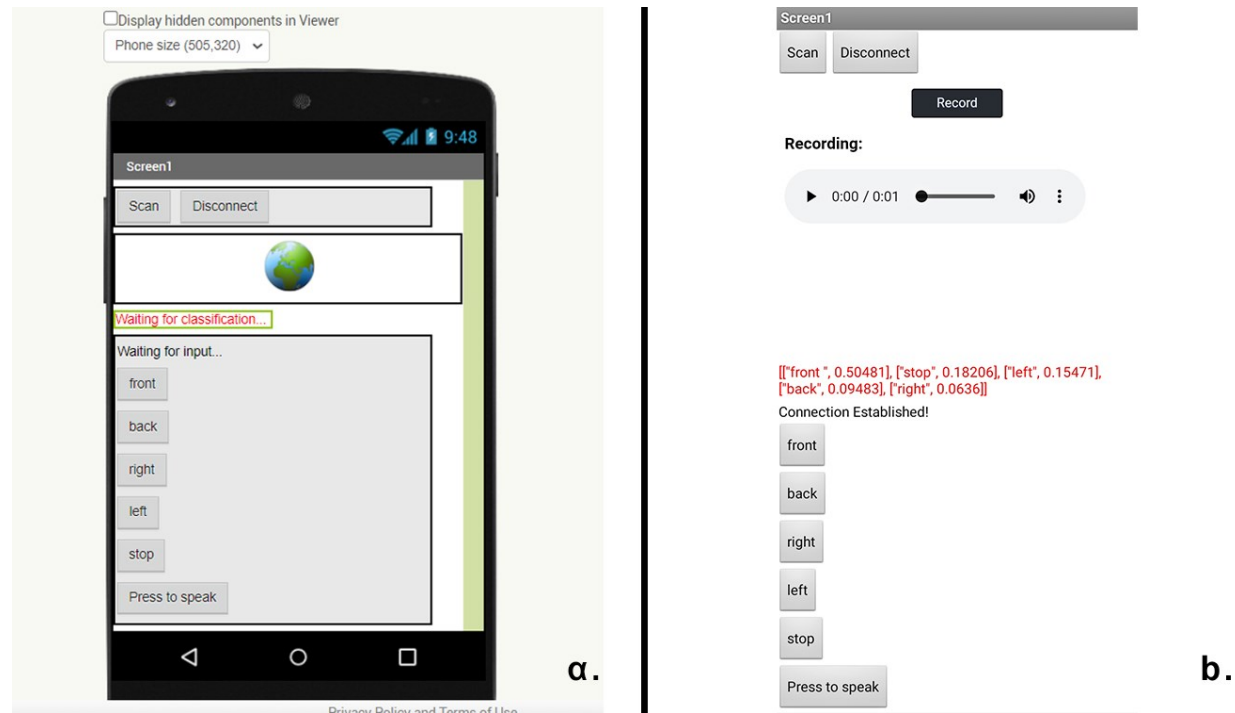


Figure 34: a. the application as appeared in the designer view; b. preview of the application when installed to a smart device

Tout d'abord, vous devez ajouter une commande qui permet au composant Classification\_label d'afficher les résultats de la classification audio. Pour ce faire, sélectionnez le composant Classification\_label (1), et depuis le menu flottant, faites glisser et placez le « set Classification\_label ». Text to » commande de bloc (2) à l'intérieur de la commande du gestionnaire d'événements (Figure 37a). Ensuite, déplacez le curseur sur le champ de résultat (3), puis faites glisser et placez la commande de bloc « get result » (4) sur « set Classification\_label ». Envoyer un message texte à » (Figure 37b).



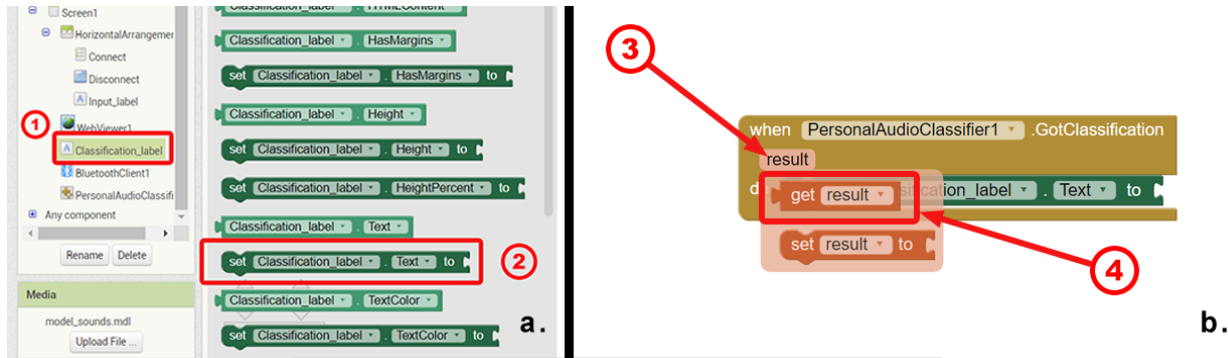


Figure 35: a. Finding the “set... Text to” block command; b. finding the “set result” block command

Grâce à ce processus, le texte « en attente de classification », dans Classification\_label (voir Figure 34), deviendra les résultats de la classification audio, basés sur le modèle de classification que nous avons téléchargé, et suivis du niveau de confiance (par exemple, si la commande vocale avant est enregistrée, alors le texte de l'étiquette pourrait potentiellement changer pour ressembler à celui montré sur la figure 36b, c'est-à-dire « [front, 0,50], [stop, 0,18], [gauche, 0,15], etc. »).

Ensuite, sous le « set Classification\_label ». Text to », placez une condition « if then... else if then... else » (2), située dans le menu Contrôle (1) (Figure 38), pour déterminer ce que l'application fera en fonction des résultats obtenus à partir de la classification. processus.

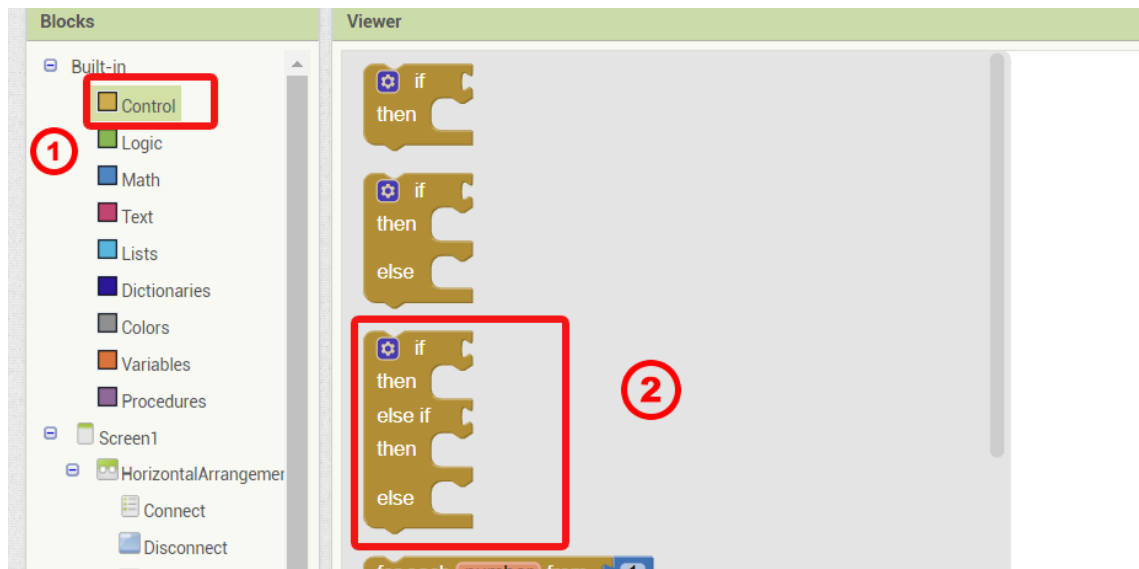
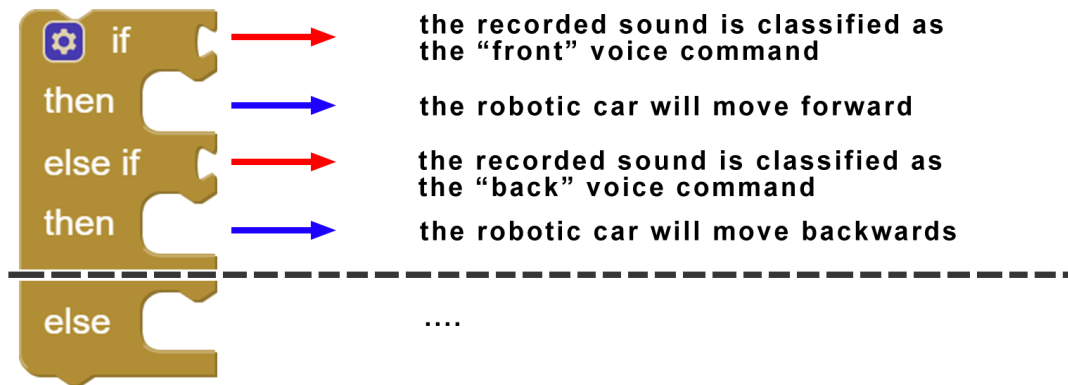


Figure 36: Finding the “if then...else if then...else” condition

Le diagramme suivant présente les paramètres que nous devons insérer dans la condition « si alors... sinon si alors... sinon ».





En particulier, nous avons besoin d'un ensemble de commandes qui reconnaîtront et vérifieront quelle catégorie/étiquette classifiée est appelée. Ces commandes seront placées à l'intérieur de l'instruction « if » ou « else if ». Nous avons également besoin de commandes supplémentaires qui demanderont à la voiture robotique d'effectuer le mouvement approprié, en fonction des résultats de la classification. Ces commandes seront placées dans l'instruction « then ».

Pour vérifier quelle catégorie/étiquette classifiée est appelée, un bloc égal (=) (2) du menu logique (1) (Figure 39a) et un bloc de commande « sélectionner la liste d'éléments de la liste... index... » (4), situé dans le menu Listes (3) (Figure 39b) doit être utilisé.

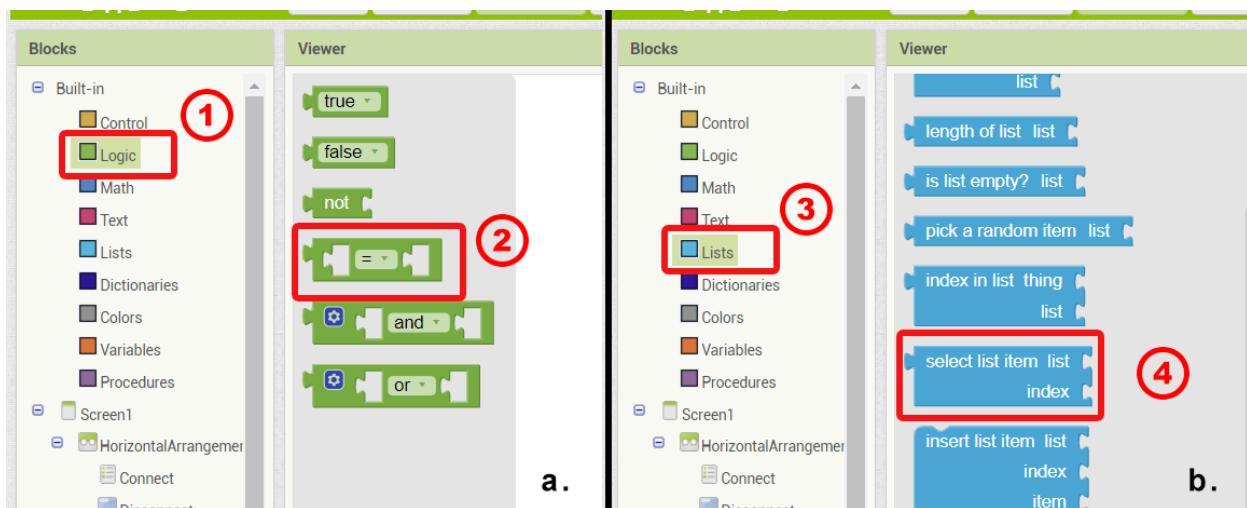


Figure 37: a. Finding the equal (=) block; b. Finding the "select list item list index" block

Tout d'abord, placez le bloc égal (=) à l'intérieur de l'instruction if, comme indiqué dans le bloc de commandes suivant.

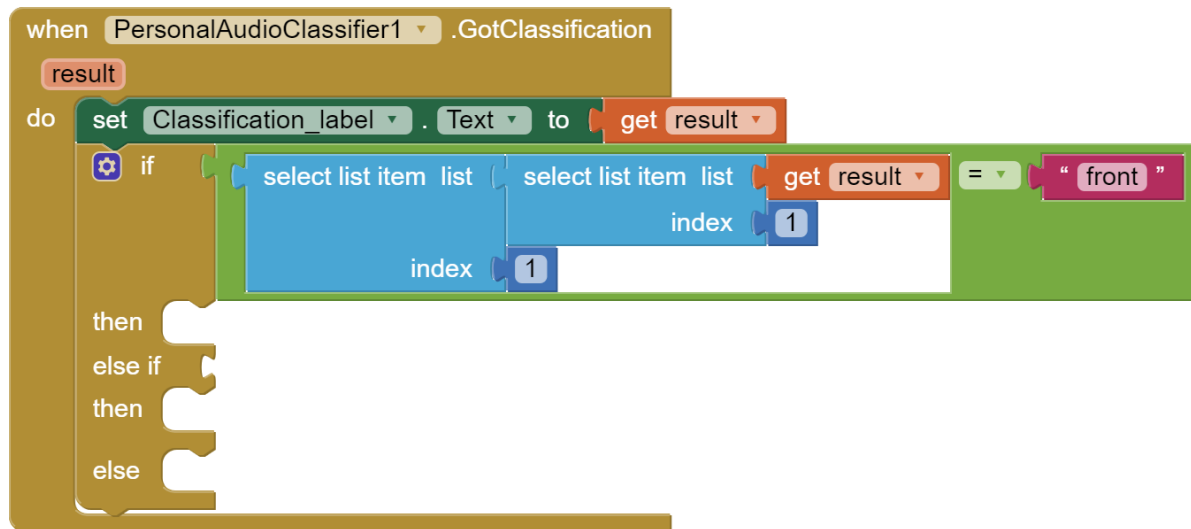


Le résultat renvoyé est en fait une liste des classifications du modèle téléchargé. La liste contient cinq sous-listes. Chaque sous-liste contient l'une des étiquettes/catégories (c'est-à-dire « avant », « arrière », etc.) que le classificateur considère comme une correspondance, suivie du niveau de confiance. Dans l'exemple indicatif susmentionné : [front, 0,50], [stop, 0,18], [left, 0,15], etc., le classificateur est sûr à 50 % que la commande reçue est « avant », à 18 % sûr que la commande reçue est « stop », etc. Nous devons extraire le premier élément de la première sous-liste (« back » dans le cas susmentionné) et tester s'il s'agit de la catégorie à laquelle appartient le son entrant.

Pour ce faire, deux blocs de commandes « select list item list... index... » seront utilisés.



Pour commencer, nous obtiendrons le premier élément (index) du résultat obtenu (notre liste principale). Cet élément deviendra alors notre nouvelle liste principale de laquelle nous extrairons à nouveau le premier élément (index), qui dans notre exemple est « front » (tapé dans un bloc de saisie de texte). Le premier élément de la liste est toujours à l'index 1 et, dans notre cas, est saisi dans un bloc numérique de base.

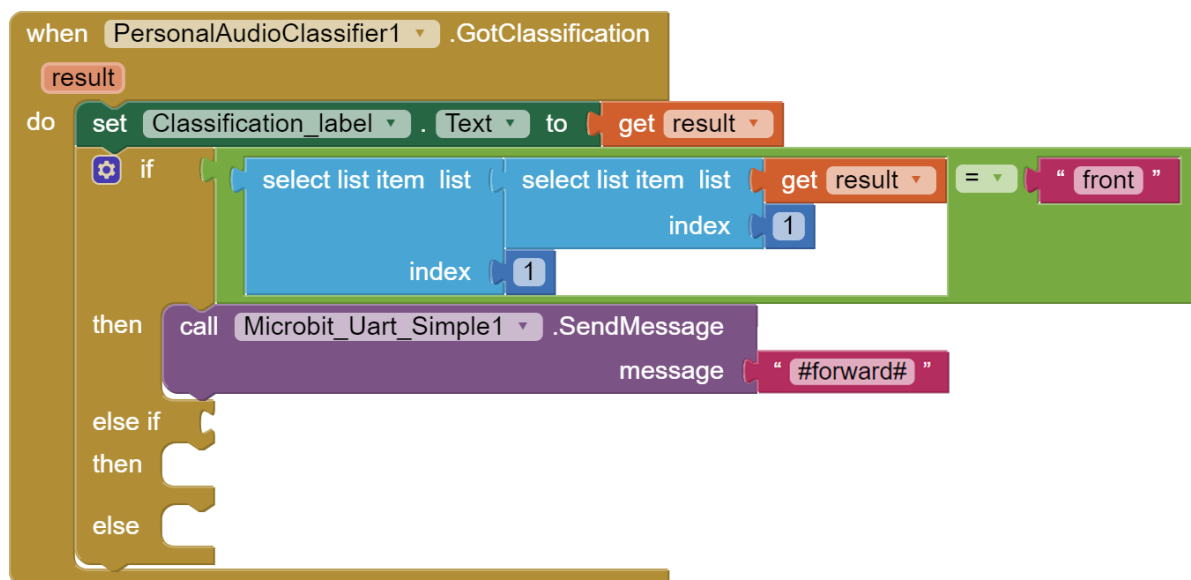


**Remarque :** Le bloc numérique de base se trouve dans le menu Mathématiques, tandis que le bloc de saisie de texte se trouve dans le menu Texte.

L'étape suivante consiste à demander à la voiture robotique d'avancer. Pour ce faire, nous devons demander à l'application de transmettre (via Bluetooth) le message correspondant (c'est-à-dire #forwardt#) à la voiture robotique (c'est-à-dire, comme cela a été déclaré dans le script Makecode).

Par conséquent, à l'intérieur de l'instruction « then », placez un bloc « call Microbit\_Uart\_Simple1 .SendMessage message ». Ensuite, attachez/enclenchez un bloc de saisie de texte « » et tapez le mot « #forward# ».

Voici à quoi ressemblera le code après avoir complété les deux premières instructions de la condition « si alors... sinon si alors... sinon ».



Répétez le même processus pour les autres commandes afin de programmer votre application pour demander à la voiture robotique d'effectuer d'autres mouvements, en fonction du son entrant classifié.

**Remarque :** si vous souhaitez ajouter d'autres conditions « sinon si », cliquez sur l'engrenage bleu (1) à côté de l'instruction « if », et depuis le menu flottant (2), faites glisser et déposez autant de nouvelles conditions « sinon si » que vous le souhaitez. besoin (Figure 40).

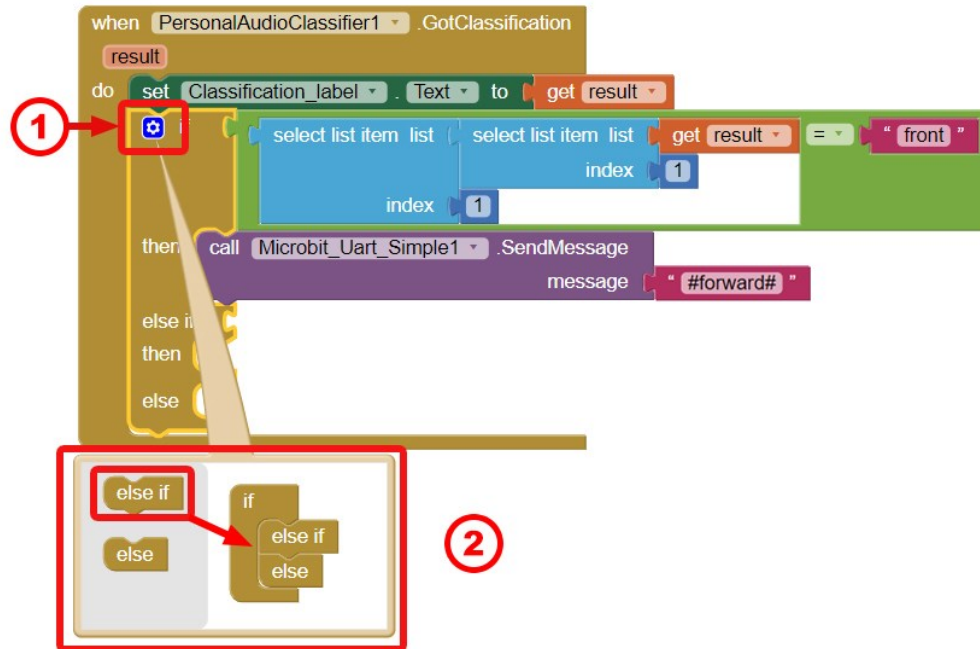


Figure 38: Adding more else if conditions

**Remarque :** À l'intérieur de chaque bloc de saisie de texte, il est crucial a) d'utiliser la même étiquette que celle que vous avez utilisée lors de la production du modèle de classification, b) d'insérer le même message que vous avez déclaré dans le script Makecode.

L'image suivante présente l'intégralité du script pour l'exemple de classification qui a été créée dans le présent document.



Lorsque vous avez terminé toutes les étapes mentionnées ci-dessus, l'application est prête à être téléchargée et installée sur votre appareil intelligent. Accédez au menu Créer et sélectionnez « Application Android (.apk) » dans le menu déroulant pour lancer le processus de production du fichier .apk. Ceci pourrait prendre quelques minutes. Ensuite, installez l'application sur votre appareil intelligent via l'application MIT AI2 Companion.

**Remarque importante :** Dans certains cas (surtout si le modèle entraîné a trop d'étiquettes/catégories), lorsque le modèle entraîné peut ne pas fonctionner correctement une fois intégré à l'application, conduisant à de faux résultats. Si vos élèves rencontrent un tel problème, encouragez-les à expérimenter avec un modèle entraîné plus petit (avec seulement 2 ou 3 étiquettes) et réfléchissez à des scénarios alternatifs dans lesquels ce service d'IA serait plus utile (par exemple, démarrer et arrêter la voiture). Dans le cas d'un modèle entraîné plus simple, le script PersonalAudioClassifier pourrait ressembler à celui présenté dans la figure 41.

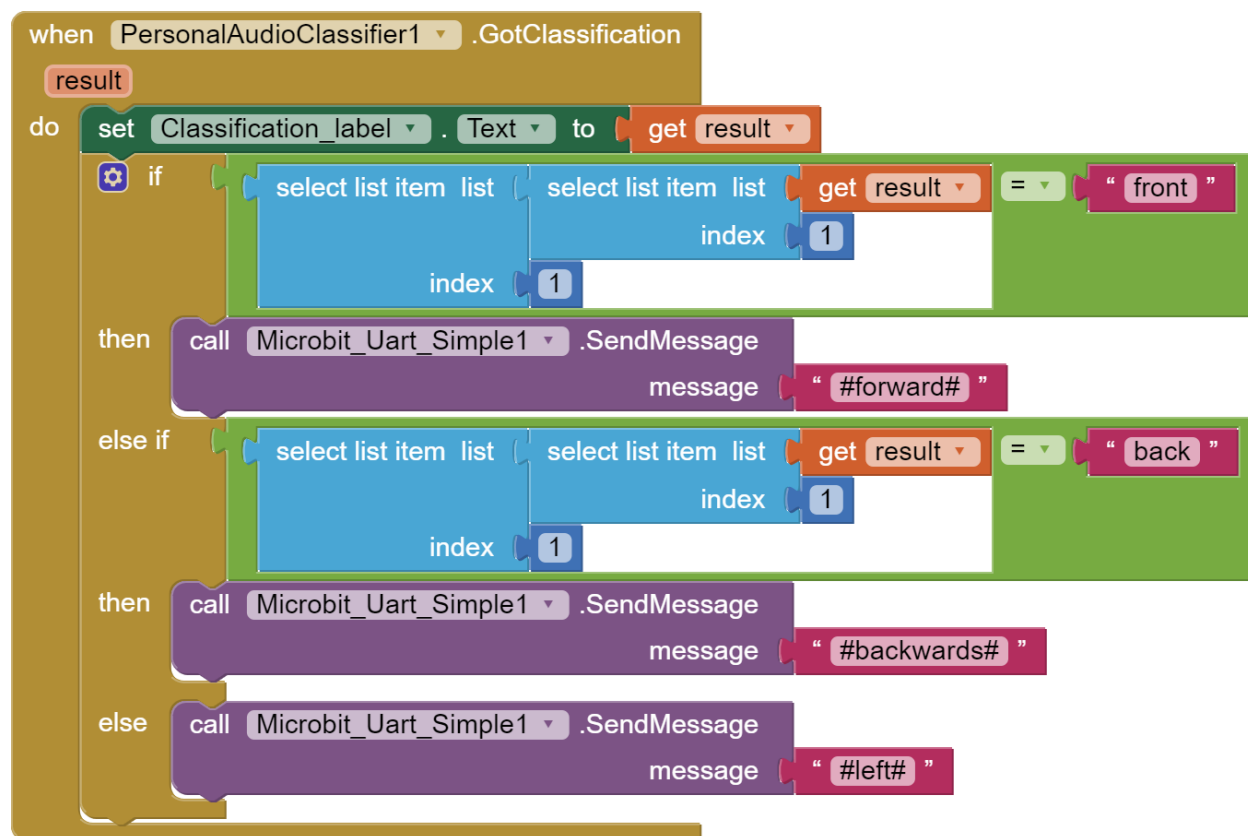


Figure 39: A script for programming Personal Audio Classifier when the trained model contained only two categories, such as front and back.

### 3.7.3 Expérience 4

Pour présenter cette activité en douceur à vos élèves, vous pouvez utiliser la feuille de travail correspondante (Students\_worksheet\_for\_Activity\_4.pdf). Avant de le faire, assurez-vous qu'ils comprennent le but de cette activité (c'est-à-dire évaluer un modèle entraîné en l'intégrant dans une application et découvrir les limites de l'IA).

Une fois que vos élèves ont créé l'application, encouragez chaque équipe à tester l'application créée par d'autres équipes. Encouragez-les ensuite à partager leurs expériences en plénière.

Voici quelques questions que vous pouvez poser pour initier le dialogue :

- Votre application a-t-elle fonctionné correctement ?
- Pensez-vous que des voix différentes ou des prononciations différentes affectent le résultat ?

- Pensez-vous à un paramètre susceptible d'entraîner un dysfonctionnement ?
- Quels sont les risques ou les dangers liés à l'utilisation d'un modèle entraîné contenant des erreurs ?
- Parfois, un modèle entraîné avec trop de catégories peut entraîner des dysfonctionnements. Pouvez-vous penser à un scénario alternatif pour votre voiture robotique où un modèle entraîné plus petit (avec moins d'étiquettes/catégories) serait nécessaire ?

Grâce à cette activité, votre élève comprendra que :

- Beaucoup de travail doit être fait pour qu'un système d'IA interagisse de manière plus naturelle
- Des prononciations ou des tons de voix différents peuvent entraîner des dysfonctionnements
- Il existe de sérieux risques à utiliser un système d'IA dans la vie quotidienne, si l'ensemble de données n'est pas correctement formé ou contient des biais.

## 3.8 Activité 5 : Introduire l'idée d'impact sociétal

### 3.8.1 Description

Dans cette activité, les élèves seront initiés à la 5ème grande idée, à savoir l'impact sociétal, en réfléchissant aux expériences qu'ils ont acquises en réalisant les quatre activités précédentes. Ils seront notamment encouragés à réfléchir aux avantages, aux inconvénients et aux risques inhérents à l'utilisation des services et outils d'IA, ainsi qu'à la surveillance des données et à la prise de décisions fondées sur des ensembles de données spécifiques. Cette activité peut être mise en œuvre séparément ou combinée aux quatre précédentes. Ils prendront ainsi conscience de plusieurs décisions éthiques qui doivent être prises en compte lors de la conception et de l'utilisation des services et technologies d'IA et d'IoT.

Un certain nombre de questions liées aux enjeux éthiques ont déjà été abordées lors d'activités précédentes. Voici quelques exemples supplémentaires pour engager le dialogue dans ce sens :

1) Vous pouvez encourager vos élèves à réfléchir à un scénario dans lequel la voiture robotisée collecte des données plus sensibles (par exemple des images de personnes pour permettre à la voiture de reconnaître les piétons) et transmet ces données au cloud ou à d'autres services.

- Quels sont les avantages et les inconvénients de cette technologie ?
- Quels paramètres doivent être pris en compte concernant la sécurité de ces données ?

2) Encouragez vos élèves à réfléchir à la manière dont un modèle biaisé peut affecter la façon dont un robot intelligent « pense », conduisant à la construction de représentations biaisées du monde. Encouragez-les à imaginer un scénario dans lequel une voiture robotique sans conducteur, utilisée pour sauver des personnes coincées, est entraînée à reconnaître les signaux audio de détresse qu'elle reçoit, mais n'est pas correctement entraînée. Quelles seraient les conséquences de cette déformation de la réalité ?

3) Imaginez un cas où une voiture robotique est entraînée à reconnaître uniquement les voix masculines et uniquement les voix masculines baryton. Comment un modèle formé aussi biaisé peut-il affecter la vie de plusieurs autres personnes ?



### 3.9 Matériel et ressources

Type de Resource	Titre	Sujet	Lien
Pdf file	T2.4_Creating_the_robotic_car	Guidelines for creating the robotic car	
Pdf file	T2.4_WarmUp_activities_for_the_robotic_car	Warm up programming activities for becoming familiar with the commands for the robotic car	
Pdf file	Circuit_card_Activity1	Material for helping students with the creation of the circuit for the 1 <sup>st</sup> activity	
Pdf file	Half_baked_Activity1	Document with a half-baked solution of the code for the 1 <sup>st</sup> activity	
Pdf file	T2.4_App_Inventor_Warm_Up	A warm up activity for becoming familiar with MIT App Inventor and create an application for controlling the robotic car remotely	
.aia file	Remote_control_Microbit	An App Inventor file for using the file that produced in the framework of the App Inventor Warm Up activity, and can be used for the 2 <sup>nd</sup> Activity	
Pdf file	T2.4_Programming_the_robotic_car	A document with guidelines for creating the script in order to program the robotic car to follow the orders received by the designed application	
.aia file	Robotic_Car_SpeechRecognizer	An App Inventor file that can be used for the 4 <sup>th</sup> Activity	
Pdf file	Students_Worksheet_for_Activity_2	Students' worksheet for helping them toward implementing the 2 <sup>nd</sup> Activity	
Pdf file	Students_Worksheet_for_Activity_4	Students' worksheet for helping them toward implementing the 4 <sup>th</sup> Activity	

### 3.10 Le matériel pour la voiture robotique

La figure 42 présente les composants électroniques de base dont vous avez besoin pour créer la voiture robotique. Vous aurez notamment besoin d'un microcontrôleur BBC micro:bit (1), d'un Kitronik Compact Motor Driver (2), d'un support de pile 3AA (ou 4AA) (de préférence avec fils pré-attachés) (3), de 2 motoréducteurs DC ( de préférence avec des fils pré-attachés) (4) et 2 roues (5).

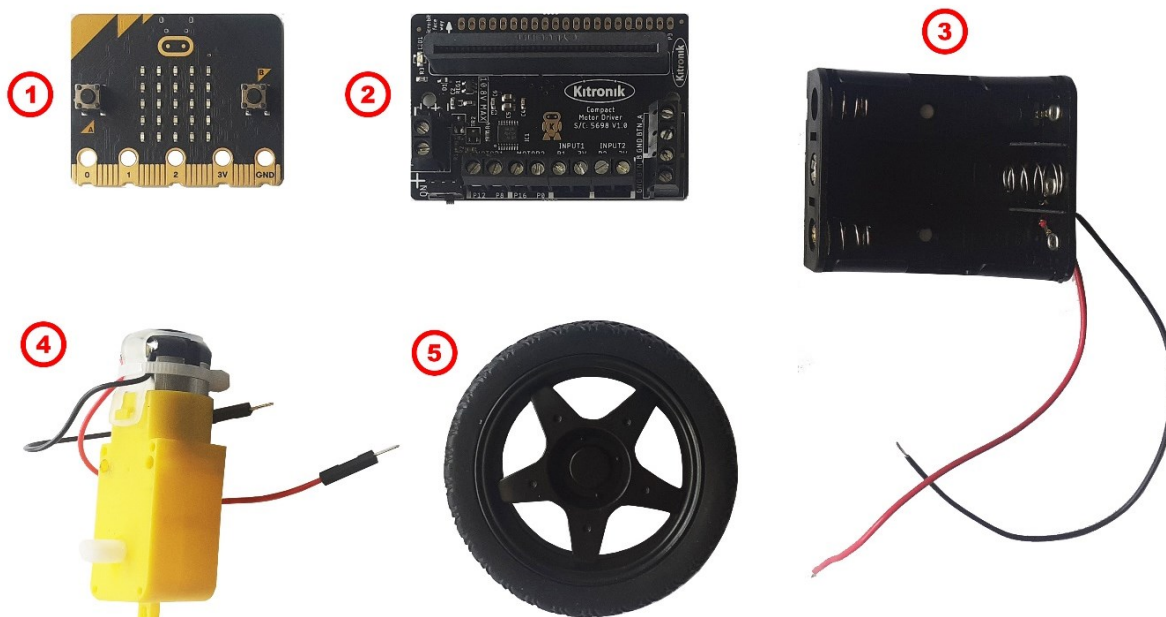


Figure 40: The electronic components needed for creating the robotic car