

Introduction

1. Approche pédagogique

Avec Makeblock Education, le développement des connaissances et des compétences repose sur la combinaison d'outils numériques et physiques qui fonctionnent ensemble pour fournir une expérience d'apprentissage pratique. Les élèves s'engagent activement dans le sujet de la leçon afin de résoudre un problème ou de créer quelque chose de nouveau. Cela leur permet de développer des concepts théoriques à partir d'une expérience pratique, ainsi que de transférer et d'appliquer ces concepts à des tâches pratiques, approfondissant ainsi leur compréhension. Il s'agit notamment de tirer les leçons des erreurs commises en procédant à une analyse détaillée et systématique des processus concernés.

Avec l'apprentissage pratique, les étudiants gèrent et favorisent leur processus d'éducation en jouant un rôle plus actif pendant le cours, plutôt que de se contenter d'écouter un cours magistral délivré par l'enseignant.

Cette introduction donnera un aperçu des nouvelles fonctionnalités de mBot2 et des avantages de son utilisation dans l'éducation. Elle présentera brièvement les étapes pour commencer à programmer le robot avec mBlock5, l'éditeur de blocs de Makeblock Education, et expliquera ensuite les activités (pas de majuscule) de démarrage individuelles. Ces activités constituent une expérience d'apprentissage progressive et couvrent les nouvelles fonctionnalités étape par étape. En partant de références réelles dans les tâches données, les contextes des capteurs, des actionneurs et de la pensée informatique sont abordés, ainsi que des exemples de codes faciles à comprendre et à développer pour chaque tâche. Les activités suggèrent toujours d'impliquer d'autres médias numériques et de penser à l'avenir pour appliquer l'apprentissage à de nouveaux défis et elles se terminent par une phase de réflexion.

2. Introduction à mBot2

mBot2 est un robot éducatif de nouvelle génération conçu pour l'apprentissage de l'informatique et des STEAM (en français Science Technologie Ingénierie Art Mathématiques). Ses capacités étendues en font une solution d'entrée de gamme idéale pour l'enseignement en primaire, collège, lycée et au-delà. mBot2 est conçu pour permettre aux élèves de réaliser des leçons interactives et intelligentes qui sont engageantes, amusantes et reflètent des applications réelles des technologies de pointe.

mBot2 est alimenté par CyberPi, un microcontrôleur puissant et polyvalent. Ses capteurs et actionneurs intégrés (comme un microphone et un haut-parleur, une unité de mesure inertielle avec gyroscope et un capteur d'accélération, un capteur de lumière, des boutons pour les

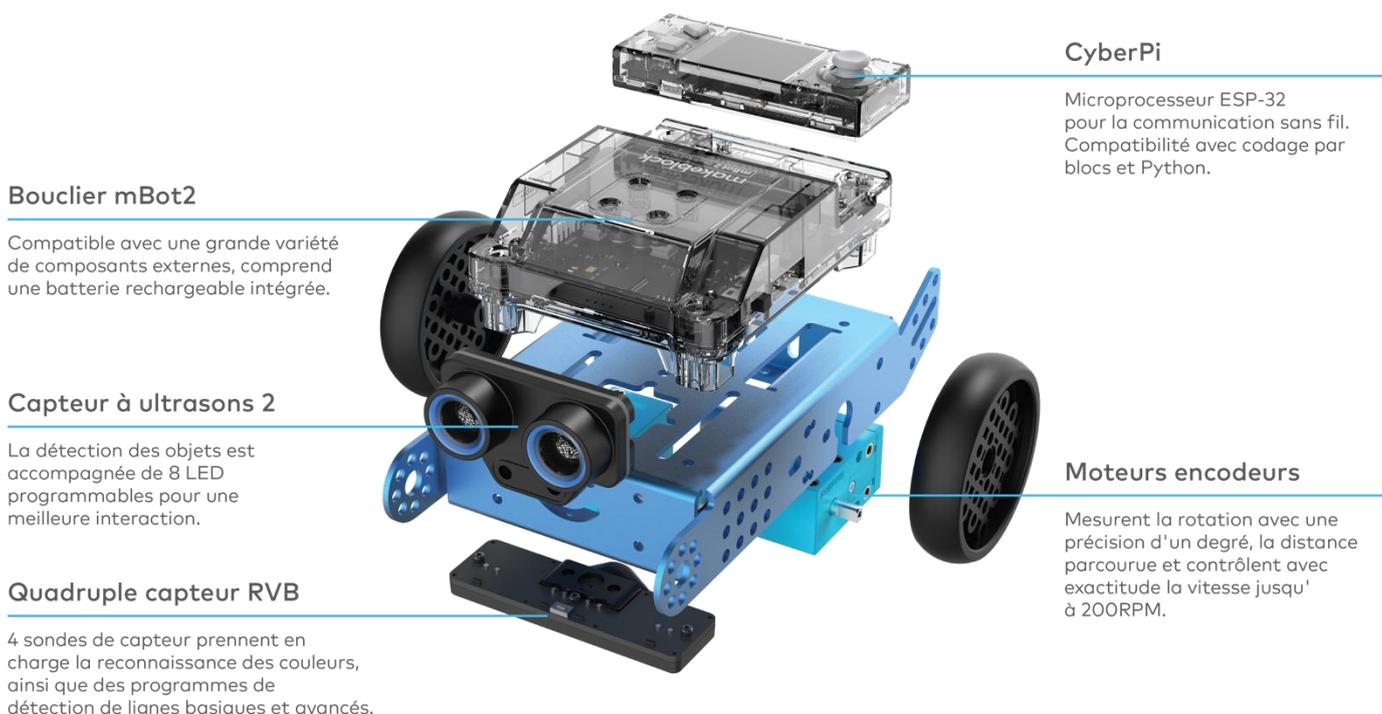
opérations de menu et un écran couleur) sont complétés par un capteur de distance à ultrasons et un capteur de suivi de ligne avec quatre éléments RVB.

Outre cette gamme de capteurs et d'actionneurs, le mBot2 est capable de communiquer en Wi-Fi, ce qui permet un large éventail d'applications sur les sujets du programme scolaire pour le codage, la robotique, le traitement des données et l'intelligence artificielle, en liaison avec d'autres sujets tels que les mathématiques, la physique, etc.

Les enseignants peuvent, par exemple, connecter sans fil plusieurs mBots dans une classe, afin de créer un réseau local de robots qui communiquent entre eux, partagent des informations et exécutent des tâches. Ils peuvent également utiliser un CyberPi autonome comme dispositif intelligent pour communiquer avec le mBot2, créant ainsi un écosystème intelligent ou une télécommande amusante. Lorsqu'il est connecté à Internet, mBot2 peut exécuter des fonctions avancées telles que la reconnaissance vocale ou communiquer avec le cloud pour obtenir des informations.

Les capteurs et actionneurs spécifiques au mBot2 seront présentés dans les sous-sections suivantes et la gamme complète fera partie des activités de démarrage. Vous en trouverez également un aperçu à la fin de ce document.

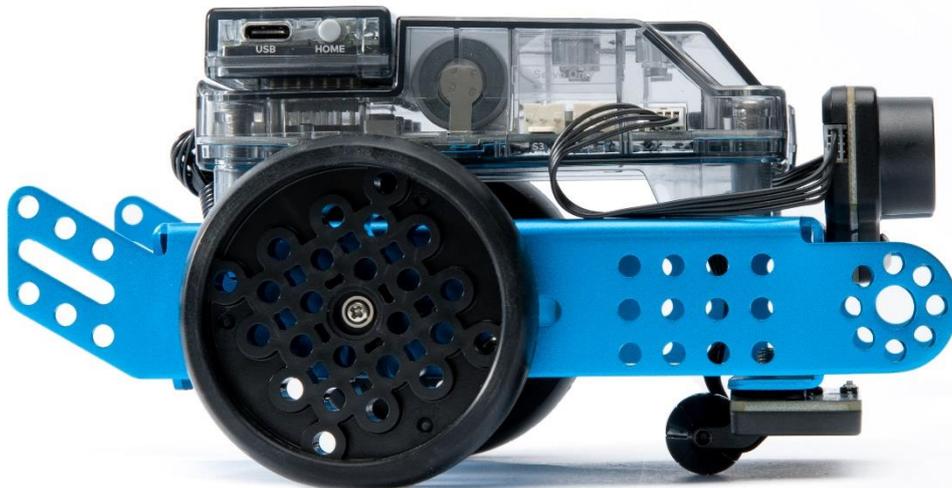
Voici un aperçu des différents composants qui composent le mBot2 :



2.1 Moteurs encodeurs

Les moteurs mBot2 sont équipés d'encodeurs optiques qui permettent un contrôle de haute précision. Grâce à cela, les élèves peuvent contrôler avec précision la rotation, la vitesse et la position des roues et du robot. En outre, les moteurs peuvent également être utilisés comme des servos, voire comme des boutons, pour renvoyer des données au système, tout comme un capteur le ferait.

Grâce à ces caractéristiques, les leçons peuvent être plus réalistes et éducatives par rapport aux robots qui ont moins de contrôle sur leurs moteurs. Les activités peuvent intégrer correctement les concepts mathématiques, par exemple, en parcourant des distances précises, en calculant des virages exacts et même en traçant un itinéraire dans un labyrinthe et en transférant les résultats à l'ordinateur.



La leçon 1 des Activités de démarrage offre une introduction aux différents blocs de codage pour les moteurs codeurs dans l'environnement de programmation mBlock, ainsi qu'une tâche facile permettant aux étudiants d'explorer les possibilités par eux-mêmes.

2.2 Capteurs à ultrasons et quadruple capteur RGB

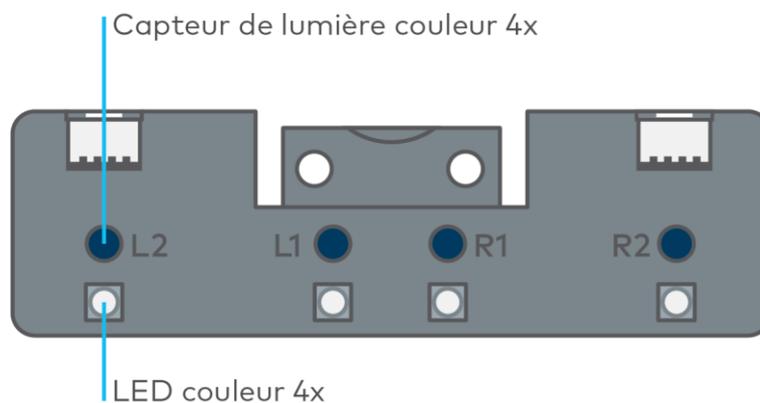
Le capteur à ultrasons fourni avec le mBot2 peut effectuer des relevés plus précis et plus cohérents que ses prédécesseurs. En outre, il intègre des LED bleues qui permettent de nouveaux modes d'interaction avec le robot. Les élèves peuvent utiliser ces lumières supplémentaires pour afficher les réponses.



Le mBot2 intègre un autre composant avancé : le capteur quadruple RVB. Ce capteur a la capacité non seulement d'identifier les couleurs, mais aussi de suivre des lignes afin d'aider le mBot2 à suivre des chemins, ou de détecter des jonctions ou des virages à 90° - tout cela en même temps.

Les fonctions de base peuvent être exécutées avec peu de connaissances préalables, mais en raison de ses caractéristiques avancées, il est important de connaître les meilleures pratiques afin de tirer le meilleur parti du capteur.

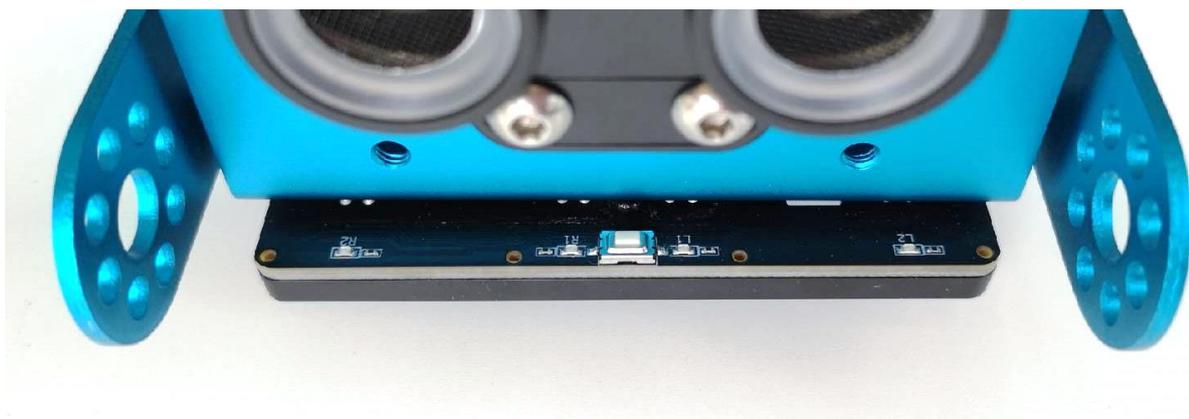
Au lieu d'utiliser 2 capteurs pour suivre une seule ligne devant le robot, le capteur quadruple RVB dispose de 4 capteurs qui lui permettent d'identifier un plus large éventail de conditions : tout en suivant une piste, le capteur peut identifier des carrefours à gauche, à droite ou même des deux côtés, ce qui signifie que le mBot2 peut naviguer dans des cartes plus complexes que les autres robots. Et comme chacun des quatre capteurs est un capteur de couleur, des marques colorées peuvent indiquer des positions sur la carte - le robot « sait » où il se trouve.



Ce faisceau de détection plus large peut être utilisé pour des cartes comportant un réseau de lignes avec plusieurs mBot2 se déplaçant entre différentes stations, comme dans un « entrepôt intelligent », ou pour simuler les routes d'une ville, avec ses différents feux de circulation et ses règles de circulation. Cela s'avérera très pratique dans les concours de robotique pédagogique).

Les capteurs de suivi de ligne précédents utilisaient la lumière infrarouge qui n'est pas visible à l'œil humain. Ces capteurs ne fonctionnent qu'avec une seule longueur d'onde, tandis que le nouveau capteur examine simultanément trois longueurs d'onde différentes, tout comme l'œil humain : le rouge, le vert et le bleu. Un mélange de l'intensité de ces longueurs d'onde est interprété par notre cerveau comme la perception des couleurs. Le capteur fonctionne de manière similaire et, en plus de percevoir les couleurs, il essaie de différencier une piste de l'arrière-plan pour les tâches de suivi de ligne.

Avec la carte fournie, nous avons des marqueurs de couleur supplémentaires à l'intérieur de la piste pour déclencher des actions supplémentaires, si elles sont programmées en conséquence. Pour que le capteur interprète ces couleurs comme faisant partie de la ligne, il doit être calibré sur la couleur la plus brillante de la piste, à savoir le jaune. Placez le capteur sur le code couleur jaune pendant que le mBot2 est allumé et double-cliquez sur le petit bouton situé sur la partie supérieure du capteur (voir l'image ci-dessous). Lorsque les LEDs commencent à clignoter, faites glisser le capteur sur le fond blanc et de couleur jusqu'à ce que le clignotement s'arrête (2-3 secondes). Après cela, la couleur sera considérée comme faisant partie de la piste (pour une étude approfondie, voir les informations supplémentaires sur le capteur quadruple RVB dans différents environnements et la leçon 5).

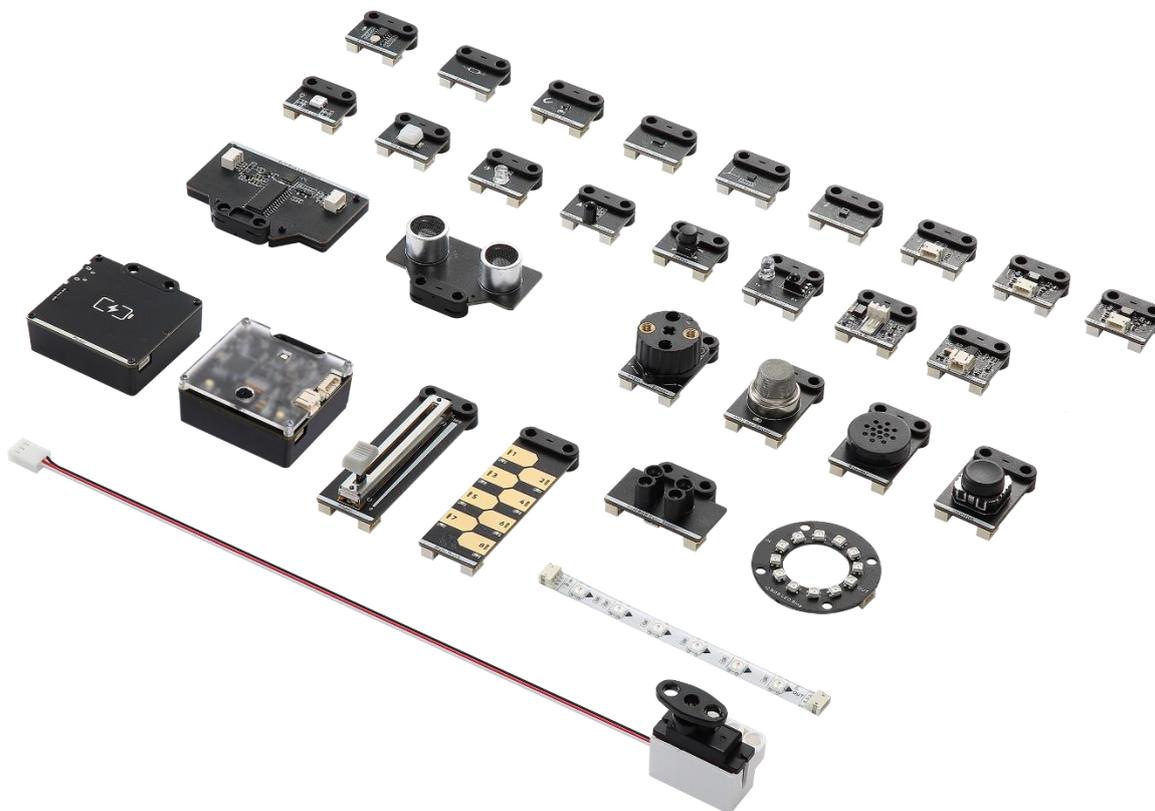


La leçon 5 des activités de démarrage fournit une explication des différents blocs disponibles dans mBlock pour le capteur quadruple RVB, et les étudiants peuvent également mettre en pratique ce qu'ils ont appris avec un simple exercice.

2.3 Plus de possibilités avec mBot2

Le champ des activités peut être encore élargi en combinant le mBot2 avec les modules mBuild développés par Makeblock Education. Grâce à leur unité de microcontrôleur (MCU) interne, ces capteurs et actionneurs intelligents peuvent être directement connectés sans nécessiter de câblage ou de configuration complexe, ce qui permet aux étudiants de consacrer plus de temps à l'idéation et à la création. Ces composants utilisent un seul type de connecteur qui ne peut pas être inversé, de sorte que les élèves ne peuvent pas faire d'erreur en les connectant. Cela leur donne plus de confiance et donc une meilleure expérience d'apprentissage.

Voici quelques exemples de capteurs et d'actionneurs : caméra intelligente, multi-touches, curseur, capteur de température, etc. Avec tous ces capteurs, les enseignants disposent d'une grande variété de choix pour créer des scénarios du monde réel et s'adapter aux besoins de l'enseignement et du programme d'études - ainsi qu'aux centres d'intérêts des élèves.



Mais avec le mBot2 seul, vous avez déjà de nombreuses possibilités (voir les activités). De plus la carte du mBot2 offre des connexions supplémentaires : les interfaces intégrées à 2 et 3 broches peuvent être utilisées pour connecter directement des moteurs à courant continu, des servos, des bandes de LEDs, et même des composants tiers qui sont couramment disponibles, y compris une grande variété de capteurs compatibles avec Arduino. Vous pouvez même en faire des personnalisés.

3. Programmation du mBot2

mBlock est la plateforme de codage de mBot2, conçue pour offrir une expérience éducative améliorée, et un chemin de progression continue pour l'étudiant. Grâce aux extensions de mBlock5, les enseignants peuvent facilement inclure certaines des technologies les plus récentes et les plus influentes dans leurs cours, comme l'Internet des objets ou l'intelligence artificielle. De plus, en intégrant à la fois le codage par blocs et Python, mBlock5 offre un parcours d'apprentissage permettant aux étudiants de développer des compétences informatiques de base et professionnelles.

mBlock5 peut être utilisé comme un logiciel installé sur les ordinateurs, les portables et les appareils mobiles, ou sur un navigateur web. Il est compatible avec différents systèmes d'exploitation tels que Windows, Mac, Linux, Chromebook, iOS, Android. Étant open source, mBlock5 offre la possibilité de créer de nouvelles extensions pour les logiciels et le matériel, permettant ainsi aux enseignants de personnaliser les outils de codage en fonction de leurs

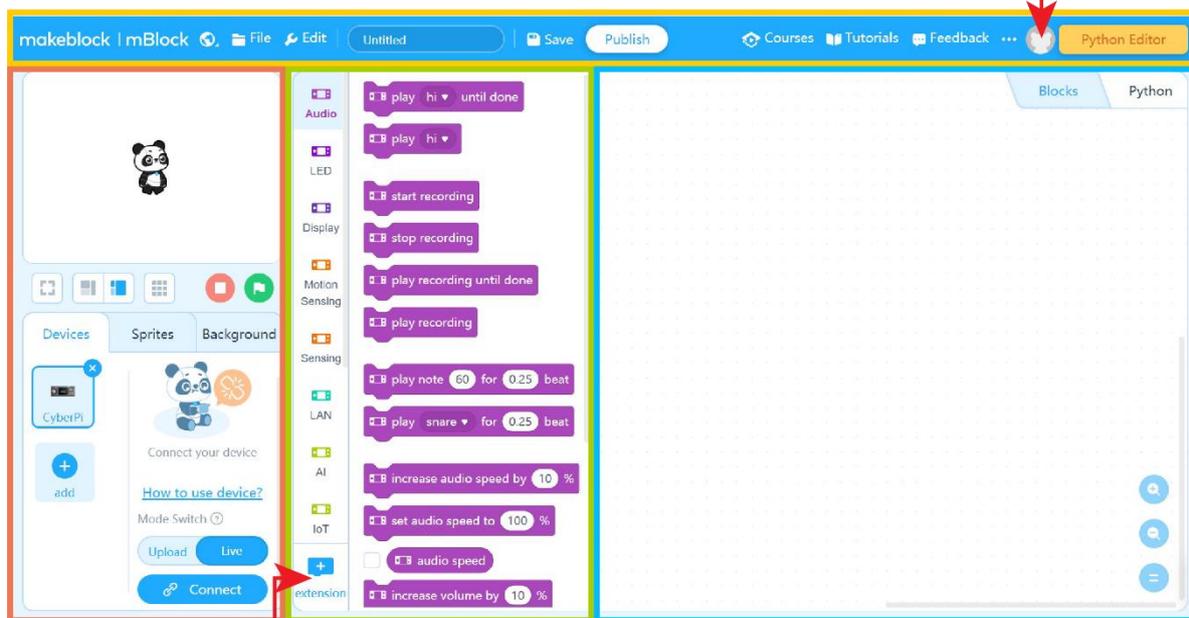
besoins. Les utilisateurs peuvent également rechercher et partager des projets dans la Communauté Math Makeblock.

Commencer à programmer avec mBlock est aussi simple que de glisser et déposer des blocs. L'interface mBlock comporte les éléments suivants :

Barre de menu

Sélectionner la langue, ouvrir ou enregistrer un fichier, trouver des exemples de programmes, etc.

Configurer votre compte pour accéder aux services cloud



Zone de scène

Affichez la scène du projet, personnalisez les sprites et les arrière-plans, connectez les périphériques.

Zone de blocs

Bibliothèque de blocs de codage triés par catégories à code couleur.

Zone de script

Faites glisser les blocs de codage vers cette zone et disposez-les pour contrôler la scène des appareils connectés.

Gestionnaire d'extension

Des informations détaillées sur les caractéristiques et l'utilisation de mBlock peuvent être consultées dans <https://education.makeblock.com/help/category/mblock-block-based/>

4. Aperçu des activités

Les activités de démarrage de mBot2 sont destinées aux élèves de 11 à 14 ans et à leurs enseignants. Dans l'ensemble, les leçons sont conçues pour être accessibles, malgré des niveaux de complexité différents et croissants.

Chaque leçon présente une fonction différente et passionnante du mBot2, ainsi que quelques concepts de base de la programmation. Les connaissances acquises lors de chaque leçon permettent de progresser vers des activités plus avancées, tandis que les leçons gagnent en complexité au fur et à mesure que l'apprenant avance. Par exemple, les mouvements du robot sont présentés dans la leçon 1 et sont utilisés dans de nombreuses leçons suivantes. Les possibilités du capteur quadruple RVB sont présentées dans la leçon 5, pour être ensuite utilisées dans la leçon 8.

Ces leçons sont brièvement décrites ci-dessous :

| Nom de l'activité | Description | Concepts clés |
|--------------------------|---|--|
| 1. Allons-y | Les élèves découvrent le mBot2 et le logiciel mBlock et apprennent à piloter le robot avec précision. Ces connaissances seront utilisées dans la plupart des leçons suivantes. Les élèves concevront également un labyrinthe simple et programmeront le mBot2 pour qu'il s'y déplace (manuellement). | Mouvements précis et blocs de codage correspondants. |
| 2. Détection = données | Les élèves travailleront avec les différents capteurs intégrés dans le mBot2 ; ils apprendront à les utiliser avec les blocs de code correspondants, et à visualiser les données des capteurs sur l'écran couleur intégré. | Mode de fonctionnement des capteurs. Différentes approches pour afficher et visualiser les données sur l'écran. Différences entre le mode Live et le mode Upload dans mBlock 5. |
| 3. Écoutez mBot2 | Les élèves apprendront à contrôler le haut-parleur et le microphone avec les blocs de code de mBlock 5. Ils créeront également un programme dans lequel le mBot2 jouera un son enregistré s'il remplit une certaine condition pendant qu'il se déplace. | Text to speech (TTS) et reconnaissance vocale (Speech to Text, STT) à l'aide du haut-parleur et du microphone intégrés. Exécuter plusieurs tâches simultanément. |
| 4. Voir grâce au son | Les élèves apprendront ce que sont les ultrasons, comment ils sont utilisés dans un capteur, et ils créeront également un programme pour que le mBot2 forme une boucle en se déplaçant après avoir détecté des obstacles sur la route. | Détection d'un obstacle ou d'une portée à l'aide du capteur à ultrasons. Utilisation de boucles et d'instructions conditionnelles pour piloter le mBot2 tout en évitant les obstacles. |
| 5. Visites guidées | Les élèves apprendront comment fonctionne un capteur de couleur, comment ils sont utilisés dans la vie réelle, et ils programmeront le mBot2 pour qu'il devienne un bus touristique qui visite les différents sites d'une ville. Ces connaissances seront également appliquées dans les leçons 7 et 8. | Mode de fonctionnement (physique de la lumière) du capteur de couleur/suiveur de ligne. Identification des couleurs et des lignes - Faire suivre une ligne au mBot2 et lui faire exécuter des actions basées sur la détection des couleurs. |
| 6. Conduite prudente | Les élèves apprendront à utiliser l'accéléromètre gyroscopique du mBot2 et ses blocs de code, et ils programmeront le mBot2 pour qu'il adapte son comportement de conduite s'il détecte des inclinaisons sur la route. | Mode de fonctionnement des gyroscopes et des accéléromètres (en tant qu'unités de mesure inertielle, IMU). Codage du mBot2 pour s'adapter aux conditions de la route en fonction des données IMU. |
| 7. Un jeu en réseau | Les étudiants apprendront à faire communiquer plusieurs mBot2s entre eux sans fil, sans avoir besoin d'un point d'accès WIFI. Ils programmeront un jeu simple où plusieurs mBot2s cherchent une couleur et le premier à la trouver gagne. Ces connaissances seront également utilisées dans la leçon 8. | Transfert de données sans fil dans les réseaux <i>ad hoc</i> . Échange de données dans les boucles et les événements. |
| 8. mBot2 à votre service | En apprenant à établir une connexion WIFI avec le mBot2, les élèves apprendront également à utiliser la reconnaissance vocale embarquée, et ils appliqueront ces connaissances dans une activité où le mBot2 devient un robot serveur qui parle à ses clients. | Utilisation du mode infrastructure WIFI avec le mBot2 pour la reconnaissance et la synthèse vocale. Déchargement des tâches lourdes de l'informatique, comme la reconnaissance vocale, vers des services sur le cloud. Structuration du code en appliquant des « blocs propres » (fonctions). |
| 9. mBot2 dans la nature | Dans cette leçon spéciale, les élèves apprendront certains principes de l'intelligence artificielle en utilisant l'extension Teachable Machine dans mBlock5. Ils appliqueront leurs connaissances pour recréer un écosystème naturel où le mBot2 se comporte comme un animal. | Apprendre l'apprentissage automatique et l'appliquer avec un traitement local uniquement sur la programmation par blocs. Établir un nouveau protocole de communication entre le mBot2 et l'ordinateur. |

Tout au long des leçons, les élèves sont encouragés, avec l'accord des enseignants, à documenter leurs résultats d'apprentissage par vidéo et à les publier, afin d'être fiers de leur travail et de mener leur conversation sur l'informatique et les STEAM en classe.

Leçon 1 : Allons-y !

Sujet : STEAM
Durée : 45 minutes

Classe(s) : CM2 et plus
Difficulté : Débutant

★ Objectifs de la leçon

À la fin de cette leçon, les élèves seront capables de :

- Conduire le mBot2 avec précision.

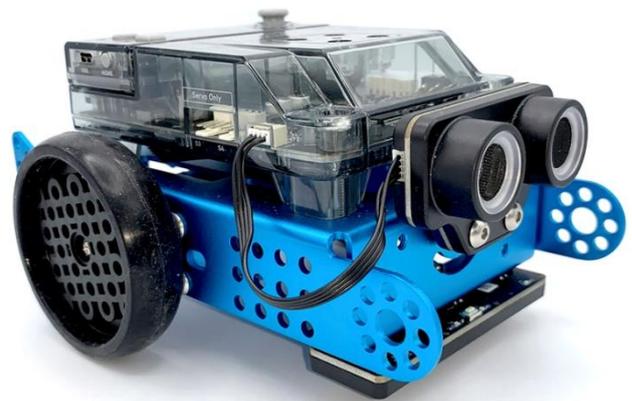
★ Aperçu

Les robots sont des machines autonomes qui remplacent l'effort humain ; ils peuvent percevoir leur environnement et exécuter des programmes informatiques pour prendre des décisions et entreprendre des actions. Un programme informatique est un ensemble d'instructions et de conditions exécutées par un ordinateur pour réaliser une tâche. Vous pensez peut-être n'avoir jamais vu ou utilisé un robot, mais en fait ils sont bien plus répandus qu'on ne le pense. Les robots peuvent prendre diverses formes et avoir des capacités différentes en fonction de leur contexte. Par exemple, les robots sont présents dans les foyers, comme le robot aspirateur ou le robot tondeuse à gazon ; vous pouvez également les trouver dans les usines qui assemblent des produits. Nous avons même envoyé des robots sur d'autres planètes.

🔗 Focus

À la fin de cette leçon, les élèves sauront :

- Quels mouvements peut faire le mBot2
- Quels sont les différents blocs de programmation que vous pouvez utiliser pour faire bouger le mBot2



Liste de contrôle avant la leçon

De quoi avez-vous besoin ?

- PC ou ordinateur portable (avec sortie USB) avec le logiciel mBlock installé, la version web (également pour Chromebook), ou une tablette avec l'application mBlock installée
- mBot2 avec un CyberPi
- Câble USB-C ou un dongle Bluetooth Makeblock
- Papier A3
- Stylos

Plan de cours

| Durée | Contenu |
|------------|--|
| 5 minutes | 1. Introduction <ul style="list-style-type: none"> • Capteurs et données dans la vie quotidienne. • Qu'est-ce que le CyberPi ? |
| 10 minutes | 2. Travaux pratiques <ul style="list-style-type: none"> • Se familiariser avec les différents blocs de programmation du mBot2. |
| 25 minutes | 3. Test <ul style="list-style-type: none"> • Programmation de votre propre robot à travers un labyrinthe que vous aurez créé. |
| 5 minutes | 4. Récapitulatif <ul style="list-style-type: none"> • Showtime : montrez ce que vous avez fait avec votre robot dans un film court et amusant pour en discuter plus tard. • Si votre enseignant le permet, partagez le résultat final sur les médias sociaux avec le hashtag #mBot2inclass • Réflexion : De quoi êtes-vous le plus fier ? Qu'est-ce que vous aimeriez améliorer dans votre robot ? |

1. Échauffement (5 min)

Étape 1 : Échauffement

Cette étape se compose de deux parties :

1. Les robots dans la vie quotidienne
2. Apprendre à connaître le mBot2

1. Les robots dans la vie quotidienne

Nous avons des robots dans de nombreux endroits différents. Certaines personnes en ont chez elles et d'autres travaillent avec eux tous les jours. Les robots peuvent prendre de nombreuses formes différentes et avoir des capacités différentes en fonction de leur contexte. Pouvez-vous citer trois robots que vous rencontrez (régulièrement) dans la vie quotidienne ?



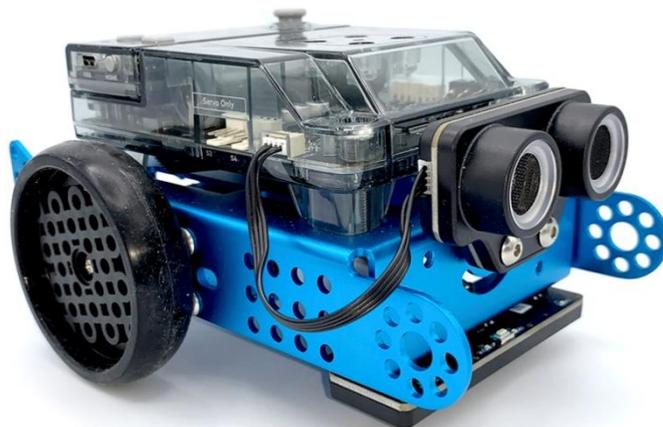
Il y a sans doute beaucoup d'autres robots qui ont été développés et que vous n'avez jamais vus. Recherchez sur Internet trois autres robots que vous pouvez utiliser dans la vie quotidienne.

Il y a de fortes chances que vous côtoyiez des robots plus tard dans votre vie professionnelle ou personnelle. De nombreuses entreprises différentes utilisent les robots. En général, ces robots font des tâches répétitives, qui doivent être effectuées avec une grande précision ou qui sont dangereuses. Dans cette leçon, nous allons principalement travailler sur la programmation d'un robot pour qu'il se déplace avec précision. Pouvez-vous citer trois professions ou entreprises pour lesquelles il est très important qu'un robot travaille de manière très précise ?

2. Apprendre à connaître le mBot2

Le mBot2 est un robot programmable équipé de différents composants qui lui permettent de détecter, d'agir et de communiquer avec son environnement. Le mBot2 est permis aux élèves de s'initier à l'informatique et à la technologie, y compris à la manière dont on peut utiliser un robot dans la vie réelle.

Le mBot2 peut être programmé à l'aide de mBlock. Dans mBlock, la programmation peut être aussi simple que de faire glisser et assembler des blocs. Comme vous le verrez dans cette leçon, vous pouvez, par exemple, programmer le mBot2 pour qu'il traverse un labyrinthe enlever le tiret avec votre aide (et non de manière autonome). Le mBot2 est équipé d'une paire de moteurs spéciaux, qui peuvent enregistrer la rotation de l'axe et donc la vitesse et la distance parcourue par le robot. Ils sont appelés moteurs encodeurs en raison du capteur intégré (encodeur). Ce type de moteurs permet de contrôler des paramètres spécifiques tels que l'angle de rotation et la vitesse des moteurs. Vous apprendrez comment utiliser ces moteurs dans les étapes suivantes.



2. Travaux pratiques (10 min)

Étape 2 : Travaux pratiques

Cette étape se compose de deux parties :

1. Se familiariser avec les différents blocs de programmation du mBot2.
2. Recréer et tester quelques exemples de programmation pour contrôler le mBot2.

1. Se familiariser avec les différents blocs de programmation du mBot2

Comme vous l'avez observé précédemment, les robots peuvent être utilisés de nombreuses manières différentes. Si vous avez réfléchi à un objectif à atteindre avec votre robot, cela signifie très probablement que vous savez ce que le robot doit faire pour vous aider, et parfois ces tâches doivent être exécutées avec une certaine précision.

Lorsque vous commencerez à programmer le mBot2, vous remarquerez qu'il existe de nombreux blocs de codage différents que vous pouvez utiliser pour que le robot se déplace. Vous trouverez ces blocs dans mBlock 5 dans la catégorie "chassis". Ces blocs de codage sont bleus.

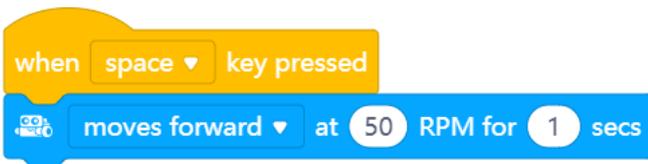


Vous pouvez voir ci-dessous quelques exemples des blocs de codage nécessaires pour faire bouger le robot. Certains d'entre eux se ressemblent, mais ils font tous quelque chose de différent. Pour cette leçon, nous allons travailler en mode En direct. Assurez-vous que le mode correct est sélectionné dans le logiciel. Les différences entre le mode En direct et le mode Téléverser seront expliquées dans la leçon 2.

Bloc de code :



Ce bloc de code vous permet de déplacer le mBot2 en avant, en arrière, à gauche et à droite à une vitesse de rotation spécifique des roues et pendant un certain nombre de secondes. L'exemple ci-dessous montre comment vous pouvez faire bouger le mBot2 pendant une seconde à une vitesse de 50 rotations par minute. Ceci est utile, par exemple, lorsque le mBot2 doit pousser une charge en avant.

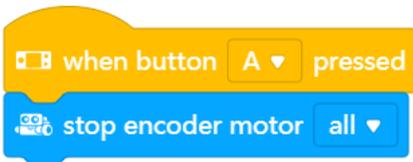


Bloc de code :



Ce bloc de code vous permet de faire en sorte que votre robot s'arrête de bouger. Cela peut être un blocage utile pendant une phase de test. Si un programme ne fonctionne pas comme prévu, vous pouvez utiliser ce bloc pour que le robot s'arrête immédiatement.

Par exemple, vous pouvez faire en sorte que le robot s'arrête lorsque vous appuyez sur le bouton A du CyberPi. Pour ce faire, utilisez l'exemple de programmation ci-dessous.



Bloc de code :



Avec ce bloc de code, vous pouvez faire tourner le mBot2 d'un certain nombre de degrés, et vous pouvez choisir si la rotation doit être vers la gauche ou vers la droite.

Lorsque l'exemple de programmation ci-dessous est défini, le mBot2 peut être contrôlé à l'aide des touches fléchées. En appuyant sur la flèche droite, le mBot2 tournera de 90 degrés vers la droite, en appuyant sur la flèche gauche, il tournera de 90 degrés vers la gauche.

when left arrow key pressed
turns left 90° until done

when right arrow key pressed
turns right 90° until done

Bloc de code :

moves forward 100 cm until done

Avec ce bloc de code, vous pouvez faire avancer ou reculer le mBot2 d'une certaine distance. Lorsque vous utilisez l'exemple de programmation ci-dessous, le mBot2 se déplacera de 100 cm en avant. Pour commencer, déplacez le joystick du CyberPi vers le haut.

when joystick pulled up
moves forward 100 cm until done

Bloc de code :

encoder motor EM1 rotates at 50 RPM, encoder motor EM2 rotates at -50 RPM

Vous pouvez également contrôler les moteurs de la mBot2 indépendamment, et c'est l'un des blocs que vous pouvez utiliser pour cela. Dans l'exemple ci-dessous, le mBot2 se déplacera dans une courbe pendant 3 secondes avant de s'arrêter. Remarquez que la valeur de l'un des moteurs est négative, car les moteurs sont montés dans des directions opposées. Pour déplacer le robot dans une direction, vous devez donc faire tourner l'un des moteurs dans le sens opposé. Que se passe-t-il si les deux roues tournent à 40 tours par minute ?

when button B pressed
encoder motor EM1 rotates at 50 RPM, encoder motor EM2 rotates at -30 RPM
wait 3 seconds
stop encoder motor all

Bloc de code :

encoder motor (1) EM1 rotated angle (°)

Comme les moteurs de mBot2 peuvent mesurer leur vitesse et leur rotation, vous pouvez également lire ces valeurs. La vitesse mesurée est la vitesse de rotation en tours par minute ($360^\circ = 1$ tour ; en science, il est plus courant d'utiliser « degré par seconde » ($1 \text{ RPM} = 6^\circ$ par seconde)). Dans le mBlock 5, si vous cochez la case à côté de ce bloc, vous pouvez lire les

valeurs sur la scène au-dessus du panda. L'exemple de programmation ci-dessous remet les valeurs de la rotation à zéro, puis ordonne au mBot2 d'avancer pendant une seconde. Quelle valeur lisez-vous sur le plateau après le déplacement du robot ?

```

when joystick middle pressed
  reset encoder motor all rotated angle
  moves forward at 50 RPM for 1 secs
  
```

2. Recréer et tester quelques exemples de programmation pour contrôler le mBot2.

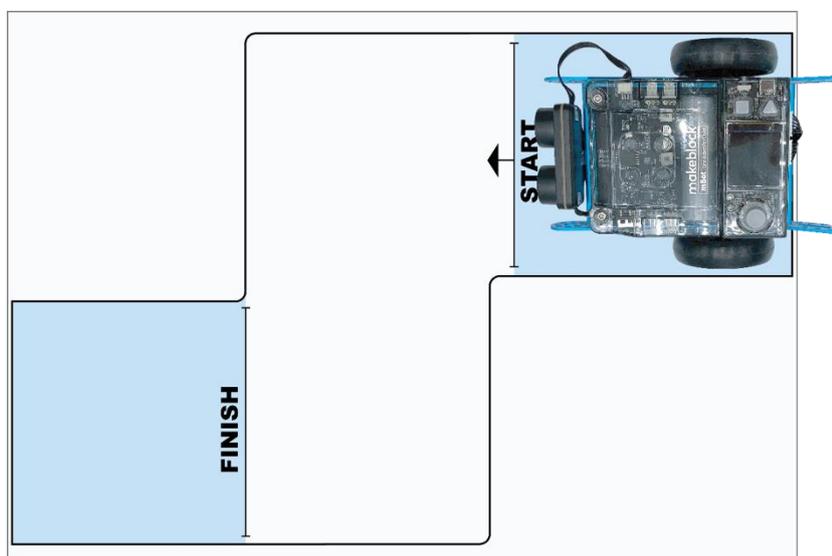
Vous pouvez voir ci-dessous quelques paires de blocs. Essayez chaque paire côte à côte. Que se passe-t-il ? Quelles sont les différences entre les blocs de code ?

| | Option 1 | Option 2 |
|----|----------|----------|
| 1. | | |
| 2. | | |
| 3. | | |

3. Test (25 min)

Étape 3 : Test

Il est maintenant temps de mettre votre mBot2 au travail. Prenez une feuille A3 et dessinez un labyrinthe dans lequel votre mBot2 doit se déplacer. Ne compliquez pas trop les choses et tenez compte de la largeur du robot. Le mBot2 n'a pas besoin de trouver le bon chemin tout seul. Le mBot2 a juste besoin de suivre la route que vous programmez. Besoin d'inspiration ? Vous pouvez voir un exemple ci-dessous.



Utilisez les connaissances que vous avez acquises à l'étape 2 de cette leçon. Bien sûr, vous pouvez faire de nombreuses expériences vous-même avec les différents exemples de programmation dans mBlock5.

Pour réaliser ce travail, il est utile de recourir au plan par étapes suivant.

| | Explication |
|---|--|
| Étape 1 : Que voulez-vous faire ? | <ul style="list-style-type: none"> • Quel trajet voulez-vous que le mBot2 suivre ? • Comment est composé le labyrinthe ? • Quelles sont les distances que le robot doit parcourir ? • Le robot doit-il également effectuer des virages ? Dans quel sens et de combien de degrés ? |
| Étape 2 : De quoi avez-vous besoin ? | <ul style="list-style-type: none"> • De quoi avez-vous besoin en plus du mBot2 ? |
| Étape 3 : De quels blocs de code avez-vous besoin pour faire fonctionner le mBot2 ? | <ul style="list-style-type: none"> • Comment allez-vous faire fonctionner le mBot2 ? • Quels blocs de code utiliserez-vous ? • Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou langage UML) • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock. |
| Étape 4 : Test et mise en œuvre | <ul style="list-style-type: none"> • La première version est-elle prête ? Testez-la ! Pendant le test, notez les points à améliorer. • Travaillez sur les points d'amélioration jusqu'à ce que votre mBot2 traverse votre labyrinthe sans erreur. |

Est-il possible de faire en sorte que le mBot2 traverse le labyrinthe sans problème ? Ensuite, relevez le défi de créer un labyrinthe plus difficile pour votre robot. Ou demandez à l'un de vos camarades de classe d'établir un itinéraire pour vous.

4. Récapitulatif (5 min)

Étape 4 : Récapitulatif

Avez-vous réussi à conduire le robot à travers le labyrinthe sans aucune erreur ?

Dans cette leçon, vous avez été initié aux robots dans la vie quotidienne et au mBot2. Vous savez comment contrôler les mouvements de mBot2 et quels blocs de programmation vous pouvez utiliser pour cela.

Il est maintenant temps de procéder à une courte réflexion. Réfléchissez par vous-même puis discutez-en avec le groupe :

- Qu'est-ce qui s'est bien passé, à votre avis ?
- Qu'est-ce qui pourrait être amélioré ?
- Quelles sont les parties de la leçon que vous avez trouvées faciles et celles que vous avez trouvées plus difficiles ?
- Sur quoi souhaitez-vous plus d'explications ?
- Qui pourrait vous aider ?

Leçon 2 : Détection et données

Sujet : STEAM

Classe(s) : 6ème et plus

Durée : 45 minutes

Difficulté : Débutant

★ Objectifs de la leçon

À la fin de cette leçon, les élèves seront capables de :

- Reconnaître et utiliser les blocs de code pour faire fonctionner les capteurs
- Afficher les données des capteurs sur l'écran du mBot2
- Créer leur propre programme informatique en mBlock pour le mBot2 afin de contrôler les capteurs et d'afficher les données sur l'écran



★ Vue d'ensemble

Un robot peut être un jouet amusant, mais saviez-vous que vous pouvez également l'utiliser pour la recherche ? Un robot possède plusieurs capteurs avec lesquels il peut percevoir son environnement et également collecter des données. Par exemple, vous pouvez demander au mBot2 de mesurer le son et la température. Vous pouvez lire les données sur l'écran du mBot2 et les utiliser pour des recherches sur la densité sonore ou la température de la salle de classe, par exemple.

🔗 Focus

À la fin de cette leçon, les élèves sauront :

- Quels sont les capteurs du mBot2
- Comment fonctionnent les capteurs du mBot2
- Quelle est la différence entre le mode Live et le mode Upload
- Afficher les données des capteurs sur l'écran du mBot2

Liste de contrôle avant la leçon

De quoi avez-vous besoin ?

- PC ou ordinateur portable (avec sortie USB) avec le logiciel mBlock installé, la version web (également pour ChromeBook), ou une tablette avec l'application mBlock installée
- mBot2 avec un CyberPi
- Câble USB-C ou un dongle Bluetooth Makeblock

Plan de cours

Cette leçon se compose de quatre étapes et dure au total 45 minutes.

| Durée | Contenu |
|------------|--|
| 5 minutes | 1. Introduction <ul style="list-style-type: none">• Capteurs et données dans la vie quotidienne.• Qu'est-ce que le CyberPi ? |
| 10 minutes | 2. Travaux pratiques <ul style="list-style-type: none">• Se familiariser avec les différents capteurs du mBot2.• Recréer et tester quelques exemples de programmation pour les capteurs.• Afficher les données des capteurs sur l'écran.• Différence entre le mode En direct et le mode Téléverser. |
| 25 minutes | 3. Test <ul style="list-style-type: none">• Écrire votre propre programme pour le robot. |
| 5 minutes | 4. Récapitulatif <ul style="list-style-type: none">• Showtime: montrez ce que vous avez fait avec votre robot dans un film court et amusant pour en discuter plus tard.• Si votre enseignant le permet, partagez le résultat final sur les réseaux sociaux avec le hashtag #mBot2inclass• Réflexion : De quoi êtes-vous le plus fier ? Qu'est-ce que vous aimeriez améliorer dans votre robot ? |

1. Introduction (5 min)

Étape 1 : Introduction

Cette étape se compose de deux parties :

1. Capteurs et données dans la vie quotidienne
2. Qu'est-ce que le CyberPi ?

1. Capteurs et données dans la vie quotidienne

Les capteurs qui collectent des données se trouvent dans de nombreux endroits différents de la vie quotidienne. Beaucoup plus que vous ne le pensez sans doute au premier abord. Par exemple, à l'aide de capteurs, l'éclairage extérieur de votre maison s'allume automatiquement lorsqu'il fait sombre dehors. Ou encore, le chauffage s'éteint automatiquement lorsqu'il fait trop chaud dans la salle de classe. La quantité de lumière extérieure et la chaleur dans la salle de classe sont toutes enregistrées à l'aide de capteurs. Tes camarades de classe et vous-même pouvez-vous penser à d'autres exemples ?

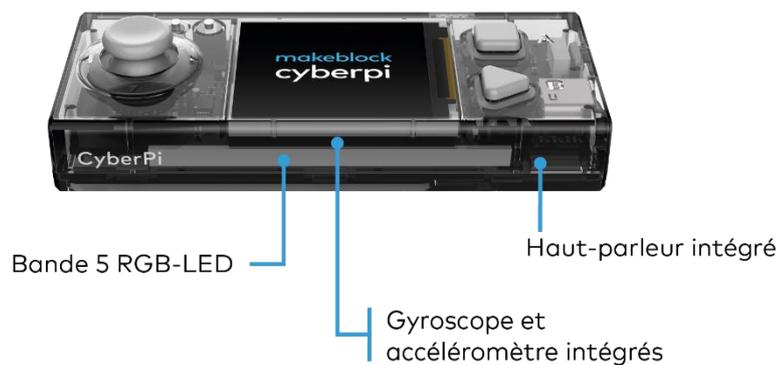
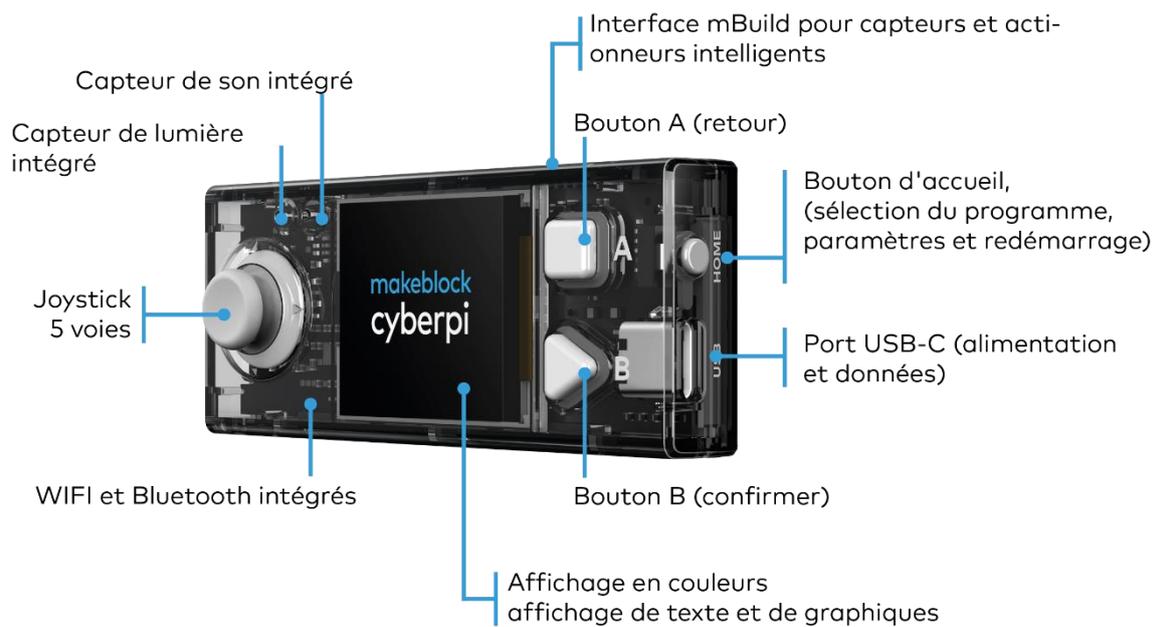
2. Qu'est-ce que le CyberPi ?

Pour faire fonctionner le mBot2, vous devez écrire un programme informatique. Cela se fait dans l'éditeur de code mBlock. Le programme informatique que vous écrivez consiste en une série de commandes que le mBot2 doit exécuter.



Un CyberPi est monté sur le mBot2. Un CyberPi est un petit micro-ordinateur programmable. Vous transmettez des commandes au CyberPi en utilisant les blocs de code dans mBlock. Le CyberPi transmet ces commandes au mBot2. Vous pouvez retirer le CyberPi du mBot2.

Le CyberPi possède de nombreuses fonctions différentes, comme un microphone, un haut-parleur et un joystick. Il y a également de nombreux capteurs sur le CyberPi. Regardez l'image ci-dessous.



2. Travaux pratiques (10 min)

Étape 2 : Travaux pratiques

Cette étape se compose de quatre parties :

1. Se familiariser avec les différents capteurs du mBot2.
2. Recréer et tester quelques exemples de programmation pour les capteurs.
3. Afficher les données des capteurs.
4. Différence entre le mode En direct et le mode Téléverser.

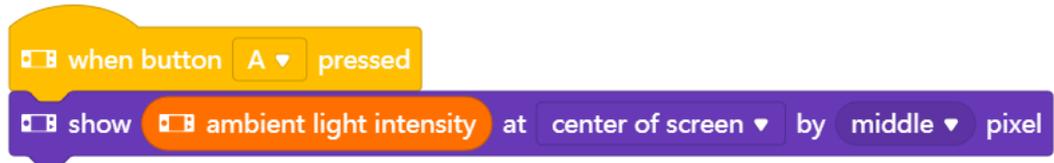
1. Apprendre à connaître les capteurs mBot2

Chaque robot fonctionne avec des capteurs. Les capteurs peuvent être comparés à vos sens (=goût, toucher, odorat, ouïe, vue). Grâce à ces capteurs, le mBot2 « voit » son environnement.

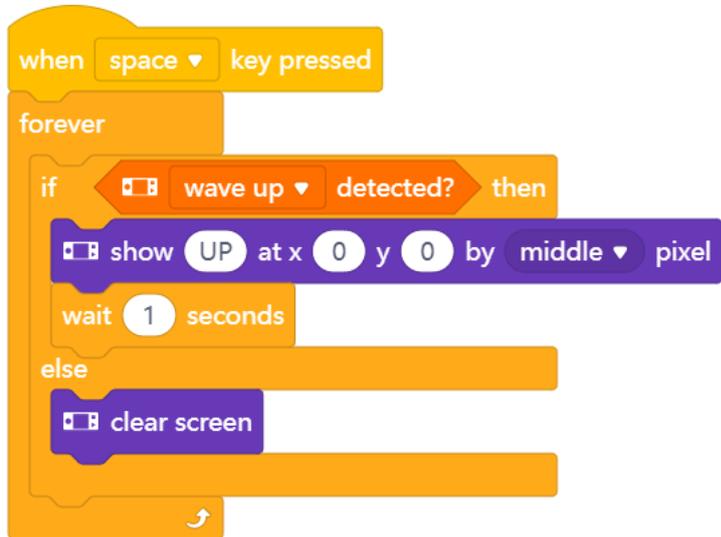
Il existe différents types de capteurs qui peuvent permettre au mBot2 de « voir », tels que :

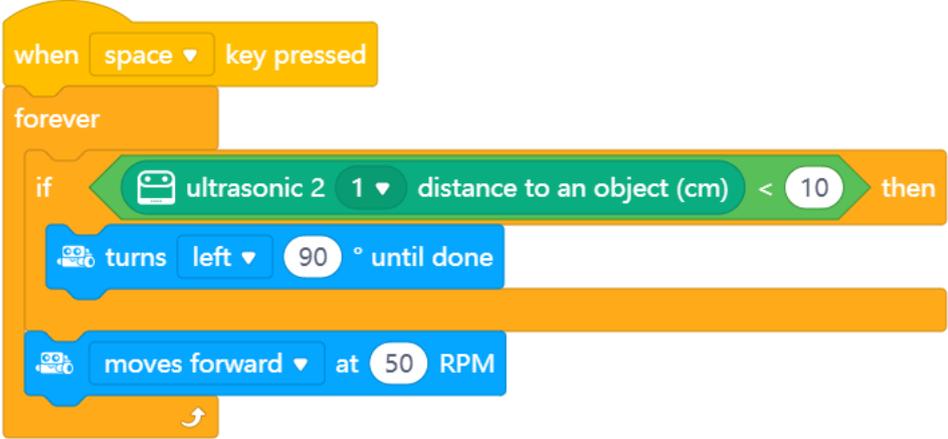
- Capteur de lumière
- Capteur de son
- Gyroscope et accéléromètre
- Capteur quadruple RVB
- Minuterie

Le tableau ci-dessous liste les capteurs du mBot2. Chaque capteur est accompagné d'une brève explication et d'un exemple de programmation.

| Type de capteur | Que fait ce capteur ? |
|--------------------|---|
| Capteur de lumière | <p>La lumière est générée par de l'énergie lumineuse. Le capteur de lumière est un dispositif qui convertit l'énergie lumineuse en énergie électrique. Un capteur de lumière est souvent utilisé, par exemple, pour :</p> <ul style="list-style-type: none"> • Adapter automatiquement la luminosité de l'écran des smartphones à l'environnement ; • Contrôler l'éclairage dans les maisons ou pour allumer automatiquement les phares des véhicules <p>Dans l'exemple de programmation ci-dessous, l'énergie lumineuse de l'environnement est affichée sur l'écran du CyberPi. Vous pouvez étendre l'exemple de programmation pour que le mBot2 s'arrête de rouler, par exemple, lorsque l'énergie lumineuse de l'environnement est inférieure à une certaine valeur, comme lorsqu'il fait nuit.</p> |
| | <p style="text-align: center;">Exemple de programmation</p>  |
| Type de capteur | Que fait ce capteur ? |
| Capteur de son | <p>Le son est une vibration mécanique qui se propage en ondes. Dans l'air, il s'agit de fluctuations de pression et de densité. Si la vibration est dans la gamme audible (entre 16 et 20 000 vibrations par seconde) et suffisamment intense, nous pouvons l'entendre sous forme de tons ou de sons.</p> <p>La puissance du son est appelée l'intensité sonore.</p> <p>Le capteur du microphone convertit le son en un signal électrique qui peut être évalué en termes de hauteur et de puissance.</p> <p>Un capteur de son est souvent utilisé pour les choses suivantes :</p> <ul style="list-style-type: none"> • Appels téléphoniques, enregistrement sonore • Systèmes d'assistance à commande vocale, par exemple sur un téléphone portable ou dans une maison intelligente <p>Dans l'exemple de programmation ci-dessous, l'intensité sonore de l'environnement est affichée sur l'écran du CyberPi. Vous pouvez étendre</p> |



| | |
|----------------------------|---|
| | <p>l'exemple de programmation de manière à ce que le mBot2 mesure en permanence l'intensité sonore pendant la visite de la salle de classe et l'affiche sur l'écran.</p> |
| | Exemple de programmation |
| |  |
| Type de capteur | Que fait ce capteur ? |
| Gyroscope et accéléromètre | <p>Un gyroscope mesure les mouvements de basculement, plus précisément la vitesse des mouvements de rotation/basculement. Un accéléromètre mesure la variation de la vitesse. Ces capteurs peuvent être mis en œuvre sous forme de composants micromécaniques sur un composant électronique. Les deux capteurs fournissent des informations différentes sur la position dans l'espace.</p> <p>Un gyroscope est souvent utilisé pour, par exemple :</p> <ul style="list-style-type: none">• Assurer la stabilité des navires en haute mer ou• Équilibrer un Segway ou un « hoverboard » afin de ne vous éviter une chute rapide <p>Les capteurs d'accélération, quant à eux, indiquent si le smartphone est orienté vers le haut ou vers le bas, ou si le véhicule est en train de subir un accident (changement violent et très rapide de la vitesse), afin de déclencher les airbags.</p> <p>Dans l'exemple de programmation ci-dessous, le mouvement d'inclinaison du mBot2 est affiché à l'écran. Vous pouvez étendre l'exemple de programmation pour que le mBot2 mesure et enregistre en permanence le mouvement de basculement sur l'écran pendant un tour de la salle de classe.</p> <p>Vous en apprendrez davantage sur ce capteur et ses capacités dans la leçon 6.</p> |
| | Exemple de programmation |
| |  |

| Type de capteur | Que fait ce capteur ? |
|---------------------------------|---|
| Capteur à ultrasons | <p>Le son, qui est une vibration sous forme de variations de densité et de pression, se propage dans l'air à une vitesse constante sous forme d'ondes (la vitesse du son dans l'air est d'environ 334 m/s). Plus la vibration est élevée, plus le son apparaît supprimer le tiret jusqu'à une limite au-delà de laquelle les gens ne peuvent plus percevoir ce son. Ces vibrations très élevées sont appelées ultrasons. Puisque la vitesse du son dans l'air est connue, le son peut être utilisé pour déterminer les distances aux objets ; pour ce faire, on émet un son et on mesure le temps jusqu'à ce que ce son soit réfléchi par l'objet. Cette onde sonore réfléchie est également appelée écho. Les ultrasons sont généralement utilisés à cette fin.</p> <p>Par exemple, un capteur à ultrasons est souvent utilisé :</p> <ul style="list-style-type: none"> • pour une procédure d'imagerie, par exemple lors d'une grossesse (chaque pixel est une mesure de distance) ou • dans le contrôle des robots pour éviter les collisions <p>Dans l'exemple de programmation ci-dessous, il faut indiquer la procédure pour installer l'extension comme à la page 49, sinon l'exemple n'est pas faisable, le capteur à ultrasons est utilisé pour empêcher le mBot2 de foncer sur un obstacle. Lorsque le mBot2 se trouve à moins de 10 cm d'un obstacle, le robot effectue un virage de 90° vers la gauche, puis continue simplement à rouler. Vous pouvez étendre cet exemple de programmation pour que le mBot2 se déplace au hasard dans une salle de classe sans se heurter aux tables et aux chaises.</p> <p>Vous en apprendrez davantage sur ce capteur et ses capacités dans la leçon 4.</p> |
| Exemple de programmation | |
| |  <pre> when space key pressed forever if ultrasonic 2 distance to an object (cm) < 10 then turns left 90° until done moves forward at 50 RPM </pre> |
| Type de capteur | Que fait ce capteur ? |
| Capteur quadruple RVB | <p>Le capteur quadruple RVB se compose de quatre capteurs individuels de lumière et de couleur. Ils mesurent l'intensité de la lumière qui pénètre dans le capteur à partir des zones rouge, verte et bleue du spectre lumineux. Cela permet au capteur de détecter les couleurs des objets situés directement devant lui, comme les marquages au sol, et permet également au robot de suivre une ligne noire pour s'orienter.</p> <p>Un capteur quadruple RVB est largement utilisé dans des applications comme :</p> |



- montrer à un robot d'entrepôt un chemin à travers l'entrepôt (également différentes couleurs pour les différents chemins), ou
 - s'assurer que la bonne couleur est utilisée lors de la peinture
- Dans l'exemple de programmation ci-dessous, le mBot2 avance lorsque le robot voit la couleur blanche. Vous pouvez étendre l'exemple de programmation pour que le mBot2 suive un itinéraire à travers la classe, s'arrête ou tourne en fonction de différentes couleurs. Vous en apprendrez davantage sur ce capteur et ses capacités dans la leçon 5.

Exemple de programmation

```
when space key pressed
forever
  if quad rgb sensor 1 probe (2) R1 detected white ? then
    moves forward at 50 RPM
  else
    stop encoder motor all
```

Type de capteur

Que fait ce capteur ?

Minuterie

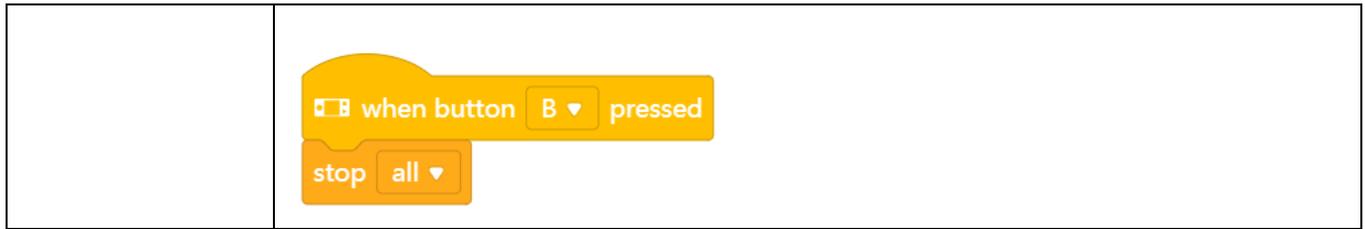
La minuterie est une sorte de chronomètre qui indique le temps en secondes depuis que le CyberPi a été allumé ou réinitialisé. Ce compteur peut également être mis à zéro par une commande pour faciliter les mesures de temps.

Par exemple, vous pouvez utiliser le chronomètre pour organiser une course de plusieurs mBots les uns contre les autres et mesurer exactement leur vitesse à la ligne d'arrivée.

L'exemple de programmation suivant vous montre comment régler et utiliser la minuterie. En appuyant sur la touche 'A', vous remettez la minuterie à zéro et l'affichez en permanence à l'écran. Vous pouvez terminer l'affichage et le programme dans son ensemble en appuyant sur 'B'. Essayez d'étendre l'exemple de programmation afin que le minuteur commence à compter lorsque le mBot2 commence à se déplacer.

Exemple de programmation

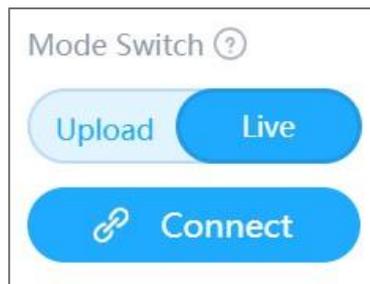
```
when button A pressed
reset timer
forever
  show timer(s) at center of screen by middle pixel
```



2. Test et extension des exemples de programmation pour les capteurs

Dans le tableau ci-dessus, il y a un exemple de programmation pour chaque capteur. Vous allez recréer ces exemples de programmation dans mBlock et les tester. Pensez à une extension pour deux des exemples de programmation. Le tableau ci-dessus mentionne déjà une suggestion d'extension pour chaque exemple de programmation. Peut-être connaissez-vous une extension beaucoup plus agréable !

En écrivant un programme informatique, vous pouvez immédiatement tester ce que vous créez. Pour ce faire, vous utilisez le mode En direct. Vous sélectionnez le mode En direct en déplaçant le commutateur de mode vers la droite. Regardez l'image ci-dessous.



3. Afficher les données des capteurs

En testant les exemples de programmation, vous avez vu que chaque capteur affiche quelque chose sur l'écran du CyberPi. Par exemple, le capteur de lumière enregistre l'intensité lumineuse de l'environnement. Le minuteur enregistre l'heure, et le capteur gyroscopique garde la trace de la façon dont le mBot2 s'incline. Tout ce qu'un capteur enregistre s'appelle des données.

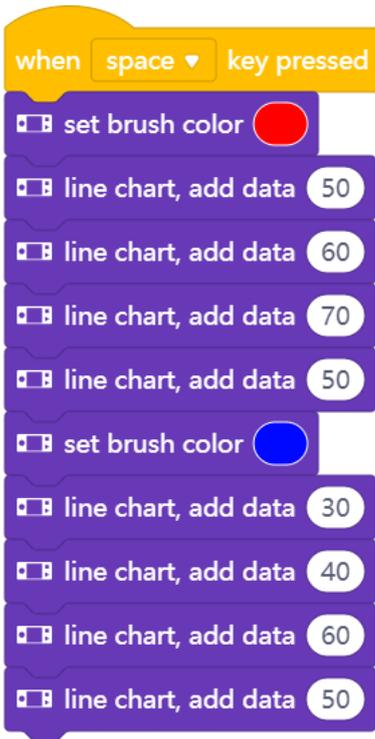
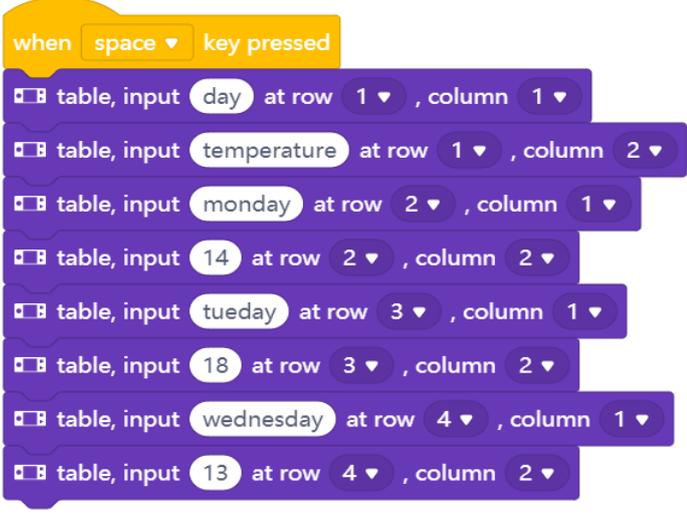
Vous pouvez afficher les données des capteurs sur l'écran du mBot2. Dans les exemples de programmation, vous avez vu que vous pouvez afficher les données avec, par exemple, un nombre ou un texte.



Vous pouvez afficher les données des capteurs de différentes manières sur l'écran du mBot2. Vous pouvez utiliser différents blocs de code pour cela. Ces blocs de code peuvent être trouvés dans mBlock sous la catégorie 'Affichage'.

Dans le diagramme ci-dessous, vous voyez trois exemples de la façon d'afficher des données sur l'écran du CyberPi. Reproduisez les exemples de programmation et testez-les en mode réel. Regardez ce qui se passe sur l'écran du la mBot2.

| Exemple de programmation | |
|--------------------------|--|
| Diagramme linéaire | |

| | |
|---|---|
| <p>Diagramme linéaire (couleur)</p> |  <p>when space key pressed</p> <ul style="list-style-type: none"> set brush color (red) line chart, add data (50) line chart, add data (60) line chart, add data (70) line chart, add data (50) set brush color (blue) line chart, add data (30) line chart, add data (40) line chart, add data (60) line chart, add data (50) |
| <p>Tableau</p> |  <p>when space key pressed</p> <ul style="list-style-type: none"> table, input day at row 1, column 1 table, input temperature at row 1, column 2 table, input monday at row 2, column 1 table, input 14 at row 2, column 2 table, input tuesday at row 3, column 1 table, input 18 at row 3, column 2 table, input wednesday at row 4, column 1 table, input 13 at row 4, column 2 |

4. Mode En direct et mode Téléverser

Vous devez transférer le programme informatique que vous avez construit dans mBlock vers le CyberPi.

Le CyberPi fait fonctionner le mBot2 en fonction des commandes que vous avez écrites dans mBlock. En écrivant un programme informatique, vous pouvez immédiatement tester ce que vous créez. Pour ce faire, vous utilisez le mode En direct. Vous sélectionnez le mode En direct en déplaçant le commutateur de mode vers la droite. Vous ne devez pas déconnecter le CyberPi de l'ordinateur pendant les tests en mode En direct !

Le CyberPi dispose également d'un mode Téléverser. En mode téléchargement, le programme informatique est transféré sur le CyberPi. Le programme est stocké sur le CyberPi jusqu'à ce

que vous le remplaciez par un autre programme. Vous pouvez simplement débrancher le CyberPi.

Dans le diagramme ci-dessous, vous pouvez voir les différences entre le mode En direct et le mode Téléverser.

| Mode | Différences expliquées |
|-----------------|--|
| Mode En direct | <ul style="list-style-type: none">• Programmes exécutés sur votre ordinateur (votre PC ou votre portable). Ils ne seront pas stockés sur le CyberPi.• Le CyberPi doit rester connecté à votre ordinateur.• L'éditeur de code mBlock doit toujours être ouvert sur votre ordinateur.• Ce mode est utilisé pour la programmation en scène. |
| Mode Téléverser | <ul style="list-style-type: none">• Les programmes s'exécutent sur le CyberPi et non sur votre ordinateur et sont également stockés sur le CyberPi.• Le CyberPi n'a pas besoin d'être connecté à votre ordinateur.• L'éditeur de code mBlock n'a pas besoin d'être ouvert sur votre ordinateur.• Un programme que vous téléchargez sur le CyberPi est enregistré sur le CyberPi jusqu'à ce que vous le remplaciez par un autre programme. |

3. Test (25 min)

Étape 3 : Test

Vous avez déjà appris beaucoup de choses sur les capteurs du mBot2 et sur la manière d'afficher les données des capteurs sur l'écran du CyberPi. Vous allez maintenant collecter vous-même des données en utilisant les capteurs du mBot2 et les afficher sur l'écran du CyberPi. Vous le ferez en conduisant le mBot2 dans la salle de classe ou l'école. C'est à vous de décider quelles données vous collectez et comment vous les affichez sur l'écran du CyberPi. Utilisez les connaissances que vous avez acquises à l'étape 2 de cette leçon. Bien sûr, vous pouvez faire de nombreuses expériences vous-même avec les différentes possibilités de programmation dans mBlock5.

Lorsque vous réfléchissez à cette mission, il est utile de recourir au plan par étapes suivant. Avez-vous une idée de ce que vous voulez faire ? Si c'est le cas, discutez-en d'abord avec votre professeur pour savoir si cela est possible.

| | Explication |
|---|---|
| Étape 1 : Que voulez-vous faire ? | <ul style="list-style-type: none"> • Quel trajet voulez-vous que le mBot2 suivre ? • Quelles données voulez-vous que le mBot2 collecte ? |
| Étape 2 : De quoi avez-vous besoin ? | <ul style="list-style-type: none"> • De quoi avez-vous besoin en plus du mBot2 ? |
| Étape 3 : De quels blocs de code avez-vous besoin pour faire fonctionner le mBot2 ? | <ul style="list-style-type: none"> • Comment allez-vous faire fonctionner le mBot2 ? • Quelles données voulez-vous que le mBot2 collecte ? • Quels blocs de code utiliserez-vous ? • Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML) • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock. |
| Étape 4 : De quelle manière voulez-vous afficher les données des capteurs sur l'écran ? | <ul style="list-style-type: none"> • Comment voulez-vous afficher les données sur l'écran ? • Quels blocs de code utiliserez-vous ? • Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML) • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock. |
| Étape 5 : Test et mise en œuvre | <ul style="list-style-type: none"> • La première version est-elle prête ? Testez-la ! Pendant le test, notez les points à améliorer. • Travaillez sur les points d'amélioration jusqu'à ce que le mBot2 fasse exactement ce que vous aviez en tête. • Vous avez réussi ? Filmez le résultat final et demandez à votre professeur si vous pouvez le publier sur les réseaux sociaux avec le hashtag #mBot2inclass |

4. Récapitulatif (5 min)

Étape 4 : Récapitulatif

Quelles données votre mBot2 a-t-il recueillies ? Ça s'est bien passé ? Avez-vous pu voir toutes les données sur l'écran du CyberPi ?

Dans cette leçon, vous avez appris à connaître les différents capteurs du mBot2 et les situations où vous pouvez les rencontrer dans la vie quotidienne. Vous savez comment programmer ces capteurs et comment afficher les données sur l'écran. Vous avez également appris la différence entre les modes « Téléverser » et « En direct » de mBlock.

Il est maintenant temps de procéder à une courte réflexion. Réfléchissez par vous-même puis discutez-en avec le groupe :

- Qu'est-ce qui s'est bien passé, à votre avis ?
- Qu'est-ce qui pourrait être amélioré ?
- Quelles sont les parties de la leçon que vous avez trouvées faciles et celles que vous avez trouvées plus difficiles ?
- Sur quoi souhaitez-vous plus d'explications ?
- Qui pourrait vous aider ?

Leçon 3 : Écoutez mBot2

Sujet : STEAM
Durée : 45 minutes

Classe(s) : CM2 et plus
Difficulté : Débutant

★ Objectifs de la leçon

À la fin de cette leçon, les élèves seront capables de :

- Reconnaître et utiliser les blocs de code pour contrôler le haut-parleur et le microphone
- Construire leur propre programme informatique dans mBlock pour que le mBot2 enregistre des sons et les restitue.
- Exécuter plusieurs tâches de programmation côte à côte dans un programme informatique

★ Aperçu

Souvent, les robots peuvent parler. Prenez, par exemple, le robot aspirateur qui utilise un message vocal pour vous informer que sa batterie est faible. Ou un jouet robot qui chante une chanson. Le mBot2 peut également produire et enregistrer du son. Il dispose d'un microphone et d'un haut-parleur pour cela. Avec le microphone, le mBot2 enregistre le son et le haut-parleur restitue le son enregistré.

🔗 Focus

À la fin de cette leçon, les élèves sauront :

- Comment utiliser le haut-parleur pour le codage et la robotique
- Comment enregistrer et lire le son
- Exécuter plusieurs tâches simultanément

📄 Liste de contrôle avant la leçon

De quoi avez-vous besoin ?

- PC ou ordinateur portable (avec sortie USB) avec le logiciel mBlock installé, la version web (également pour Chromebook), ou une tablette avec l'application mBlock installée
- mBot2 avec un CyberPi
- Câble USB-C ou un dongle Bluetooth Makeblock



Plan de cours

Cette leçon se compose de quatre étapes et dure au total 45 minutes.

| Durée | Contenu |
|------------|---|
| 5 minutes | 1. Introduction <ul style="list-style-type: none">• Les haut-parleurs dans la vie quotidienne• Comment fonctionne le haut-parleur ? |
| 10 minutes | 2. Travaux pratiques <ul style="list-style-type: none">• Se familiariser avec les différents blocs de code du haut-parleur et du microphone.• Reproduire, tester et étendre certains exemples de programmation du haut-parleur et du microphone. |
| 25 minutes | 3. Test <ul style="list-style-type: none">• Écrire votre propre programme pour le robot. |
| 5 minutes | 4. Récapitulatif <ul style="list-style-type: none">• Showtime : montrez ce que vous avez fait avec votre robot dans un film court et amusant pour en discuter plus tard.• Si votre enseignant le permet, partagez le résultat final sur les médias sociaux avec le hashtag #mBot2inclass• Réflexion: De quoi êtes-vous le plus fier ? Qu'est-ce que vous aimeriez améliorer dans votre robot ? |

1. Introduction (5 min)

Étape 1 : Introduction

Cette étape se compose de deux parties :

1. Les haut-parleurs dans la vie quotidienne
2. Comment fonctionne le haut-parleur ?

1. Les haut-parleurs dans la vie quotidienne

Les haut-parleurs sont utilisés tout autour de vous dans la vie quotidienne. Vous trouverez des systèmes de haut-parleurs de grande taille dans les théâtres, les salles de concert et les salles d'événements. Ces enceintes sont utilisées pour amplifier le son pour un grand groupe de personnes. On trouve des systèmes de haut-parleurs plus petits dans des appareils tels que les téléviseurs, les instruments de musique et les smartphones. Pouvez-vous, vous et les camarades de votre groupe, imaginer d'autres applications ?

2. Comment fonctionne le haut-parleur ?

Le but d'un haut-parleur est de reproduire un son. Pour ce faire, les haut-parleurs convertissent les ondes électromagnétiques en ondes sonores. Outre sa fonction principale, un haut-parleur peut également reconverter les ondes sonores en ondes électromagnétiques. Les microphones sont des appareils qui utilisent le même principe, mais qui sont optimisés dans le but de convertir les ondes sonores en signaux électriques.

Sur le mBot2, il y a un haut-parleur et un microphone. Le microphone se trouve en haut à gauche du CyberPi. Le haut-parleur se trouve à l'avant du CyberPi. Le microphone permet d'enregistrer des sons. Ces sons peuvent être restitués ultérieurement avec le haut-parleur.



En utilisant le microphone et le haut-parleur en combinaison avec les autres capteurs du mBot2, vous pouvez, par exemple, faire lire votre propre texte enregistré à un moment précis. Par exemple, pensez à un bus qui vous indique l'arrêt auquel vous êtes presque arrivé. Il existe d'innombrables autres possibilités d'utilisation du microphone et du haut-parleur. Avec les camarades de votre groupe, pensez à d'autres exemples où le mBot2 peut faire bon usage du haut-parleur et du microphone.

2. Travaux pratiques (10 min)

Étape 2 : Travaux pratiques

Cette étape se compose de deux parties :

1. Se familiariser avec les différents blocs de code du haut-parleur et du microphone.
2. Reproduire, tester et étendre certains exemples de programmation du haut-parleur.

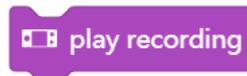
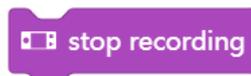
1. Se familiariser avec les différents blocs de code du haut-parleur et du microphone

Dans mBlock 5, il y a plusieurs blocs de code que vous pouvez utiliser pour programmer le haut-parleur et le microphone. Vous trouverez ces blocs de code dans la catégorie « Audio » de la zone des blocs dans mBlock. Ces blocs de codage sont violets.

Dans le tableau ci-dessous, vous verrez certains de ces blocs de code pour programmer le haut-parleur et le microphone.

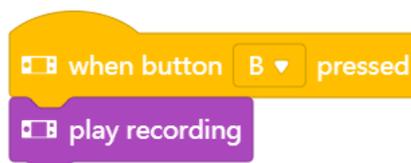
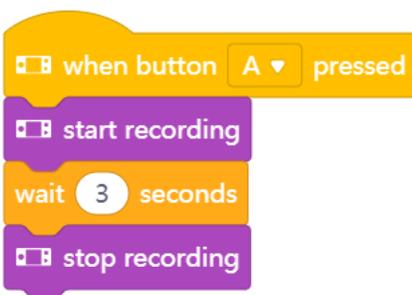


Bloc de code :



Lorsque vous enregistrez et lisez un son, vous utilisez en fait toujours trois blocs de code. Avec le bloc de code « start recording », vous laissez CyberPi commencer à enregistrer un son. Le signal électrique provenant du microphone est stocké dans la mémoire du CyberPi, et l'enregistrement s'arrête automatiquement après 10 secondes. Vous pouvez également indiquer quand l'enregistrement doit s'arrêter. Cette opération s'effectue à l'aide du bloc de code « Arrêter l'enregistrement ». Le son enregistré est temporairement stocké dans la mémoire du CyberPi. Vous éteignez le mBot2 ? Le son enregistré est alors perdu. La lecture du son enregistré est contrôlée par le bloc de code « play recording ».

Dans l'exemple de programmation ci-dessous, lorsque vous appuyez sur le bouton A, le CyberPi commence à enregistrer. Après 3 secondes, l'enregistrement s'arrête. Lorsque vous appuyez sur le bouton B, vous entendez ce qui a été enregistré. En utilisant deux blocs d'événements, deux tâches de programmation peuvent s'exécuter indépendamment l'une de l'autre sans qu'une boucle principale soit nécessaire.



Bloc de code :

play recording until done

Vous venez d'apprendre à lire un son enregistré. Avec ce bloc de code, vous vous assurez que le son enregistré par le microphone est lu jusqu'au bout. Cela signifie que l'exécution d'autres blocs est impossible jusqu'à ce que celui-ci soit terminé. Une fois le son terminé, vous pouvez faire faire autre chose au mBot2. Par exemple, faites un certain bruit pour savoir que l'enregistrement est terminé.

Dans l'exemple de programmation à droite, le CyberPi commence à enregistrer lorsque vous appuyez sur le bouton A. Si vous appuyez sur le bouton B, l'enregistrement s'arrête. Si vous déplacez le joystick vers le bas, vous entendrez ce que le mBot2 a enregistré. Après la fin de l'enregistrement, vous entendrez un son « prompt ». Les blocs de code de cet exemple sont placés dans une structure dite en boucle. Cela garantit que les entrées des boutons et du joystick sont contrôlées en permanence.

```

when space key pressed
  forever
    if button A pressed? then
      start recording
    if button A pressed? then
      stop recording
    if joystick pulled down? then
      play recording until done
      play prompt
  
```

2. Reproduire, tester et étendre certains exemples de programmation du haut-parleur

Dans le tableau ci-dessus, chaque bloc de code du haut-parleur est accompagné d'un exemple de programmation. Vous allez recréer ces exemples de programmation dans mBlock et les tester. Pensez à une extension à un exemple de programmation. Testez les exemples de programmation et votre propre extension en mode En direct.

3. Test (25 min)

Étape 3 : Test

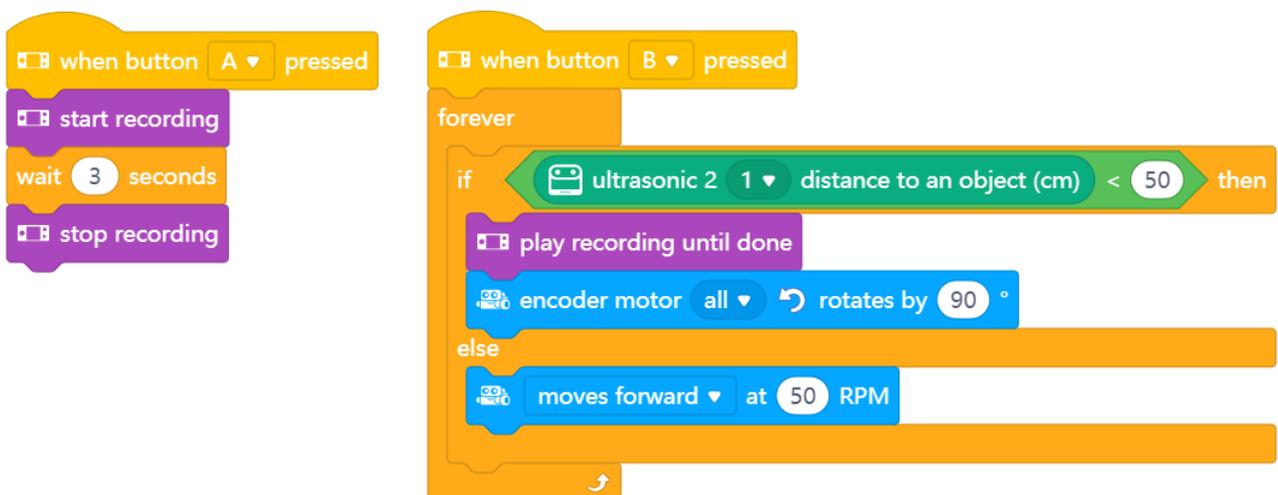
Vous avez déjà appris beaucoup de choses sur le haut-parleur et le microphone du mBot2. Vous allez maintenant travailler seul avec eux. Vous allez également utiliser le capteur à ultrasons. Ce capteur peut détecter un objet à distance. Vous en apprendrez plus sur le capteur à ultrasons dans la leçon 4.

Dans ce travail, vous allez faire en sorte que le mBot2 fasse un tour aléatoire de la salle de classe. Aléatoire signifie que vous ne déterminez pas un itinéraire à l'avance. Si le mBot2 s'approche d'un obstacle, comme une chaise, une table ou un sac à dos, vous entendrez un son et le mBot2 fera demi-tour. Vous enregistrerez vous-même le son avec le microphone. Ce son peut être, par exemple, un signal sonore fort ou le bruit d'une voiture qui ralentit. Vous pouvez également enregistrer un texte ou demander au mBot2 de jouer une chanson.

Il est assez difficile de trouver tout seul un programme informatique pour effectuer tout cela ! Heureusement, vous pouvez obtenir de l'aide. Sur la page suivante, vous voyez deux exemples de programme informatique. Vous pouvez développer et adapter ces programmes informatiques par vous-même. Soyez créatifs !

Exemple 1

Le mBot2 émet un son lorsqu'il s'approche d'un obstacle et fait demi-tour. Vous avez enregistré le son vous-même tout à l'heure.

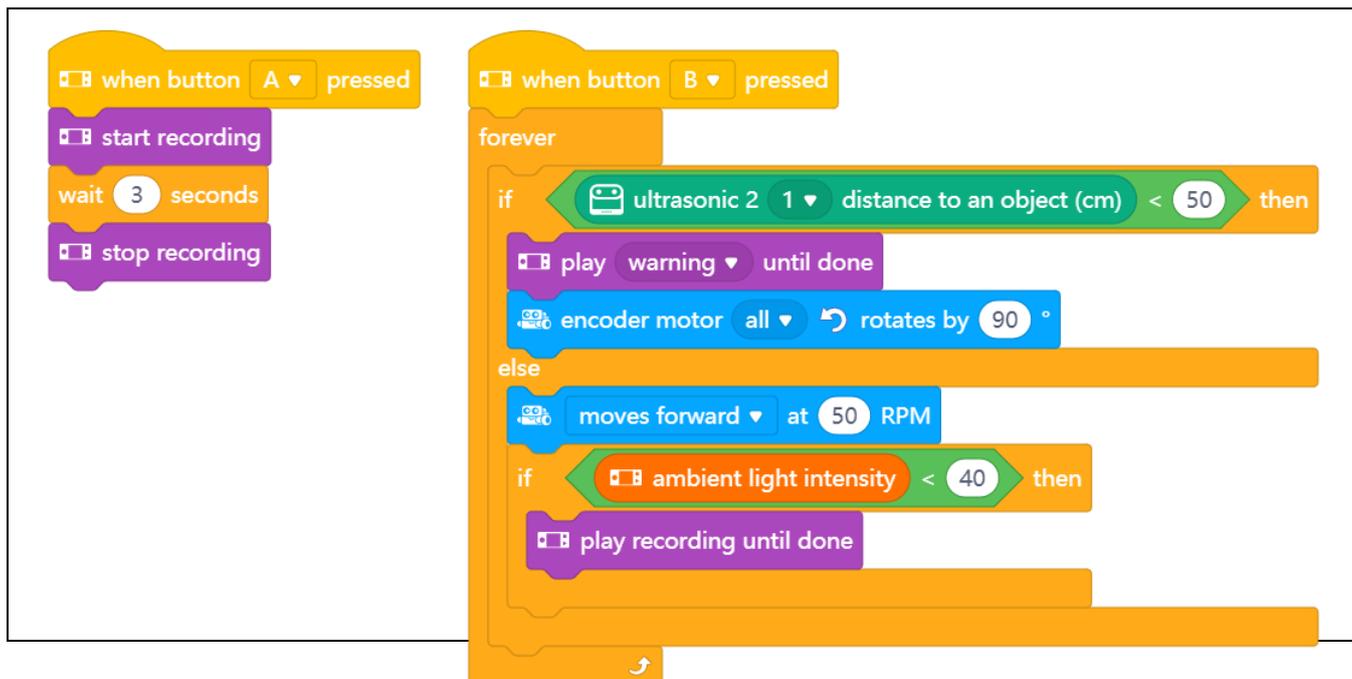


```
when button A pressed
  start recording
  wait 3 seconds
  stop recording

when button B pressed
  forever loop
    if ultrasonic 2 1 distance to an object (cm) < 50 then
      play recording until done
      encoder motor all rotates by 90 degrees
    else
      moves forward at 50 RPM
```

Exemple 2

Le mBot2 se déplace et joue un son que vous avez enregistré lorsque la lumière ambiante est faible. Lorsque le mBot2 s'approche d'un obstacle, il émet un son d'avertissement et fait demi-tour.



Utilisez les connaissances que vous avez acquises à l'étape 2 de cette leçon. Bien sûr, vous pouvez faire de nombreuses expériences vous-même avec les différentes possibilités de programmation dans mBlock5.

Lorsque vous réfléchissez à cette mission, il est utile de recourir au plan par étapes suivant. Avez-vous une idée de ce que vous voulez faire ? Si c'est le cas, discutez-en d'abord avec votre professeur pour savoir si cela est possible.

| | Explication |
|---|--|
| Étape 1 : Que voulez-vous faire ? | <ul style="list-style-type: none"> • Quand le mBot2 doit-il faire demi-tour ? • Quel son doit émettre le mBot2 lorsqu'il s'approche d'un obstacle ? |
| Étape 2 : De quoi avez-vous besoin ? | <ul style="list-style-type: none"> • De quoi avez-vous besoin en plus du mBot2 ? |
| Étape 3 : De quels blocs de code avez-vous besoin pour faire fonctionner le mBot2 ? | <ul style="list-style-type: none"> • Comment allez-vous faire fonctionner le mBot2 ? • Quels blocs de code utiliserez-vous ? • Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML) • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock. |
| Étape 4 : De quelle manière voulez-vous afficher les données des capteurs sur l'écran ? | <ul style="list-style-type: none"> • Quel son doit émettre le mBot2 ? • Quels blocs de code utiliserez-vous ? • Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML) • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock. |



| | |
|---------------------------------|--|
| Étape 5 : Test et mise en œuvre | <ul style="list-style-type: none">• La première version est-elle prête ? Testez-la ! Pendant le test, notez les points à améliorer.• Travaillez sur les points d'amélioration jusqu'à ce que votre mBot2 fasse exactement ce que vous aviez en tête.• Vous avez réussi ? Filmez le résultat final et demandez à votre professeur si vous pouvez le publier sur les réseaux sociaux avec le hashtag #mBot2inclass |
|---------------------------------|--|

4. Récapitulatif (5 min)

Étape 4 : Récapitulatif

Comment s'est passée cette mission ? Le mBot2 a-t-il fait immédiatement ce que vous aviez en tête ?

Dans cette leçon, vous avez appris comment fonctionnent les haut-parleurs et où vous les rencontrez dans la vie quotidienne. Vous savez comment programmer le CyberPi et le mBot pour qu'ils puissent enregistrer via le microphone intégré et être lus par les haut-parleurs. Vous savez également comment faire fonctionner plusieurs programmes indépendamment les uns des autres.

Il est maintenant temps de procéder à une courte réflexion. Réfléchissez par vous-même puis discutez-en avec le groupe :

- Qu'est-ce qui s'est bien passé, à votre avis ?
- Qu'est-ce qui pourrait être amélioré ?
- Quelles sont les parties de la leçon que vous avez trouvées faciles et celles que vous avez trouvées plus difficiles ?
- Sur quoi souhaitez-vous plus d'explications ?
- Qui pourrait vous aider ?

Leçon 4 : Voir grâce au son

Sujet : STEAM

Durée : 45 minutes

Classe(s) : CM2 et plus

Difficulté : Débutant

★ Objectifs de la leçon

À la fin de cette leçon, les élèves sauront :

- Ce que sont les ultrasons et comment fonctionne un capteur basé sur les ultrasons
- Les applications de ces capteurs dans la vie quotidienne et en robotique

★ Aperçu

Les robots autonomes doivent être conscients de leur environnement lorsqu'ils conduisent sans surveillance afin d'éviter toute collision avec des obstacles ou des personnes. Cela vaut également pour le mBot2. Le mBot2 a un capteur spécifique pour cela à l'avant. Il s'agit d'un capteur à ultrasons qui permet au robot de détecter les objets situés sur son chemin à l'aide de sons que les humains ne peuvent pas entendre, d'où leur nom (ultrasons).

Focus

À la fin de cette leçon, les élèves seront capables de :

- Utiliser un capteur à ultrasons pour la détection de la portée et des obstacles
- Faire réagir le mBot2 aux obstacles et les éviter en conduisant

Faire passer ces deux lignes page suivante

De quoi avez-vous besoin ?

- PC ou ordinateur portable (avec sortie USB) avec le logiciel mBlock installé, la version web (également pour ChromeBook), ou une tablette avec l'application mBlock installée
- Le mBot2 avec un CyberPi
- Un câble USB-C ou un dongle Bluetooth Makeblock
- Petits obstacles (la taille minimale est de 10x10x10 cm)
- Un espace de travail d'au moins 1m² où le mBot2 peut se déplacer

Plan de cours

Cette leçon se compose de quatre étapes et dure au total 45 minutes.

| Durée | Contenu |
|------------|--|
| 5 minutes | 1. Introduction <ul style="list-style-type: none">• Les capteurs à ultrasons dans la vie quotidienne• Comment fonctionne un capteur à ultrasons ? |
| 10 minutes | 2. Travaux pratiques <ul style="list-style-type: none">• Apprendre à connaître les différents blocs de code du capteur à ultrasons.• Recréer et tester quelques exemples de programmation du capteur à ultrasons. |
| 25 minutes | 3. Test <ul style="list-style-type: none">• Écrire votre propre programme pour le robot. |
| 5 minutes | 4. Récapitulatif <ul style="list-style-type: none">• Showtime : montrez ce que vous avez fait avec votre robot dans un film court et amusant pour en discuter plus tard.• Si votre enseignant le permet, partagez le résultat final sur les médias sociaux avec le hashtag #mBot2inclass• Réflexion : De quoi êtes-vous le plus fier ? Qu'est-ce que vous aimeriez améliorer dans votre robot ? |

1. Introduction (5 min)

Étape 1 : Introduction

Cette étape se compose de deux parties :

1. Les capteurs à ultrasons dans la vie quotidienne
2. Comment fonctionne un capteur à ultrasons ?

1. Les capteurs à ultrasons dans la vie quotidienne

Les ultrasons sont des sons à très haute fréquence. Si haute que les gens ne peuvent pas l'entendre. Grâce à des microphones spéciaux, ce son peut être enregistré et affiché sur un ordinateur : il ressemble à des vibrations sous forme d'onde. Les ultrasons ayant une fréquence très élevée, la forme d'onde présente des vallées et des pics très rapprochés. Les ultrasons peuvent être utiles. Par exemple, vous pouvez utiliser les vibrations des ultrasons pour nettoyer des objets, ou vous pouvez également les utiliser pour détecter des objets.

Les dauphins et les chauves-souris utilisent les ultrasons de cette façon. Ils émettent un son ultrasonique, comme un court cri. Lorsque ce son rebondit sur quelque chose, ils captent avec leurs oreilles les vibrations (écho) qui reviennent. C'est ainsi qu'ils savent où se trouvent les autres animaux, par exemple, ou si un obstacle se trouve sur leur chemin. Le radar que nous avons aujourd'hui utilise le même principe, mais avec des ondes radio au lieu du son.

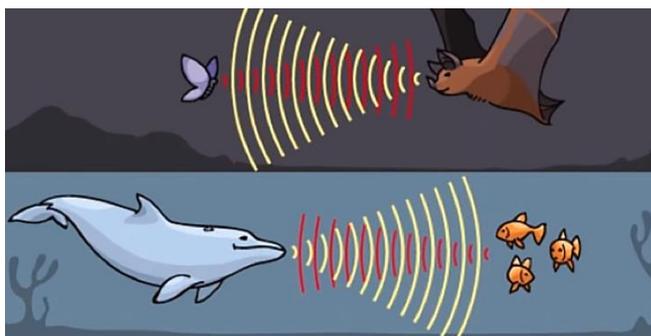
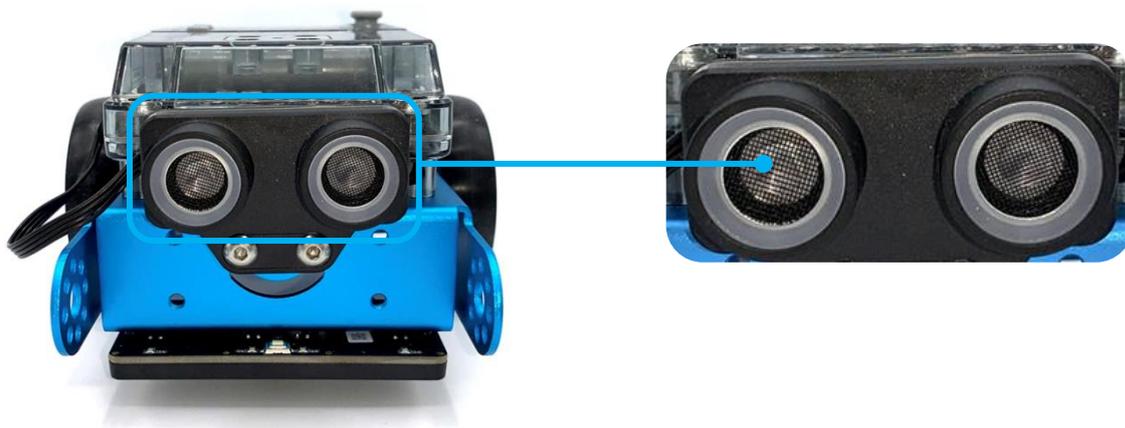


Figure 2. Exemple de chauve-souris et de dauphin

2. Comment fonctionne un capteur à ultrasons ?

Le mBot2 possède un émetteur et un récepteur d'ultrasons. Ceux-ci sont situés à l'avant du mBot2 dans les petits cylindres qui peuvent être considérés comme des « yeux ». Ce capteur envoie de courtes rafales de sons ultrasoniques et détecte tout écho. Le mBot2 s'approche-t-il d'un obstacle ? Ensuite, les ultrasons sont renvoyés vers le mBot2. En fonction du temps que met le son à revenir vers le capteur, le mBot2 calcule en interne la distance qui le sépare de celui-ci.

Ces données peuvent être utilisées pour décider de la manière dont le robot doit réagir : vous le dites au robot en le programmant. Une telle action peut être, par exemple, que le mBot2 doit s'arrêter de conduire ou faire un virage.



2. Travaux pratiques (10 min)

Étape 2 : Travaux pratiques

Cette étape se compose de deux parties :

1. Apprendre à connaître les différents blocs de code du capteur à ultrasons
2. Recréer et tester quelques exemples de programmation du capteur à ultrasons.

1. Apprendre à connaître les différents blocs de code du capteur à ultrasons

Dans mBlock, il y a plusieurs blocs de code pour le capteur à ultrasons que vous pouvez utiliser dans vos programmes. Pour cela, vous devez ajouter l'extension 'Ultrasonic Sensor 2' de la bibliothèque d'extension. Vous trouverez ces blocs de code dans la catégorie 'Ultrasonic Sensor' de la zone des blocs dans mBlock. Ces blocs de codage sont verts.



Dans le tableau ci-dessous, vous verrez certains de ces blocs de code pour programmer le capteur à ultrasons

Bloc de code :

ultrasonic 2 1 distance to an object (cm)

Avec ce bloc de code, vous mesurez la distance entre le capteur et un obstacle. La plage de détection est comprise entre 3 et 300 cm. Vous pouvez utiliser la valeur de la distance pour que le robot effectue une certaine action. Par exemple, vous pouvez empêcher le mBot2 d'entrer en collision avec des obstacles.

Dans l'exemple de programmation ci-dessous, le mBot2 avance. Lorsque le mBot2 se trouve à moins de 10 cm d'un obstacle, le robot effectue un virage de 90° vers la gauche. Puis le mBot2 avance à nouveau.

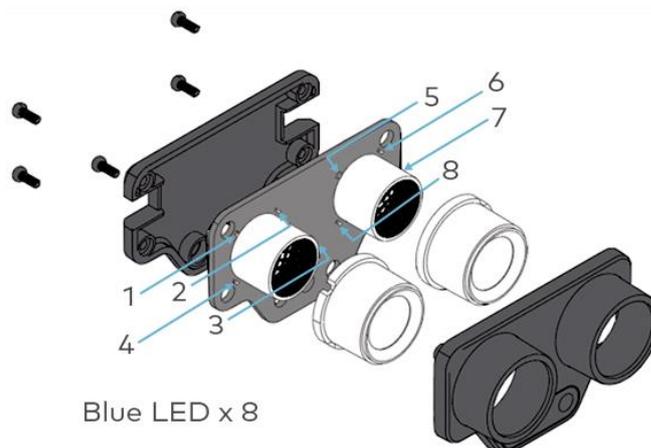
```

when button A pressed
  moves forward at 50 RPM
  forever
    if ultrasonic 2 1 distance to an object (cm) < 10 then
      turns left 90° until done
      moves forward at 50 RPM
  
```

Bloc de code :

ultrasonic 2 1 increases ambient light all brightness by 20 %

Le capteur à ultrasons de mBot2 possède huit lumières LED bleues. Vous pouvez utiliser ces lumières pour que le mBot2 montre une certaine « émotion », par exemple, ou comme une communication visuelle non-verbale générale. Faites en sorte que les lumières LED brillent très fort lorsque le mBot2 est heureux et moins fort lorsqu'il est triste. Il existe également un bloc de code spécifique pour afficher des « émotions » prédéfinies. Dans l'image ci-dessous, vous pouvez voir où se trouvent exactement les lumières LED.



Dans mBlock, il y a plusieurs blocs de code qui vous permettent de programmer la luminosité des lumières LED : vous pouvez régler la luminosité des LED en diminuant ou en augmentant la luminosité d'un certain pourcentage. Vous pouvez également régler la luminosité directement sur un pourcentage spécifique. La plage va de 0 % à 100 %. Ces deux opérations peuvent être effectuées soit pour une seule LED, soit pour toutes en même temps. Dans l'exemple de programmation ci-dessous, la luminosité de la lampe LED 1 est augmentée de 50 %.



2. Recréer et tester quelques exemples de programmation du capteur à ultrasons

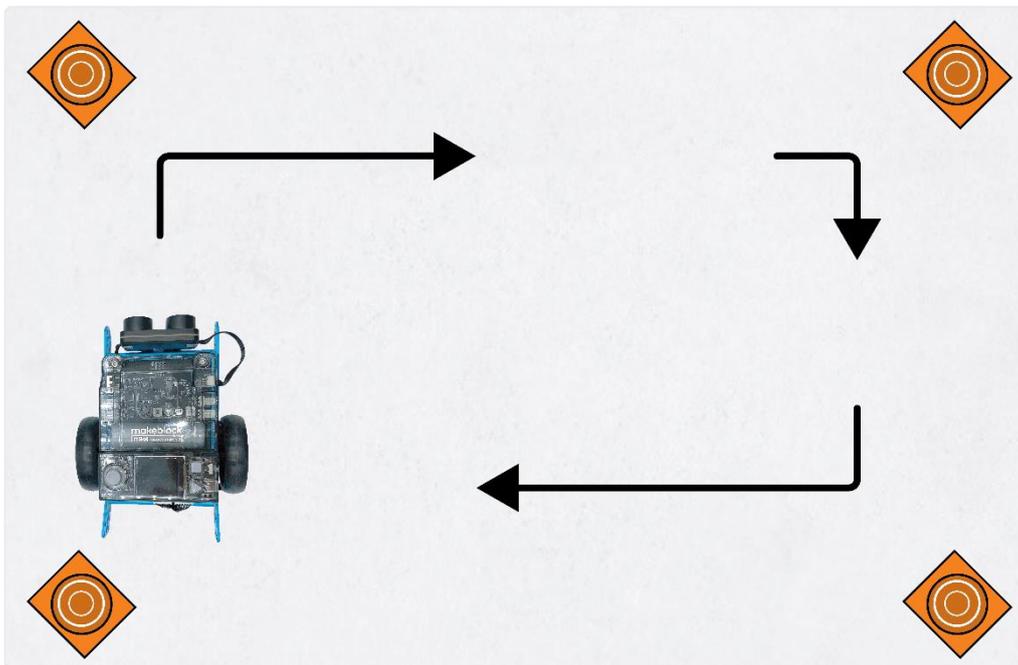
Dans le tableau ci-dessus, chaque bloc de code du capteur à ultrasons est accompagné d'un exemple de programmation. Vous allez recréer ces exemples de programmation dans mBlock et les tester. Pour un exemple de programmation, proposez vos propres idées sur la manière d'étendre l'application.

3. Test (25 min)

Étape 3 : Test

Vous avez déjà appris beaucoup de choses sur le capteur à ultrasons du mBot2. Vous allez maintenant créer votre propre programme informatique dans mBlock en utilisant le capteur à ultrasons.

Dans ce travail, vous allez faire en sorte que le mBot2 se déplace sur une trajectoire fermée. Pour ce faire, vous placez un objet à chaque coin. Vous programmez le mBot2 de manière à ce qu'il fasse un tour à chaque objet et revienne finalement au point de départ. Regardez l'image ci-dessous.



Utilisez les connaissances que vous avez acquises à l'étape 2 de cette leçon. Bien sûr, vous pouvez aussi expérimenter les différentes possibilités de programmation de mBlock. Pour réaliser ce travail, il est utile de recourir au plan par étapes suivant. Avez-vous une idée de ce que vous voulez faire ? Si c'est le cas, discutez-en d'abord avec votre professeur pour savoir si cela est possible.

| | Explication |
|---|--|
| Étape 1 : Que voulez-vous faire ? | <ul style="list-style-type: none"> • Dans quel sens le mBot2 doit-il se déplacer ? |
| Étape 2 : De quoi avez-vous besoin ? | <ul style="list-style-type: none"> • De quoi avez-vous besoin en plus du mBot2 ? |
| Étape 3 : De quels blocs de code avez-vous besoin pour faire se déplacer le mBot2 ? | <ul style="list-style-type: none"> • Comment faire pour que le mBot2 change de direction à chaque coin ? • De quels blocs de code avez-vous besoin pour ce faire ? |



| | |
|---------------------------------|---|
| | <ul style="list-style-type: none">• Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML)• Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock |
| Étape 4 : Test et mise en œuvre | <ul style="list-style-type: none">• La première version est-elle prête ? Testez-la ! Pendant le test, notez les points à améliorer• Travaillez sur les points d'amélioration jusqu'à ce que votre mBot2 traverse votre parcours sans erreur• Vous avez réussi ? Filmez le résultat final et demandez à votre professeur si vous pouvez le publier sur les réseaux sociaux avec le hashtag #mBot2inclass |

4. Récapitulatif (5 min)

Étape 4 : Récapitulatif

Avez-vous réussi à faire faire un tour à votre mBot2 ?

Dans cette leçon, vous avez découvert les capteurs à ultrasons et les situations où vous pouvez les rencontrer dans la vie quotidienne. Vous savez comment programmer le capteur à ultrasons du mBot2, par exemple, pour empêcher le mBot2 de foncer dans les obstacles. Vous savez également comment utiliser les lumières LED dans le capteur à ultrasons pour que le mBot2 montre une certaine émotion.

Il est maintenant temps de procéder à une courte réflexion. Réfléchissez par vous-même puis discutez-en avec le groupe :

- Qu'est-ce qui s'est bien passé, à votre avis ?
- Qu'est-ce qui pourrait être amélioré ?
- Quelles sont les parties de la leçon que vous avez trouvées faciles et celles que vous avez trouvées plus difficiles ?
- Sur quoi souhaitez-vous plus d'explications ?
- Qui pourrait vous aider ?

Leçon 5 : Visites guidées

Sujet : STEAM **Classe(s) :** trop compliqué pour des CM2
Durée : 45 minutes **Difficulté :** Débutant

★ Objectifs de la leçon

À la fin de cette leçon, les élèves sauront :

- Comment fonctionnent les capteurs de couleurs
- Leurs applications dans la vie quotidienne et en robotique

★ Aperçu

Imaginez que vous êtes dans une ville étrangère, que vous faites un tour des plus beaux sites à voir. Peut-être avez-vous déjà fait un tel voyage, et si ce n'est pas le cas, vous avez probablement vu des images à la télévision et sur Internet. Dans cette leçon, le mBot2 est transformé en une telle voiture de tourisme.

Parce qu'il serait très ennuyeux pour un chauffeur de faire le même trajet tous les jours (et d'expliquer chaque vue encore et encore), nous allons programmer ce bus. Ainsi, les touristes peuvent profiter d'un trajet agréable sans que le conducteur s'ennuie avec des tâches répétitives.

Focus

À la fin de cette leçon, les élèves seront capables de :

- Faire en sorte que le mBot2 suive une ligne
- Faire en sorte que le mBot2 effectue des actions basées sur les couleurs

Liste de contrôle avant la leçon

De quoi avez-vous besoin ?

- PC ou ordinateur portable (avec sortie USB) avec le logiciel mBlock installé, la version web (également pour ChromeBook), ou une tablette avec l'application mBlock installée
- mBot2 avec un CyberPi
- Câble USB-C ou un dongle Bluetooth Makeblock

Plan de cours

Cette leçon se compose de quatre étapes et dure au total 45 minutes.

| Durée | Contenu |
|------------|--|
| 5 minutes | 1. Échauffement <ul style="list-style-type: none">• Les capteurs de couleurs dans la vie quotidienne• Comment fonctionnent les capteurs de couleurs? Quelle est la spécificité du capteur de couleurs du mBot2? |
| 10 minutes | 2. Travaux pratiques <ul style="list-style-type: none">• Apprendre à connaître les différents blocs de code du capteur quadruple RVB• Recréer et tester quelques exemples de programmation du capteur quadruple RVB• Comment le mBot2 peut-il suivre une ligne? |
| 25 minutes | 3. Test <ul style="list-style-type: none">• Écrire votre propre programme pour le robot |
| 5 minutes | 4. Récapitulatif <ul style="list-style-type: none">• Showtime : montrez ce que vous avez fait avec votre robot dans un film court et amusant pour en discuter plus tard• Si votre enseignant le permet, partagez le résultat final sur les médias sociaux avec le hashtag #mBot2inclass• Réflexion: De quoi êtes-vous le plus fier? Qu'est-ce que vous aimeriez améliorer dans votre robot? |

1. Introduction (5 min)

Étape 1 : Introduction

Cette étape se compose de deux parties:

1. Les capteurs de couleurs dans la vie quotidienne
2. Comment fonctionnent les capteurs de couleurs ? Quelle est la spécificité du capteur de couleurs du mBot2 ?

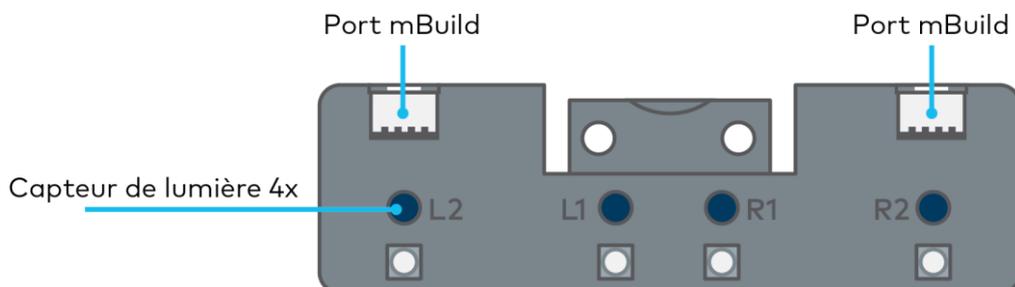
1. Les capteurs de couleurs dans la vie quotidienne

Les capteurs de couleur peuvent détecter l'intensité de la lumière et les couleurs. Vous rencontrerez ce type de capteur dans de nombreuses situations différents de la vie quotidienne. Par exemple, un robot d'entrepôt peut trouver le bon itinéraire dans l'entrepôt en fonction des différentes couleurs du sol. Ce type de capteur est également utilisé lorsqu'une correspondance précise des couleurs est requise, par exemple pour la peinture des véhicules. Un appareil de mesure doté d'un capteur de couleurs, le colorimètre, peut vérifier si la bonne couleur est utilisée pour peindre un véhicule au pistolet. Pouvez-vous, vous et les camarades de votre groupe, imaginer d'autres applications ?

2. Comment fonctionnent les capteurs de couleurs ? Quelle est la spécificité du capteur de couleurs du mBot2 ?

Un capteur de couleurs est constitué de trois capteurs (internes) différents qui mesurent la lumière réfléchiée par un objet en fonction de l'intensité des valeurs de couleur rouge, verte et bleue. Les capteurs de couleur sont donc également appelés « capteurs RVB ». Toute couleur est décodée en trois valeurs par le capteur - ainsi, un « mélange » spécifique de ces valeurs représente une couleur.

Le mBot2 possède quatre de ces capteurs RVB intégrés dans un seul capteur. Regardez l'image ci-dessous.



C'est le capteur Quad RGB qui est sur le mBot2. Les capteurs RVB sont nommés L1, L2, R1 et R2 (L pour le côté gauche, R pour le côté droit). Ils détectent automatiquement les valeurs RVB de la couleur réfléchiée et comparent le mélange de valeurs en interne avec différents pré-réglages pour les couleurs. Cela facilitera grandement le codage, car le capteur peut alors signaler la

couleur et vous ne devez pas vérifier les codes RVB vous-même. Le capteur peut détecter 6 couleurs différentes, plus le noir et le blanc. Au lieu de la couleur, il peut rendre compte des nuances de gris ou des lignes et des jonctions.

Les capteurs quadruples RVB permettent au mBot2 de suivre une ligne au sol et de réagir à des marqueurs de différentes couleurs. Le capteur quadruple RVB est placé à l'avant du mBot2. Sur la face inférieure, vous verrez les quatre capteurs. En haut, vous trouverez les noms de chacun d'entre eux, ce qui vous permettra d'identifier et de programmer les capteurs de manière spécifique. Regardez les images ci-dessous.



Capteur quadruple RVB, vue de face

Vue de dessus

Vue d'en bas

Le petit bouton situé sur la partie supérieure du capteur sert à l'étalonnage. Si vous double-cliquez dessus, les LED se mettront à clignoter et vous pourrez faire « glisser » le mBot2 avec le capteur sur la ligne à suivre. Cela permettra de calibrer le capteur pour différencier la ligne de l'arrière-plan. Si vous utilisez une ligne noire sur une surface blanche, cela n'est normalement pas nécessaire.

La carte fournie avec le mBot2 a des couleurs marquées à l'intérieur de la piste elle-même pour réagir aux codes de couleur. Cela peut amener le capteur à associer une couleur vive (comme le jaune) à l'arrière-plan au lieu de la piste. Donc, comme premier exercice, veuillez calibrer le capteur sur le code couleur jaune, il reconnaîtra alors automatiquement et de manière fiable cette couleur et toutes les couleurs « plus sombres » comme étant la ligne.

Lorsque vous programmez le mBot2, vous utiliserez un ou plusieurs de ces quatre capteurs pour vérifier la présence de certaines couleurs sur le sol tout au long de votre parcours. Vous pouvez demander au mBot2 d'exécuter une commande lorsqu'une couleur est repérée. Par exemple, « stop » quand c'est rouge et « aller à gauche » quand c'est vert.

2. Travaux pratiques (10 min)

Étape 2 : Travaux pratiques

Cette étape se compose de trois parties :

1. Apprendre à connaître les différents blocs de code du capteur quadruple RVB
2. Recréer et tester quelques exemples de programmation du capteur quadruple RVB
3. Comment le mBot2 peut-il suivre une ligne ?

1. Apprendre à connaître les différents blocs de code du capteur quadruple RVB

Dans mBlock, il y a plusieurs blocs de code que vous pouvez utiliser pour programmer le capteur quadruple RVB. Vous trouverez ces blocs de code dans la catégorie « Quadruple RVB » de la zone des blocs dans mBlock. Ces blocs de codage sont verts.



Dans le tableau ci-dessous, vous verrez certains de ces blocs de code pour programmer le capteur quadruple RVB. Les quatre premiers blocs de code servent à l'identification (simple) de la ligne. Ils peuvent être utilisés pour le suivi de ligne simple et avancé, mais aussi pour la détection des croisements et des virages serrés (90°). L'étalonnage du capteur à différentes pistes de couleur est brièvement mentionné dans la section ci-dessus. Les LED bleues situées sur la partie supérieure du capteur indiquent la détection de la ligne/du fond : si elles sont éteintes, elles détectent une couleur sombre (ligne), si elles sont bleu brillant, elles indiquent une couleur claire (fond). L'état de la LED indique donc la réflectivité du sol ou de la carte.

Bloc de code :



Ces blocs de code ne prennent en compte que les deux capteurs intérieurs L1 et R1 pour la détection des lignes. Ces blocs sont destinés à un simple suivi de ligne.

Utilisez le premier bloc dans les déclarations conditionnelles. Utilisez le deuxième bloc si vous voulez reporter la valeur à l'écran ou l'enregistrer dans une variable.

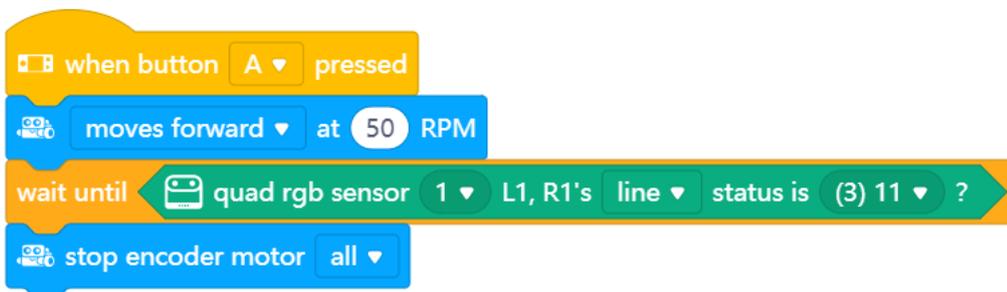
Si le capteur est au-dessus d'une ligne noire, les deux LED intérieures doivent être éteintes. Comme chaque capteur (L1, R1) peut détecter soit la ligne, soit l'arrière-plan avec ce bloc, il y a 4 combinaisons possibles : elles seront représentées par des nombres de 0 à 3 (pour référence, le bloc montre également le motif binaire) :

| L1 | R1 | Signification | État (Décimal) |
|----|----|---|----------------|
| | | Capteur sur fond blanc, les deux LED sont allumées | 0 |
| | | R1 détecte une ligne noire | 1 |
| | | L1 détecte une ligne noire | 2 |
| | | Les deux capteurs intérieurs détectent la ligne noire | 3 |

Rappelez-vous que les LED indiquent l'état de l'arrière-plan (allumé ou logique « vrai » = arrière-plan détecté), tandis que ce bloc de code détecte la couleur de la ligne (logique « vrai » = ligne détectée).

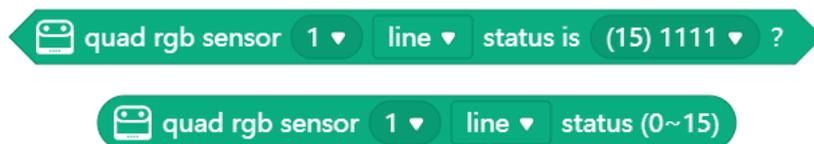
Si vous voulez utiliser plus d'un capteur quadruple RVB, vous pouvez indiquer au programme quel capteur utiliser avec le premier chiffre dans tous les blocs de code suivants. Regardez l'image ci-dessous. Le numéro 1 indique le premier capteur connecté, le numéro 2 le deuxième capteur connecté et ainsi de suite.

L'exemple suivant utilise la fonction de détection de ligne de ce bloc et arrête le robot dès qu'il arrive sur une ligne noire. Cet exemple sera changé plus tard en suivi de ligne.



Étant donné que la réaction pendant la conduite peut être critique en termes de temps, il est important d'utiliser le mode Upload. Vous pouvez essayer par vous-même et voir la différence si vous changez de mode.

Bloc de code :



Utilisez le premier bloc dans les déclarations conditionnelles. Utilisez le deuxième bloc si vous voulez reporter la valeur à l'écran ou l'enregistrer dans une variable.

Ces blocs suivent exactement la même logique que les précédents, sauf qu'ils ont une zone de détection plus grande et peuvent identifier les croisements tout en continuant à suivre les lignes. Rappelez-vous que les blocs vérifient l'état de la ligne, donc si l'un des quatre capteurs identifie une ligne (sombre), le chiffre binaire correspondant sera mis à « 1 » (logique vrai), mais les LED s'éteindront (pas de réflexion de l'arrière-plan sur ce capteur).

Avec maintenant 4 capteurs individuels, la gamme passe à 16 états différents. Le tableau suivant présente les états les plus pertinents :

| Modèle binaire | Signification | État (Décimal) |
|----------------|--|----------------|
| 0000 | les 4 capteurs sur fond blanc (pas de piste) | 0 |
| 0010 | seulement R1 sur la ligne noire | 2 |
| 0100 | seulement L1 sur la ligne noire | 4 |
| 0110 | capteurs internes R1 et L1 sur la ligne noire | 6 |
| 0111 | virage serré à droite (L1 et R1 sur la ligne noire, jonction à droite ; jonction en forme de L à droite) | 7 |
| 1110 | virage serré à gauche (L1 et R1 sur la ligne noire, jonction à gauche ; jonction en forme de L à gauche) | 14 |
| 1111 | Jonction en forme de T (L1 et R1 sur la ligne noire, jonction à droite et à gauche) | 15 |

Une jonction en forme de X ne peut être détectée qu'en passant sur la jonction en T et en vérifiant si la ligne centrale continue.

L'exemple de programme suivant fera avancer le robot tant qu'il se trouve sur une ligne noire avec les deux capteurs intérieurs. Il s'arrête s'il sort de la voie, s'il détecte une jonction ou si la ligne s'arrête soudainement :

```

when button B pressed
  forever
    if quad rgb sensor 1 line status is (6) 0110 ? then
      moves forward at 50 RPM
    else
      stop encoder motor all
  
```

Bloc de code :

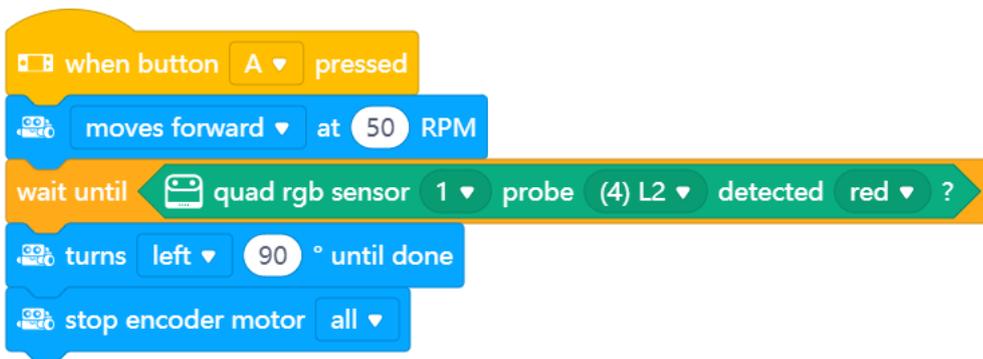
```

quad rgb sensor 1 probe (1) R2 detected line ?
  
```

Ce bloc de code permet de sélectionner l'un des quatre capteurs RVB (L2, L1, R1, R2) et de déterminer s'il doit reconnaître une ligne, un fond, du blanc, du noir ou l'une des cinq couleurs (rouge, vert, bleu, jaune, cyan, violet). Pour la détection et le suivi des lignes, les blocs précédents sont recommandés, car ils vérifient plusieurs capteurs à la fois (plus rapide, plus

précis puisque le robot ne se déplace pas entre les lectures). Mais vous pouvez utiliser ce bloc pour détecter par exemple un marqueur de couleur rouge à gauche du robot - lorsque le robot suit une ligne.

Dans l'exemple de programmation suivant, si vous appuyez sur le bouton B, le mBot2 va tout droit (en supposant qu'il suit une ligne) et fait un virage à 90° lorsqu'un marqueur rouge est détecté sur le côté gauche de la piste (imaginaire).



```
when button A pressed
  moves forward at 50 RPM
  wait until quad rgb sensor 1 probe (4) L2 detected red
  turns left 90 ° until done
  stop encoder motor all
```

Bloc de code :



```
quad rgb sensor 1 deviation (-100~100)
```

Ce bloc de code mesure l'écart du robot par rapport à une piste noire. Vous pouvez l'utiliser pour un suivi de ligne avancé et fluide. Le bloc de code qui suit la ligne précédente utilise une triple distinction : sur la voie, hors de la voie ou hors de la voie (plus la détection de la jonction dans certains cas). Mais dans la vie de tous les jours, en faisant du vélo, vous adapteriez plus précisément la direction à la courbe. Ce bloc permet un suivi de ligne proportionnel : plus le robot s'éloigne de la piste, plus il se dirige dans la direction opposée. Si le décalage par rapport à la piste est de 0, la direction est également de 0 (droite).

Dans l'étape 3, nous verrons comment utiliser ce bloc pour suivre une ligne « en douceur » en programmant un suiveur de ligne proportionnel. L'exemple de programme ci-dessous affiche l'écart à l'écran, ce qui vous permet d'essayer de déplacer le robot sur la piste à la main dans un premier temps :



```
when CyberPi starts up
  forever
    show quad rgb sensor 1 deviation (-100~100) at center of screen by middle pixel
```

2. Recréer et tester quelques exemples de programmation du capteur quadruple RVB

Dans le tableau ci-dessus, chaque bloc de code du capteur quadruple RVB est accompagné d'un exemple de programmation. Vous allez recréer ces exemples de programmation dans mBlock et les tester. Pour un exemple de programmation, proposez votre propre modification du code.

3. Comment le mBot2 peut-il suivre une ligne ?

La base de la mission de l'étape 3 de cette leçon est que le mBot2 va suivre la piste sur une carte, comme un bus touristique. À cette fin, vous devrez utiliser les exemples de code expliqués et testés jusqu'à présent et les modifier en conséquence.

Utilisez d'abord ce bloc de codage :



Si le robot est sur la piste (L1 et R1 détectent tous deux une ligne, valeur de code3), le robot doit rouler en ligne droite. Si un seul des deux capteurs intérieurs identifie une ligne, il doit alors tourner à gauche ou à droite en conséquence.

Utilisez ces blocs de conduite (seul le virage à droite est indiqué) :



Il suffit de démarrer les moteurs selon les besoins, et le bloc de code suivant sera exécuté immédiatement. Étant donné que vous ne savez pas à quelle distance le robot se trouve ou quand un prochain virage va se produire, les commandes de conduite ne peuvent pas contenir de temps ou de distance de conduite (de même pour les virages). L'itération suivante d'une boucle de programme vérifiera les nouvelles données du capteur et sélectionnera la conduite en conséquence.

Si vous utilisez la carte fournie, veuillez à calibrer le capteur comme décrit sur la couleur la plus claire (section jaune de la piste).

3. Test (25 min)

Étape 3 : Test

Vous avez déjà appris beaucoup de choses sur le capteur quadruple RVB du mBot2. Vous allez maintenant créer votre propre programme informatique dans mBlock en utilisant le capteur quadruple RVB.

Dans cette mission, vous allez transformer le mBot2 en un bus touristique. Le bus touristique fait le tour des points d'intérêt d'une ville. Dans la boîte du mBot2 il y a une carte avec une piste. Le bus suit la ligne noire sur la carte. Dans la partie précédente, vous avez déjà programmé un simple robot suiveur de ligne, il est maintenant temps d'ajouter des améliorations.

Pour faciliter la compréhension du programme, vous pouvez « construire » vos propres blocs de codage. Toute la partie du programme qui suit les lignes peut être ajoutée à une commande simple, par exemple « simple ligne ». Le bouton rouge/rose « Mes blocs » dans le menu du code permet de définir un bloc personnalisé et de structurer votre code de manière agréable. L'exemple de programme ci-dessous utilise des blocs personnalisés.





En plus de suivre la piste, le bus effectue une action à certains des points de vue. Les sites d'intérêt sont indiqués par une zone colorée (rouge, jaune, verte et bleue).

Vous pouvez imaginer les actions que le bus doit effectuer. Le bus va-t-il faire le tour de Paris ? Alors il pourrait être amusant de jouer l'hymne national français. Ou les cloches de Big Ben si le bus traverse Londres.

Bien sûr, vous pouvez aussi faire un peu plus simple. Dans l'exemple ci-dessous, vous demandez au bus de faire demi-tour au niveau du marqueur rouge et de réaliser différentes activités à d'autres marqueurs de couleur.

```
if [quad rgb sensor 1] probe [(1) R2] detected [red] ? then  
  turns left 360° until done  
  moves forward 6 cm until done
```

Utilisez les connaissances que vous avez acquises à l'étape 2 de cette leçon. Bien sûr, vous pouvez faire de nombreuses expériences vous-même avec les différentes possibilités de programmation dans mBlock5. Êtes-vous coincé ? Vous pouvez ensuite utiliser les exemples de programmation ci-dessous. Vous pouvez le copier et l'adapter à vos propres souhaits et préférences.

```

when joystick middle pressed ▼
repeat until button B ▼ pressed?
  simple_line_follow
  if quad rgb sensor 1 ▼ probe (2) R1 ▼ detected blue ▼ ? then
    moves forward ▼ at 10 RPM for 1 secs
  if quad rgb sensor 1 ▼ probe (2) R1 ▼ detected red ▼ ? then
    turns left ▼ 180 ° until done
    moves forward ▼ 4 cm ▼ until done
  if quad rgb sensor 1 ▼ probe (2) R1 ▼ detected green ▼ ? then
    turns left ▼ 90 ° until done
    wait 1 seconds
    turns right ▼ 180 ° until done
    wait 1 seconds
    turns left ▼ 90 ° until done
    moves forward ▼ 4 cm ▼ until done
  stop encoder motor all ▼
  
```

Ce programme démarre si vous appuyez sur B et s'arrête en appuyant sur A. Il effectue un simple suivi de ligne en utilisant un code personnalisé « Mes Blocs » et vérifie ensuite si le capteur R1 détecte soit du bleu (rouler très lentement), soit du rouge (faire demi-tour), soit du vert (regarder de chaque côté). Pour s'assurer que la couleur n'est pas reconnue à nouveau après que les activités ont été effectuées, le robot avance de 4 cm à chaque fois.

Le bloc « simple_line » est défini comme suit :

```

define simple_line_follow
if quad rgb sensor 1 L1, R1's line status is (3) 11 ? then
  moves forward at 50 RPM
if quad rgb sensor 1 L1, R1's line status is (1) 01 ? then
  turns right at 20 RPM
if quad rgb sensor 1 L1, R1's line status is (2) 10 ? then
  turns left at 20 RPM
if quad rgb sensor 1 L1, R1's line status is (0) 00 ? then
  moves backward at 10 RPM

```

Version avancée :

Dans l'étape précédente, l'écart par rapport à la ligne a été introduit. Ce bloc indique dans quelle mesure le robot s'écarte de sa trajectoire (de -100 à +100 ; 0 pour un centrage parfait) et peut être utilisé pour calculer la direction du robot. Il s'agit d'un suivi de ligne proportionnel - plus le robot s'écarte de sa trajectoire, plus il doit se diriger vers cette ligne. Si le robot est à droite de la trajectoire, l'écart est positif (le robot doit rouler à gauche) ; pour des valeurs négatives, le robot est à gauche de la trajectoire (il doit rouler à droite).

Pour ce faire, vous avez besoin d'une vitesse de base, c'est-à-dire la vitesse d'avancement sur une ligne droite. Pour braquer à gauche ou à droite, l'une des roues doit tourner plus vite que l'autre - plus la différence est grande, plus le braquage est important.

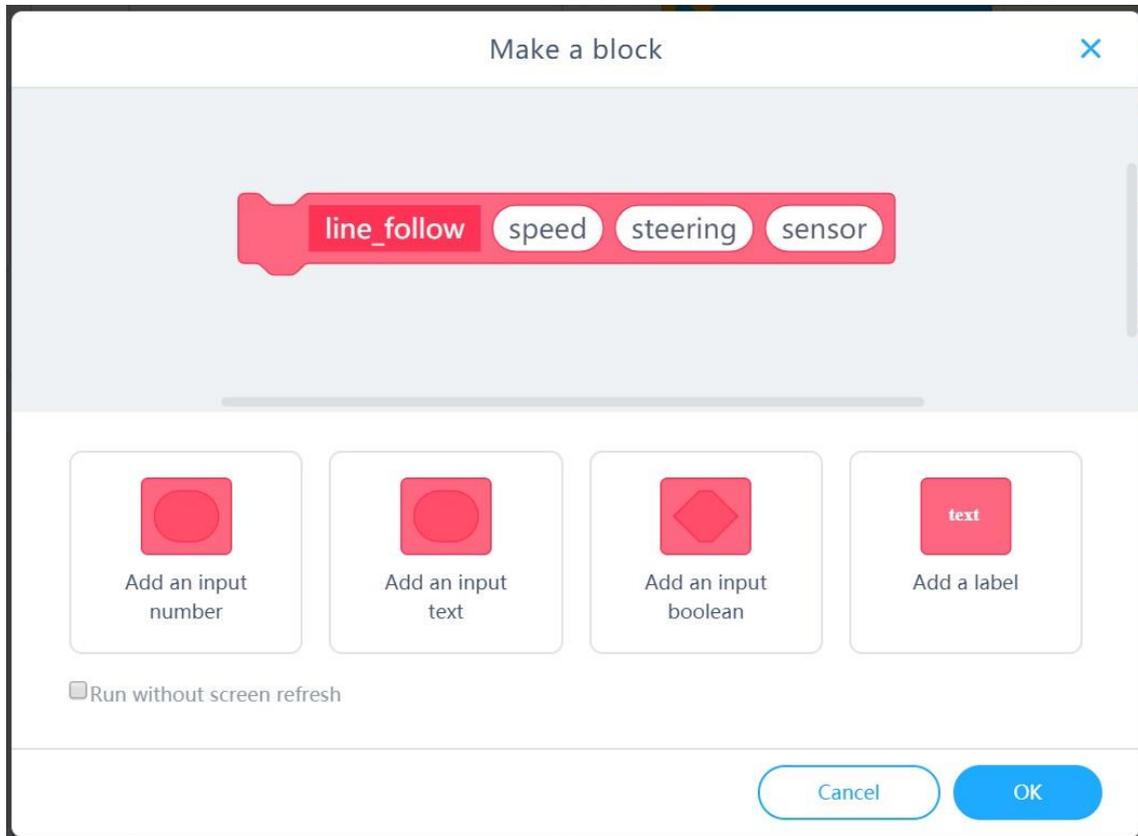
Pour calculer la direction, nous devons ajouter la déviation à la vitesse de base d'un moteur et la soustraire de l'autre. Pour permettre un ajustement du viseur, un facteur sera utilisé pour ajuster l'importance de la déviation sur la direction, puisque la plage de déviation elle-même est trop grande pour être utilisée directement pour contrôler les moteurs.

Mais à quel moteur faut-il ajouter ou soustraire la valeur calculée ? Une déviation positive signifie que le robot n'est pas sur la bonne voie sur le côté droit et qu'il doit tourner à gauche. Cela signifie que la roue droite doit tourner plus vite - nous ajoutons l'écart à l'échelle à la roue droite et le soustrayons à la roue gauche par conséquent.

$$\text{speed} + \text{steering} * \text{sensor}$$

speed - steering * sensor

Lors de la configuration de « My blocks », vous pouvez ajouter des entrées et les renommer pour leur donner un nom significatif :



La seule chose dont il faut s'occuper, c'est que le bloc de mouvement (illustré ci-dessous) nécessite des valeurs négatives pour le moteur de droite et des valeurs positives pour le moteur de gauche pour avancer (c'est parce qu'ils sont montés l'un en face de l'autre).

encoder motor EM1 rotates at 50 RPM, encoder motor EM2 rotates at 50 RPM

Pour finaliser le calcul, les valeurs du moteur de droite doivent être multipliées par -1. C'est la définition du bloc de suivi de ligne proportionnel :

```

define line_follow speed steering sensor
set left to -1 * speed + steering * sensor
set left to 1 * speed - steering * sensor
encoder motor EM1 rotates at left RPM, encoder motor EM2 rotates at right RPM
  
```



Dans la boucle principale d'un programme, vous utilisez simplement ce bloc avec une commande :

```
line_follow 50 0.6 quad rgb sensor 1 deviation (-100~100)
```

Cela appelle le sous-programme avec une vitesse de base de 50 et un facteur de direction de 0,6 et envoie également l'écart actuel de la lecture du capteur. Vous pouvez expérimenter un peu les valeurs : si le robot « oscille » (passe rapidement de la gauche à la droite, etc.), le facteur de direction est trop élevé ; s'il sort de la trajectoire même dans les courbes faibles, il est trop faible.

Pour réaliser ce travail, il est utile de recourir au plan par étapes suivant. Avez-vous une idée de ce que vous voulez faire ? Si c'est le cas, discutez-en d'abord avec votre professeur pour savoir si cela est possible.

| | Explication |
|---|---|
| Étape 1 : Que voulez-vous faire ? | <ul style="list-style-type: none"> • Dans quel sens le mBot2 doit-il se déplacer ? |
| Étape 2 : De quoi avez-vous besoin ? | <ul style="list-style-type: none"> • De quoi avez-vous besoin en plus du mBot2 ? |
| Étape 3 : De quels blocs de code avez-vous besoin pour faire se déplacer le mBot2 ? | <ul style="list-style-type: none"> • Comment faire pour que le mBot2 change de direction à chaque coin ? • De quels blocs de code avez-vous besoin pour ce faire ? • Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML) • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock |
| Étape 4 : Test et mise en œuvre | <ul style="list-style-type: none"> • La première version est-elle prête ? Testez-la ! Pendant le test, notez les points à améliorer • Travaillez sur les points d'amélioration jusqu'à ce que votre mBot2 traverse votre parcours sans erreur • Vous avez réussi ? Filmez le résultat final et demandez à votre professeur si vous pouvez le publier sur les réseaux sociaux avec le hashtag #mBot2inclass |

4. Récapitulatif (5 min)

Étape 4 : Récapitulatif

Avez-vous réussi à programmer le mBot2 comme une bus touristique autour de la piste ? Dans cette leçon, vous avez découvert les capteurs quadruples RVB et les situations où vous pouvez les rencontrer dans la vie quotidienne. Vous savez comment programmer le capteur quadruple RVB du mBot2 pour que le robot suive une ligne. Vous avez également appris à programmer le mBot2 pour qu'il effectue un parcours en fonction de différentes couleurs. Il est maintenant temps de procéder à une courte réflexion. Réfléchissez par vous-même puis discutez-en avec le groupe :

- Qu'est-ce qui s'est bien passé, à votre avis ?
- Qu'est-ce qui pourrait être amélioré ?
- Quelles sont les parties de la leçon que vous avez trouvées faciles et celles que vous avez trouvées plus difficiles ?
- Sur quoi souhaitez-vous plus d'explications ?
- Qui pourrait vous aider ?

Leçon 6 :

Conduite prudente

Sujet : STEAM
Durée : 45 minutes

Classe(s) : CM2 et plus
Difficulté : Débutant

★ Objectifs de la leçon

À la fin de cette leçon, les élèves seront capables de :

- Reconnaître et utiliser les blocs de code pour le gyroscope et l'accéléromètre dans CyberPi.
- Construire leur propre programme informatique dans mBlock pour mesurer les changements d'inclinaison du mBot2 et ajuster ses mouvements en conséquence

★ Aperçu

Un gyroscope mesure les mouvements de basculement, plus précisément la vitesse des mouvements de rotation/basculement. Un accéléromètre mesure la variation de la vitesse. Les deux capteurs fournissent des informations différentes sur la position dans l'espace.

En combinant les deux types de données, il est possible de calculer la position et le mouvement de véhicules, de robots ou même de sondes spatiales. Sur les routes cahoteuses, par exemple, les robots pourraient conduire plus lentement pour éviter de se renverser.



🔗 Focus

À la fin de cette leçon, les élèves sauront :

- Comment fonctionnent les gyroscopes et les accéléromètres et comment leurs mesures peuvent être combinées
- Comment faire en sorte qu'un robot réagisse aux changements soudains et détecte les routes accidentées ou cahoteuses ?

Liste de contrôle avant la leçon

De quoi avez-vous besoin ?

- PC ou ordinateur portable (avec sortie USB) avec le logiciel mBlock installé, la version web (également pour Chromebook), ou une tablette avec l'application mBlock installée
- mBot2 avec un CyberPi
- Câble USB-C ou un dongle Bluetooth Makeblock

Plan de cours

Cette leçon se compose de quatre étapes et dure au total 45 minutes.

| Durée | Contenu |
|------------|--|
| 5 minutes | 1. Introduction <ul style="list-style-type: none"> • Capteurs et données dans la vie quotidienne • Quels mouvements d'inclinaison le mBot2 peut-il détecter ? |
| 10 minutes | 2. Travaux pratiques <ul style="list-style-type: none"> • Se familiariser avec les différents blocs de programmation du gyroscope et de l'accéléromètre. • Recréer et tester quelques exemples de programmation des capteurs. |
| 25 minutes | 3. Test <ul style="list-style-type: none"> • Écrire votre propre programme pour le robot |
| 5 minutes | 4. Récapitulatif <ul style="list-style-type: none"> • Qu'avez-vous appris dans cette leçon ? • Showtime : montrez ce que vous avez fait avec votre robot dans un film court et amusant pour en discuter plus tard • Si votre enseignant le permet, partagez le résultat final sur les médias sociaux avec le hashtag #mBot2inclass • Réflexion: De quoi êtes-vous le plus fier ? Qu'est-ce que vous aimeriez améliorer dans votre robot ? |

1. Introduction (5 min)

Étape 1 : Introduction

Cette étape se compose de deux parties :

1. Capteurs et données dans la vie quotidienne
2. Quels mouvements d'inclinaison le mBot2 peut-il détecter ?

1. Capteurs et données dans la vie quotidienne

Comment les appareils tels que les smartphones identifient-ils leur position (tenue verticale, écran vers le haut/bas) ? Comment déterminer si un accident est suffisamment grave pour déclencher les airbags ? L'estimation de la position actuelle sans référence précise externe a été un défi en mer avec un ciel nuageux pendant des siècles et c'est toujours une tâche très importante pour les objets domestiques, les robots et même les sondes spatiales. Ils utilisent tous au moins deux capteurs différents, des gyroscopes et des accéléromètres. Un gyroscope fonctionne avec un cristal qui est sous tension électrique. Celui-ci convertit la direction de la force du mouvement de basculement en un signal électrique. Les gyroscopes mesurent la vitesse de rotation en degrés par seconde et, à l'aide des mathématiques, ils peuvent également estimer l'angle de rotation - mais ce dernier n'est pas aussi précis : les gyroscopes « dérivent » avec le temps. Les accéléromètres signalent le changement de position (comme l'augmentation ou la diminution de la vitesse) et, en ce qui concerne la gravité terrestre, ils signalent l'accélération gravitationnelle (par rapport à une chute libre). Les deux types de capteurs

peuvent mesurer sur chaque axe spatial, en indiquant la vitesse de rotation ou l'accélération sur les axes X, Y et Z.

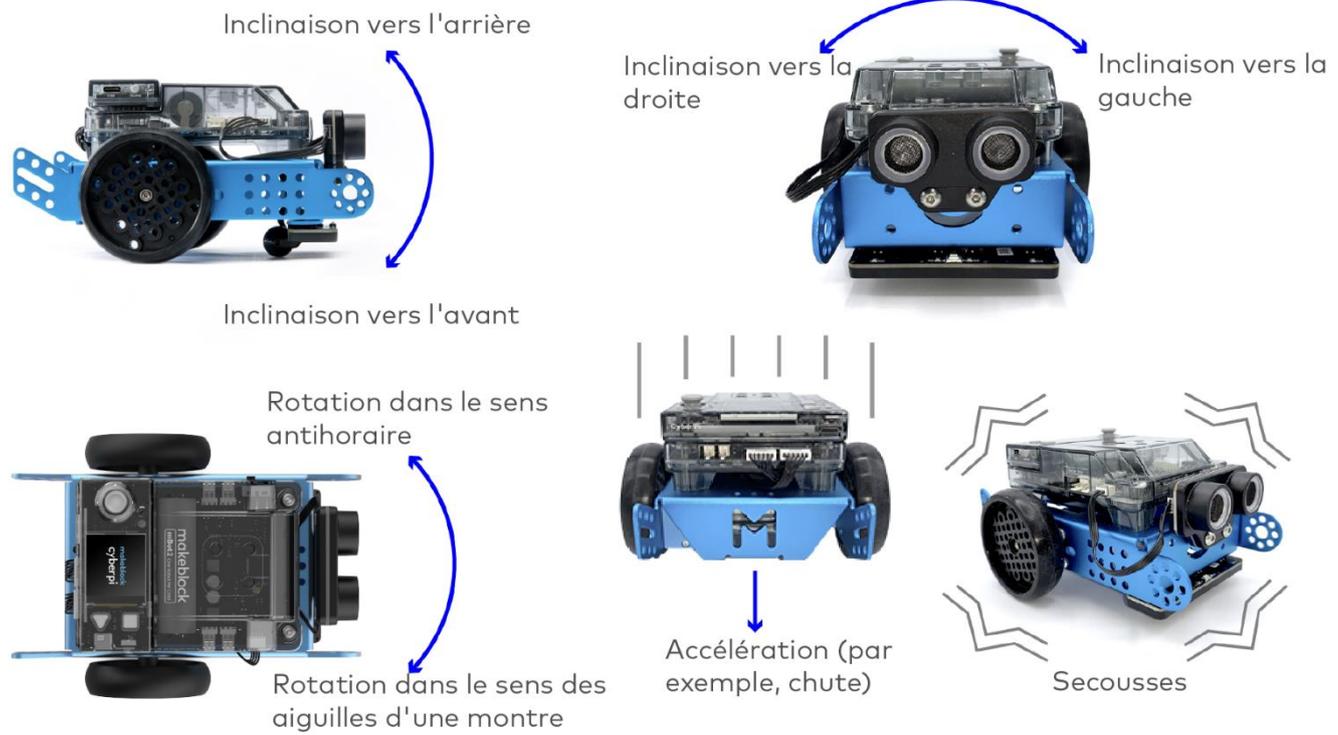
En combinant les données du gyroscope et de l'accéléromètre, on peut calculer la direction et la force des mouvements (en réalité, ce n'est pas super précis, il s'agit plutôt d'une très bonne estimation). En robotique, ces capteurs sont utilisés pour stabiliser les véhicules sur un sol irrégulier (ou en vol) et les aider à maintenir leur trajectoire programmée. Pour le mBot2, ces capteurs sont intégrés dans une petite puce sur le CyberPi.

Par conséquent, nous appellerons le capteur combiné « capteur de mouvement ».

2. Quels mouvements d'inclinaison le mBot2 peut-il détecter ?

Avant d'apprendre à programmer dans mBlock5 avec le capteur de mouvement, il est utile de savoir quels mouvements d'inclinaison le mBot2 peut mesurer. Vous pouvez afficher les mesures des mouvements d'inclinaison sur l'écran du CyberPi. Vous avez appris à le faire dans la leçon 2.

Le mBot2 peut détecter les mouvements d'inclinaison suivants :



2. Travaux pratiques (10 min)

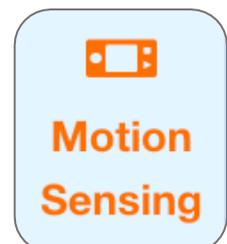
Étape 2 : Travaux pratiques

Cette étape se compose de deux parties :

1. Se familiariser avec les différents blocs de programmation du capteur de mouvement
2. Recréer et tester quelques exemples de programmation du capteur de mouvement

1. Se familiariser avec les différents blocs de programmation du capteur de mouvement

Dans mBlock il y a plusieurs blocs de code pour le capteur de mouvement que vous pouvez utiliser dans vos programmes. Ces blocs de code se trouvent dans la catégorie 'Motion Sensing' du champ de blocs dans mBlock. Ces blocs de codage sont orange.





Le tableau ci-dessous donne une explication détaillée des blocs de code ainsi qu'un exemple de code de programme pour les tester.

Bloc de code :

shaking strength

Ce bloc de code indique la force avec laquelle le mBot2 est secoué. Avec une valeur « 0 », le mBot2 n'est pas du tout secoué. Avec une valeur « 100 », un mouvement de secousse très fort est détecté. Dans l'exemple de programmation ci-dessous, la force de secousse du mBot2 est indiquée sur l'écran. Si le mBot2 cesse d'être secoué, l'affichage est vide.

```
when CyberPi starts up
  forever loop
    show shaking strength at center of screen by middle pixel
```

Bloc de code :

waving speed

Ce bloc de code est utilisé pour enregistrer la vitesse de secousse. La vitesse de secousse varie de 0 à 100. Ces unités ne représentent pas la vitesse en m/s ou en degrés/s, elles ne sont qu'une simple référence. Dans l'exemple de programmation ci-dessous, la vitesse de secousse du mBot2 est affichée à l'écran. Si le mBot2 cesse d'être secoué, l'affichage est vide.

```
when CyberPi starts up
  forever loop
    show waving speed at center of screen by middle pixel
```

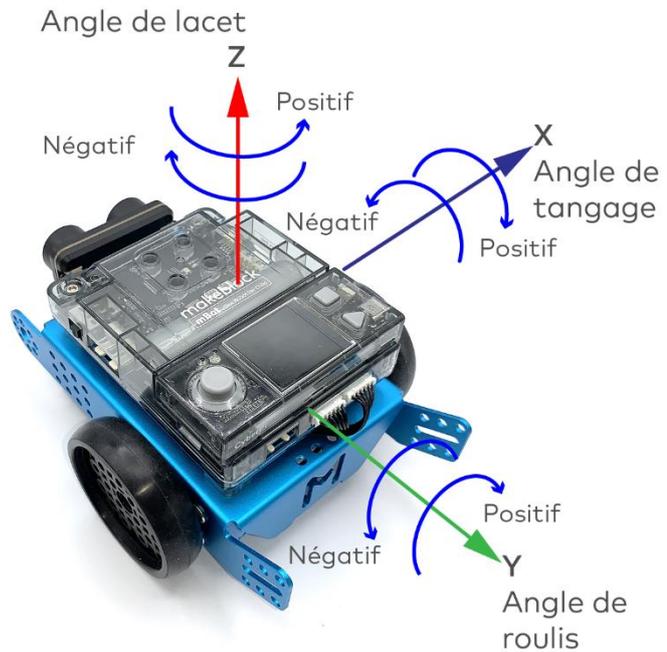
Bloc de code :

tilted forward angle (°)

Chaque inclinaison fait un mouvement à un certain angle dans une certaine direction. Lorsque vous spécifiez un angle, les axes X, Y et Z sont utilisés pour faire référence à la direction. Au lieu de ces termes mathématiques, on parle aussi de roulis, de tangage et de

lacet. Ces termes trouvent leur origine dans la navigation des avions. Le roulis est l'axe avant-arrière, le tangage fait référence aux virages de haut en bas (axe de gauche à droite) et le lacet est l'axe de haut en bas responsable du réglage de gauche à droite.

Regardez l'image ci-dessous.



Les angles sont donnés en degrés :

- Angle de roulis : -180° à $+180^\circ$
- Angle de lacet : -180° à $+180^\circ$
- Angle de tangage : -90° à $+90^\circ$

Dans le programme ci-dessous, l'angle de tangage est affiché sur la ligne supérieure et si un mouvement d'inclinaison vers l'avant ou vers l'arrière est détecté, les mots « avant » ou « arrière » apparaissent sur la dernière ligne. Pouvez-vous identifier le seuil de détection ?

```

when clicked
  forever
    show label 1 tilted backward(pitch) angle (°) at center of screen by middle pixel
    if tilted forward ? then
      show label 1 FORWARD at center of screen by middle pixel
    else
      if tilted backward ? then
        show label 1 BACKWARD at center of screen by middle pixel
      else
        show label 1 ○ at center of screen by middle pixel
  
```

Essayez d'expérimenter les différents paramètres du menu déroulant. Vous avez maintenant analysé le mouvement d'inclinaison sur l'axe tangage/x. Si vous passez à tilt_left/right, n'oubliez pas de changer les blocs booléens rapportés également :

```
when clicked
  forever
    show label 1 tilted right(roll) angle (°) at center of screen by middle pixel
    if tilted right ? then
      show label 1 RIGHT at center of screen by middle pixel
    else
      show label 1 ○ at center of screen by middle pixel
```

Bloc de code :

```
motion sensor x acceleration(m/s²)
```

Jusqu'à présent, nous avons étudié les mouvements d'inclinaison, mais les capteurs de mouvement intègrent un autre capteur, l'accéléromètre. Il mesure les changements de la vitesse du mBot2 dans le temps. Il peut s'agir d'une augmentation lente de la vitesse de conduite, d'un arrêt soudain ou même d'un choc latéral avec un autre véhicule. Comme ces mesures sont basées sur les axes x, y et z du robot, vous pouvez détecter un accident si vous recherchez un changement rapide de l'accélération. L'accéléromètre indique l'accélération en m/s^2 .

Dans l'exemple de programmation ci-dessous, l'accélération du mBot2 sur l'axe z est affichée à l'écran. Déplacez le mBot2 à la main et vérifiez l'affichage. Qu'affiche-t-il si le robot « s'écrase » sur votre main ou sur un livre ? Pouvez-vous changer le programme et simuler un impact latéral ?

```
when clicked
  forever
    print motion sensor z acceleration(m/s²) and move to a newline
```

Bloc de code :



Ce bloc de code montre la vitesse à laquelle le mBot2 tourne autour d'un certain axe. On l'appelle aussi la vitesse angulaire. La vitesse angulaire est indiquée en degrés par seconde (°/s).

Dans l'exemple de programmation ci-dessous, l'accélération du mBot2 sur l'axe z est affichée à l'écran.

```

when space key pressed
forever
  show angle speed around x axis(°/s) at center of screen by middle pixel
  
```

Bloc de code :



Ce bloc de code montre la vitesse à laquelle le mBot2 tourne autour d'un certain axe. L'angle est affiché en degrés (°) et est mesuré de manière relative. La position initiale est définie lorsque le mBot2 est allumé. Ces données sont calculées à partir de la vitesse de rotation rapportée par le gyroscope. Il peut dériver avec le temps et il n'est pas si stable, essayez d'incliner le mBot2 latéralement avec le programme ci-dessous et vérifiez les valeurs. Est-ce qu'elles reviennent à leur état initial ?

Dans l'exemple de programmation ci-dessous, l'angle auquel le mBot2 tourne autour de l'axe des y est indiqué sur l'écran.

```

when clicked
forever
  print rotated angle around y axis (°) and move to a newline
  
```

Il y a deux aspects importants : ce bloc gardera la trace des mouvements de rotation supérieurs à une révolution. Utilisez ce bloc pour compter combien de fois le robot a tourné. En regardant l'axe z, une rotation dans le sens des aiguilles d'une montre augmentera les valeurs rapportées par ce bloc :



Alors que l'angle de rotation autour de l'axe z diminue :



Après 3 rotations de 360°, le dernier bloc indiquera « -1080 », mais le bloc précédent indiquera simplement « 0 » (retour à la position initiale, en ignorant les rotations multiples). Si l'on effectue ensuite 4 rotations dans le sens inverse des aiguilles d'une montre (une fois de plus auparavant), l'angle « rotation dans le sens des aiguilles d'une montre » est à nouveau à 0°, tandis que « angle de rotation autour de l'axe z » indique maintenant 360° (toutes les figures sans dérive du gyroscope).

Si vous devez simplement prendre en compte de petits mouvements de rotation, pour les axes x et y, les blocs « angle d'inclinaison (___) » discutés précédemment fournissent une mesure plus robuste.

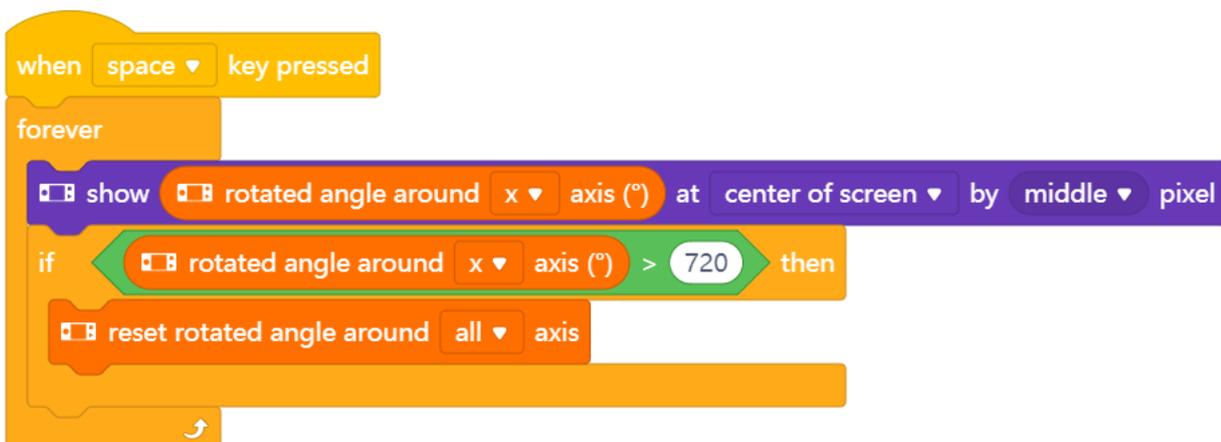
Il y a une exception : les rotations autour de l'axe de lacet/Z ne peuvent être mesurées qu'à l'aide du gyroscope. Si vous voulez mesurer les degrés, le mBot2 tournant à gauche ou à droite, vous devez travailler avec ces données.

Bloc de code :



Si vous voulez déterminer l'angle que fait le mBot2, il est plus facile de mesurer la variation par rapport à la valeur 0. Ce bloc de code réinitialise la valeur mesurée d'un angle à la valeur de départ 0, définissant ainsi une nouvelle position de départ pour les tours futurs.

Dans l'exemple de programmation ci-dessous, l'angle auquel le mBot2 tourne autour de l'axe des z est indiqué sur l'écran. Si l'angle est supérieur à 720°, la valeur mesurée de l'angle est mise à 0 et il est mesuré à nouveau.



```
when space key pressed
  forever loop
    show rotated angle around x axis (°) at center of screen by middle pixel
    if rotated angle around x axis (°) > 720 then
      reset rotated angle around all axis
```

Rappelez-vous qu'il existe deux blocs pour mesurer la rotation : l'un ne prend en compte que les rotations complètes, l'autre continue à compter. Ce bloc de réinitialisation remet à zéro tous les blocs « rotation autour de l'axe » (ou ceux sélectionnés). Il n'a pas d'effet sur le bloc

« rotation dans le sens des aiguilles d'une montre »/« rotation dans le sens inverse des aiguilles d'une montre ». Celui-ci a son propre bloc de réinitialisation :



reset yaw angle

Toute mise à zéro ne s'applique qu'aux rotations signalées par le capteur gyroscopique seul. L'angle d'inclinaison ___ se rapporte toujours à un plan parfaitement nivelé et perpendiculaire au centre de la terre et ne peut être mis à zéro à volonté.

2. Recréer et tester quelques exemples de programmation du capteur de mouvement

Dans le tableau ci-dessus chaque bloc de code du capteur de mouvement est accompagné d'un exemple de programmation. Vous allez recréer ces exemples de programmation dans mBlock et les tester.

Le capteur de mouvement du CyberPi enregistre les mouvements d'inclinaison du mBot2. Lorsque vous testez les exemples de programmation ci-dessus, vous n'avez donc besoin que du CyberPi. Si c'est plus pratique, retirez le CyberPi du mBot2 et effectuez vous-même les mouvements d'inclinaison avec le CyberPi pendant les tests. Vous pouvez tester les exemples de programmation en mode Live. Comme la batterie se trouve dans le bouclier mBot2, vous devrez connecter le CyberPi à l'ordinateur via le câble USB-C pour exécuter les programmes.

3. Test (25 min)

Étape 3 : Test

Vous avez déjà appris beaucoup de choses sur le capteur de mouvement du mBot2. Vous allez maintenant créer votre propre programme informatique dans mBlock en utilisant le capteur. Dans cette mission, vous allez créer une piste bosselée pour le mBot2. Vous pouvez fabriquer cette piste avec, par exemple, des (petits) cailloux ou de la pâte à modeler sur du carton. L'idée est que le mBot2 conduise sur le parcours. Si le mBot2 est secoué (fortement), il doit conduire plus lentement. Vous pouvez également placer un objet sur le mBot2 qui ne doit pas tomber pendant la conduite.

C'est une tâche difficile, n'est-ce pas ? Heureusement, vous ne devez pas tout inventer vous-même ! Vous trouverez ci-dessous un exemple de programmation que vous pouvez étendre. Dans cet exemple de programmation, le mBot2 ralentira s'il effectue des mouvements de bascule trop importants.

```

when space key pressed
  reset rotated angle around all axis
  repeat until button B pressed?
    if tilted right(roll) angle (°) > 10 or tilted right(roll) angle (°) < -10 then
      moves forward at 20 RPM
    else
      moves forward at 40 RPM
  stop encoder motor all

```

Utilisez les connaissances que vous avez acquises à l'étape 2 de cette leçon. Bien sûr, vous pouvez faire de nombreuses expériences vous-même avec les différentes possibilités de programmation dans mBlock5. Lorsque vous réfléchissez à cette mission, il est utile de recourir au plan par étapes suivant.

| | Explication |
|---|---|
| Étape 1 : Que voulez-vous faire ? | <ul style="list-style-type: none"> • Quel trajet voulez-vous que le mBot2 suivre ? • Quand le mBot2 doit-il ralentir ? • Voulez-vous que les capteurs du mBot2 affichent également des données sur l'écran ? |
| Étape 2 : De quoi avez-vous besoin ? | <ul style="list-style-type: none"> • De quoi avez-vous besoin en plus du mBot2 ? |
| Étape 3 : De quels blocs de code avez-vous besoin pour faire fonctionner le mBot2 ? | <ul style="list-style-type: none"> • Comment allez-vous faire fonctionner le mBot2 ? • Quels blocs de code utiliserez-vous ? • Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML) • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock |
| Étape 4 : De quelle manière voulez-vous afficher les données des capteurs sur l'écran ? | <ul style="list-style-type: none"> • Comment voulez-vous afficher les données sur l'écran ? • Quels blocs de code utiliserez-vous ? • Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML) |

| | |
|---------------------------------|--|
| | <ul style="list-style-type: none"> • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock |
| Étape 5 : Test et mise en œuvre | <ul style="list-style-type: none"> • La première version est-elle prête ? Testez-la ! Pendant le test, notez les points à améliorer • Travaillez sur les points d'amélioration jusqu'à ce que votre mBot2 fasse exactement ce que vous aviez en tête • Vous avez réussi ? Filmez le résultat final et demandez à votre professeur si vous pouvez le publier sur les réseaux sociaux avec le hashtag #mBot2inclass |

4. Récapitulatif (5 min)

Étape 4 : Récapitulatif

Comment s'est passée cette mission ? L'objet est-il resté bien en place sur le mBot2 pendant la conduite ?

Dans cette leçon, vous avez appris à connaître les capteurs permettant de mesurer le mouvement autour des axes et l'accélération, le gyroscope et l'accéléromètre. Ils sont intégrés dans un seul petit composant placé sur le CyberPi. Ces capteurs combinés sont utilisés pour déterminer la position d'un robot dans l'espace 3D. Elles sont appelées unités de mesure inertielle, IMU. Vous avez appris où les rencontrer dans la vie quotidienne. Vous savez quels sont les mouvements d'inclinaison et les changements de mouvement que le mBot2 peut mesurer et comment vous pouvez utiliser différents blocs de code pour tirer parti de ces mesures dans vos programmes avec le mBot2.

Il est maintenant temps de procéder à une courte réflexion. Réfléchissez par vous-même puis discutez-en avec le groupe :

- Qu'est-ce qui s'est bien passé, à votre avis ?
- Qu'est-ce qui pourrait être amélioré ?
- Quelles sont les parties de la leçon que vous avez trouvées faciles et celles que vous avez trouvées plus difficiles ?
- Sur quoi souhaitez-vous plus d'explications ?
- Qui pourrait vous aider ?

Leçon 7 :

Un jeu en réseau

Sujet : STEAM

Durée : 45 minutes

Classe(s) : CM2 et plus

Difficulté : Débutant

★ Objectifs de la leçon

À la fin de cette leçon, les élèves seront capables de :

- Faire communiquer plusieurs mBot2 entre eux sans fil, sans avoir besoin d'un point d'accès WIFI
- Écrire leur propre programme informatique en mBlock pour contrôler plusieurs mBot2
- Appliquer une sélection aléatoire dans un programme informatique

★ Aperçu

LAN est l'abréviation de Local Area Network (réseau local). Le mBot2, avec son module WIFI intégré, établit automatiquement une connexion réseau dès qu'il y a plusieurs mBot2 ou CyberPi dans la même pièce dans un rayon de 30 mètres. Grâce à cette connexion réseau, le mBot2 et le CyberPi peuvent communiquer entre eux et exécuter des commandes.

Focus

À la fin de cette leçon, les élèves sauront :

- Ce qu'est une connexion LAN
- Comment mettre en place cette connexion et où l'appliquer ?
- Comment la sélection aléatoire fonctionne en programmation

Liste de contrôle avant la leçon

De quoi avez-vous besoin ?

- Un PC ou ordinateur portable (avec sortie USB) avec le logiciel mBlock installé, la version web (également pour Chromebook), ou une tablette avec l'application mBlock installée
- Le mBot2 avec un CyberPi (plusieurs mBot2 et/ou CyberPi si possible, pour l'étape d'essai)
- Un câble USB-C ou un dongle Bluetooth Makeblock
- Des cartes de couleur (le bleu est recommandé)

Plan de cours

Cette leçon se compose de quatre étapes et dure au total 45 minutes.

| Durée | Contenu |
|------------|--|
| 5 minutes | 1. Introduction <ul style="list-style-type: none"> • Une connexion LAN dans la vie de tous les jours • Comment configurer une connexion LAN avec mBot2 et CyberPi |
| 10 minutes | 2. Travaux pratiques <ul style="list-style-type: none"> • Se familiariser avec les différents blocs de code pour configurer une connexion LAN • Recréer et tester quelques exemples de programmation d'une connexion LAN |
| 25 minutes | Section 3 – Test <ul style="list-style-type: none"> • Réaliser votre propre jeu |
| 5 minutes | 4. Récapitulatif <ul style="list-style-type: none"> • Showtime : montrez ce que vous avez fait avec votre robot dans un film court et amusant pour en discuter plus tard • Si votre enseignant le permet, partagez le résultat final sur les médias sociaux avec le hashtag #mBot2inclass • Réflexion: De quoi êtes-vous le plus fier ? Qu'est-ce que vous aimeriez améliorer dans votre robot ? |

1. Introduction (5 min)

Étape 1 : Introduction

Cette étape se compose de deux parties :

1. Une connexion LAN dans la vie de tous les jours
2. Comment configurer une connexion LAN avec mBot2 et CyberPi

1. Une connexion LAN dans la vie de tous les jours

LAN est l'abréviation de Local Area Network (réseau local) et désigne un réseau d'ordinateurs et de périphériques ayant une certaine étendue spatiale, généralement au sein d'un bâtiment. Le réseau permet aux ordinateurs d'échanger des données entre eux à l'intérieur de ce réseau, et (ce n'est pas obligatoire) avec d'autres réseaux comme Internet. Ils ont donc besoin d'une certaine forme de connexion (câblée ou sans fil/WIFI par ondes radio) et d'un « langage » commun (appelé protocole). Normalement, les ordinateurs individuels se connectent à un dispositif de gestion qui commute et achemine le trafic de données (« routeur »). Sur un réseau WIFI, il s'agit d'un point d'accès dédié ou le routeur Internet offre un point d'accès n'intégré. Pouvez-vous penser à des exemples où un réseau local est utilisé ? Dans cette leçon, le mBot2 et le CyberPi sont les différents appareils qui communiquent entre eux via une connexion LAN.

2. Comment configurer une connexion LAN avec mBot2 et CyberPi

Pour établir une connexion LAN entre mBot2 et CyberPi, vous n'avez pas besoin de grand chose. Le mBot2 et le CyberPi (ou les mBot2 entre eux) seront automatiquement capables de communiquer entre eux sans fil, il n'est pas nécessaire de se connecter à un point d'accès Wi-Fi ou à un routeur pour cela.

Un routeur est-il utilisé ? Alors, tous les mBot2 et CyberPi doivent être mis sur le même canal du routeur. Cela doit être défini dans les paramètres du routeur.

2. Travaux pratiques (10 min)

Étape 2 : Travaux pratiques

Cette étape se compose de deux parties :

1. Apprendre à connaître les différents blocs de code pour la communication LAN
2. Recréer et tester quelques exemples d'une communication LAN

1. Apprendre à connaître les différents blocs de code pour la communication LAN

Dans mBlock, il y a plusieurs blocs de code que vous pouvez utiliser pour programmer la connexion LAN. Vous trouverez ces blocs de code dans la catégorie 'LAN' de la zone des blocs dans mBlock. Ces blocs de codage sont verts.



Le tableau ci-dessous présente un exemple de ces blocs de code pour configurer une connexion LAN.

Bloc de code :

 broadcast message message on LAN

Ce bloc de code vous permet d'envoyer un message à tous les autres mBot2 et CyberPi sur le même réseau. Vous pouvez donner un nom au message (comme un sujet).

Bloc de code :

 when receiving message broadcast on LAN

Au lieu de vérifier constamment la réception d'une émission, il y a un bloc d'événements. Le code attaché à ce bloc n'est exécuté qu'une seule fois, si un message avec le sujet spécifié est reçu.

Dans l'exemple de programmation ci-dessous, il y a deux mBot2 et/ou CyberPi qui communiquent entre eux via une connexion LAN en utilisant ces deux blocs de code. Le premier programme est « l'unité de diffusion », destinée au premier mBot2 ou CyberPi. Le deuxième programme est le code du récepteur pour le deuxième mBot2 ou CyberPi. Vous pouvez télécharger les deux parties du code sur les deux CyberPi/mBot2, puisque le code sous-jacent s'assurera que l'unité d'envoi ne reçoit pas son propre message.

 when button A pressed

 broadcast message welcome on LAN



Après avoir téléchargé les codes de programmation, vous devez redémarrer le mBot2 et/ou le CyberPi.

Maintenant, lorsque vous appuyez sur le bouton A du premier mBot2 ou CyberPi, le second mBot2 ou CyberPi recevra le message envoyé (par exemple, bienvenue). Que se passe-t-il si vous appuyez sur B sur le deuxième CyberPi ou mBot2 ? Que devriez-vous changer pour renvoyer des messages différents ?

Bloc de code :



Ce bloc de code vous permet d'envoyer un message (avec un sujet donné comme précédemment) et une valeur spécifique (numéro ou texte) à tous les autres mBot2 et CyberPi sur le réseau.

Bloc de code :

Pour le code de réception, vous pouvez utiliser le même bloc d'événements et spécifier le sujet auquel il doit réagir, exactement comme dans l'exemple précédent. Pour faire usage de la valeur (texte ou nombre), il existe un bloc rapporteur dédié :



Dans l'exemple de programmation ci-dessous, il y a deux mBot2 et/ou CyberPi qui communiquent entre eux via une connexion LAN en utilisant ces deux blocs de code. Le premier programme est « l'unité de diffusion », destinée au premier mBot2 ou CyberPi. Le deuxième programme est le code du récepteur pour le deuxième mBot2 ou CyberPi. Vous pouvez télécharger les deux parties du code sur les deux CyberPi/mBot2, puisque le code sous-jacent s'assurera que l'unité d'envoi ne reçoit pas son propre message.



Après avoir téléchargé les codes de programmation, vous devez redémarrer le mBot2 et/ou le CyberPi. Si vous appuyez maintenant sur le joystick du premier mBot2 ou CyberPi, le second mBot2 ou CyberPi recevra le message envoyé (valeur 1).

2. Recréer et tester quelques exemples d'une communication LAN

Dans le tableau ci-dessus, chaque bloc de code de la connexion LAN est accompagné d'un exemple de programmation. Vous allez recréer ces exemples de programmation dans mBlock et les tester. Pour un exemple de programmation, proposez votre extension.

3. Test (25 min)

Étape 3 : Test

Vous avez déjà appris beaucoup de choses sur la connexion LAN du mBot2. Vous allez maintenant créer votre propre programme informatique dans mBlock en utilisant la communication LAN.

Dans cette mission, vous allez programmer un jeu par vous-même. Il s'appelle « A la recherche...de la couleur bleue ». Dans ce jeu, un groupe de mBot2 cherche la couleur bleue sur le sol. Si l'un des mBot2 trouve la couleur, il en informe les autres par un message LAN et gagne la partie.

Cela semble assez compliqué ! Heureusement, vous ne devez pas tout programmer vous-même. Vous pouvez voir un exemple de programmation ci-dessous. Le même programme doit être téléchargé sur tous les mBot2 afin qu'ils suivent tous les mêmes instructions.

```

when button A pressed
  forever
    if (quad rgb sensor 1 probe (2) R1 detected blue ? or quad rgb sensor 1 probe (3) L1 detected blue ?) then
      broadcast message blue on LAN
      stop encoder motor all
      LED all displays
      play yeah until done
      stop all
    else
      if (ultrasonic 2 distance to an object (cm) < 15) then
        moves backward at 60 RPM for 2 secs
      else
        random_moves
  
```

```

when receiving blue broadcast on LAN
  stop encoder motor all
  LED all displays
  play sad until done
  stop all

```

Pour que le mBot2 conduise de manière aléatoire, vous pouvez utiliser le bloc de code suivant :

```
pick random 1 to 10
```

Ce bloc de code est utilisé pour sélectionner de manière aléatoire une valeur dans une certaine fourchette. Une instruction particulière peut être affectée à chaque valeur de la plage. De cette façon, le mBot2 peut choisir de manière non linéaire parmi un certain nombre d'instructions. Pour rendre votre programme plus facile à comprendre, vous pouvez séparer les blocs de cette instruction et les placer sous un bloc dédié, que vous pourrez ensuite utiliser dans votre programme principal :

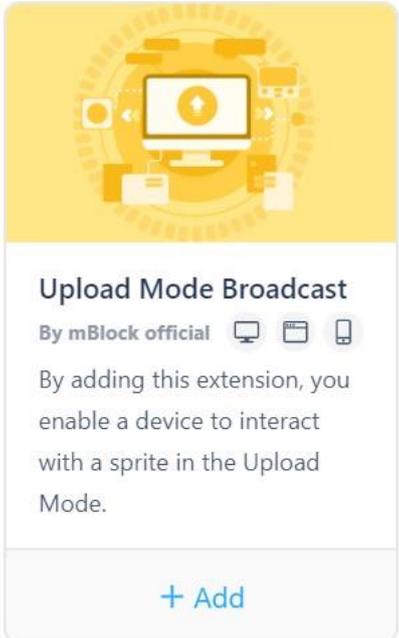
```

define random_moves
  set turn to pick random 1 to 2
  if turn = 1 then
    moves forward at 60 RPM for 1 secs
    turns left 120 ° until done
  if turn = 2 then
    moves forward at 60 RPM for 1 secs
    turns right 120 ° until done

```

Vous n'avez qu'un seul mBot2 ? Vous pouvez alors simuler la communication entre deux appareils en incluant un sprite dans le plateau mBlock. Le plateau est la zone située dans le coin supérieur gauche de l'écran du mBlock.

Vous devez ajouter une nouvelle extension aux blocs de code pour le sprite et le mBot2, appelée « Diffusion en mode téléchargement ». Il vous permet d'envoyer et de recevoir des messages entre le sprite et le robot lorsque ce dernier est en mode de chargement, de la même manière que les messages envoyés via une communication LAN.

| | Extension | Ajouter |
|----------------|---|---|
| Pour le sprite |  | <ul style="list-style-type: none"> • Cliquez sur « Sprites » • Cliquez sur « Extensions » • Cliquez sur « Extensions Sprite » • Trouvez l'extension intitulée « Upload Mode Broadcast » et ajoutez-la |
| Pour mBot2 | | <ul style="list-style-type: none"> • Cliquez sur « Appareils » • Cliquez sur « Extensions » • Cliquez sur « Extensions Appareil » • Trouvez l'extension intitulée « Diffusion en mode téléchargement » et ajoutez-la (elle ne se trouve peut-être pas sur la première page de la bibliothèque des extensions) |

Dans cet exemple de programmation, lorsque mBot2 a trouvé la couleur, il enverra un message au sprite panda dans la scène de mBlock, et le panda dira que mBot2 a trouvé la couleur (il faudra se mettre en mode "Téléverser").

```

when button A pressed
  forever
    if quad rgb sensor 1 probe (2) R1 detected blue ? or quad rgb sensor 1 probe (3) L1 detected blue ? then
      send upload mode message blue
      stop encoder motor all
      LED all displays for 3 secs
      stop all
    else
      if ultrasonic 2 1 distance to an object (cm) < 15 then
        moves backward at 60 RPM for 2 secs
      else
        random_moves
  
```

```

when receiving upload mode message blue
  say mBot2 found the color!
  play sound yes until done
  
```



Vous pouvez copier les exemples de programmation ci-dessus, mais vous pouvez également créer votre propre extension. Il est alors utile d'utiliser le plan par étapes suivant. Avez-vous une idée de ce que vous voulez faire ? Si c'est le cas, discutez-en d'abord avec votre professeur pour savoir si cela est possible.

| | Explication |
|---|--|
| Étape 1 : Que voulez-vous faire ? | <ul style="list-style-type: none">• Que voulez-vous programmer ? |
| Étape 2 : De quoi avez-vous besoin ? | <ul style="list-style-type: none">• De quel matériel avez-vous besoin pour ce faire ? |
| Étape 3 : De quels blocs de code avez-vous besoin pour ce faire ? | <ul style="list-style-type: none">• Comment allez-vous faire pour que le mBot2 se déplace et communique ?• Quels blocs de code utiliserez-vous ?• Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML) • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock. |
| Étape 4 : Test et mise en œuvre | <ul style="list-style-type: none">• Votre première version est prête ? Testez-la ! Pendant le test, notez les points à améliorer• Travaillez sur les points d'amélioration jusqu'à ce que le mBot2 fasse exactement ce que vous aviez en tête• Vous avez réussi ? Filmez le résultat final et demandez à votre professeur si vous pouvez le publier sur les réseaux sociaux avec le hashtag #mBot2inclass |

4. Récapitulatif (5 min)

Étape 4 : Récapitulatif

Avez-vous réussi à programmer le jeu ?

Dans cette leçon, vous avez découvert la connexion LAN et les situations où vous pouvez les rencontrer dans la vie quotidienne. Vous savez comment configurer une connexion LAN entre différents appareils. Vous avez également appris à créer votre propre réseau local à l'aide du mBot2 et du CyberPi. En outre, vous savez comment ajouter une sélection aléatoire à un programme informatique.

Il est maintenant temps de procéder à une courte réflexion. Réfléchissez par vous-même puis discutez-en avec le groupe :

- Qu'est-ce qui s'est bien passé, à votre avis ?
- Qu'est-ce qui pourrait être amélioré ?
- Quelles sont les parties de la leçon que vous avez trouvées faciles et celles que vous avez trouvées plus difficiles ?
- Sur quoi souhaitez-vous plus d'explications ?
- Qui pourrait vous aider ?

Leçon 8 :

mBot2 à votre service

Sujet : STEAM
Durée : 45 minutes

Classe(s) : CM2 et plus
Difficulté : Intermédiaire

★ Objectifs de la leçon

À la fin de cette leçon, les élèves seront capables de :

- Faire exécuter des tâches par le mBot2 en utilisant la reconnaissance vocale et la connexion WIFI
- Combiner la reconnaissance vocale avec des variables pour les dialogues
- Envoyer des messages via la messagerie sur le cloud
- Utiliser les instructions booléennes avec les données du capteur

★ Aperçu

La reconnaissance vocale est la technologie qui permet de convertir les mots prononcés (mais aussi des phrases entières) en texte. La reconnaissance d'un seul mot par la seule variation de sa fréquence est une tâche assez complexe. Et pour être utile, la reconnaissance vocale devrait fonctionner indépendamment de la voix individuelle : avoir une voix aiguë ou grave, parler lentement ou rapidement ne devrait faire aucune différence. La reconnaissance vocale fait partie de l'interaction avec l'utilisateur dans l'électronique grand public (comme les smartphones, les téléviseurs, la maison intelligente), mais aussi dans les systèmes d'assistance. Dans cette activité, vous allez programmer un court dialogue avec le mBot2 agissant comme un robot de service. Vous pouvez étendre cette tâche ultérieurement pour simuler un dialogue utilisateur pour d'autres tâches également.

Focus

À la fin de cette leçon, les élèves sauront :

- Ce qu'est la reconnaissance vocale et comment l'appliquer
- Comment établir une connexion Internet en utilisant le WIFI ?
- Comment utiliser la reconnaissance vocale et développer un dialogue avec l'utilisateur ?
- Envoyer des messages à des ordinateurs distants via la messagerie sur le cloud
- Utiliser des variables pour décrire une étape d'un processus (comme une commande, une interaction avec l'utilisateur).

Liste de contrôle avant la leçon

De quoi avez-vous besoin ?

- PC ou ordinateur portable (avec sortie USB) avec le logiciel mBlock installé, la version web (également pour ChromeBook), ou une tablette avec l'application mBlock installée
- mBot2 avec un CyberPi
- Câble USB-C ou un dongle Bluetooth Makeblock

Plan de cours

Cette leçon se compose de quatre étapes et dure au total 45 minutes.

| Durée | Contenu |
|------------|---|
| 5 minutes | 1. Introduction <ul style="list-style-type: none"> • La reconnaissance vocale dans la vie quotidienne • Découvrir la reconnaissance vocale avec mBot2 |
| 10 minutes | 2. Travaux pratiques <ul style="list-style-type: none"> • Configurer une connexion WIFI • Programmation et application de la reconnaissance vocale • Messagerie sur le cloud |
| 25 minutes | 3. Test <ul style="list-style-type: none"> • Programmer mBot2 pour qu'il devienne un robot serveur |
| 5 minutes | 4. Récapitulatif <ul style="list-style-type: none"> • Showtime : montrez ce que vous avez fait avec votre robot dans un film court et amusant pour en discuter plus tard • Si votre enseignant le permet, partagez le résultat final sur les médias sociaux avec le hashtag #mBot2inclass • Réflexion De quoi êtes-vous le plus fier ? Qu'est-ce que vous aimeriez améliorer dans votre robot ? |

1. Introduction (5 min)

Étape 1 : Introduction

Cette étape se compose de deux parties :

1. La reconnaissance vocale dans la vie quotidienne
2. Découvrir la reconnaissance vocale avec mBot2

1. La reconnaissance vocale dans la vie quotidienne

La reconnaissance vocale est la technologie qui permet d'identifier des mots et des phrases dans le son de la voix humaine et de les traduire en texte. Il s'agit d'une tâche de calcul, puisque tous les algorithmes peuvent examiner un changement de fréquence dans le temps. Mais même cela diffère d'une personne à l'autre, il suffit de penser à la voix basse ou haute, au fait de parler vite ou en dialecte. Même l'humeur peut changer le « ton » d'une phrase. Les recherches menées depuis des décennies en informatique et en linguistique permettent d'intégrer de plus en plus ces tâches complexes dans la vie quotidienne.

Certains systèmes de reconnaissance nécessitent une forme « d'entraînement », où l'utilisateur lit des textes spécifiques connus de l'algorithme de détection, adaptant un modèle informatique au ton de voix individuel. D'autres n'identifient que des mots très spécifiques, ils sont surtout utilisés dans les systèmes de dialogue avec les utilisateurs. Certains systèmes peuvent reconnaître et traduire des textes indépendamment du contexte.

Le CyberPi et le mBot2 peuvent utiliser la reconnaissance vocale sans avoir besoin d'un codage complexe. Cependant, comme la sortie n'est « que » du texte, le programme doit l'analyser et le traiter.

La reconnaissance vocale est utilisée dans l'électronique grand public comme les téléviseurs, les smartphones ou les maisons intelligentes pour apporter un confort supplémentaire à l'utilisateur - demander une connexion, inscrire un rendez-vous dans le calendrier, ou faire modifier la température et la lumière de la pièce simplement en le disant.

Outre le confort, la reconnaissance vocale peut être d'une grande aide pour les personnes handicapées ou apporter une sécurité supplémentaire. Par exemple, un conducteur de voiture peut approuver un changement de navigation en raison d'embouteillages à venir juste par la voix, au lieu d'interagir avec des boutons ou une interface à l'écran.

Dans cette activité, vous allez utiliser certaines des fonctionnalités de la reconnaissance vocale.

2. Découvrir la reconnaissance vocale avec mBot2

Le mBot2 offre une reconnaissance vocale basée sur le codage par blocs - il suffit de lancer la reconnaissance pendant un certain temps (comme 2, 3 ou 5 secondes) en utilisant un seul bloc et de récupérer le texte reconnu sous forme de chaîne par un autre bloc. Les blocs de codage sont expliqués ci-dessous.

Cette reconnaissance vocale peut comprendre de nombreuses langues différentes et n'est pas limitée à un petit ensemble de mots prédéfinis. Par conséquent, il s'agit d'une tâche de calcul,

elle est si exigeante que le mBot2 ou le CyberPi ne peuvent pas la gérer eux-mêmes... ils doivent faire appel à une assistance via Internet. C'est pourquoi il faut tout d'abord établir une connexion à Internet, et un compte d'utilisateur est également nécessaire. La création d'un compte utilisateur génère une « clé » permettant de se connecter aux services utilisés (pour l'utilisation des données et la confidentialité, voir ci-dessous)

La reconnaissance vocale enregistre l'audio comme vous l'avez fait dans la 3^e activité, puis envoie cet enregistrement à un réseau très puissant d'ordinateurs pour traitement. Ils se chargent de la traduction en texte et renvoient le texte au CyberPi/mBot2. Le réseau d'ordinateurs est généralement appelé « le cloud ». Il ne s'agit plus d'un ordinateur spécifique, mais plutôt d'une énorme grappe d'ordinateurs qui se répartissent les tâches de calcul. Vous pouvez utiliser Internet pour transmettre des données entre plusieurs CyberPi dans des réseaux et des lieux différents (comme à l'école ou à la maison). Ce service numérique est également traité dans le cloud.

Note sur l'utilisation des données et la confidentialité :

La reconnaissance vocale nécessite une connexion internet pour transférer l'audio enregistré et renvoyer le texte transcrit. L'enregistrement audio est traité sur des serveurs en Europe, dans le cloud Microsoft Azure, mais il ne sera pas stocké de manière permanente. Pour utiliser les services, une « clé » est nécessaire, qui sera automatiquement générée par la création d'un compte utilisateur sur Makeblock.com et utilisée en interne par les blocs de codage. Il n'est pas nécessaire d'entrer manuellement cette clé où que ce soit.

En dehors d'une adresse électronique et d'un mot de passe choisi par vous-même, aucune autre information personnelle (comme les noms, les photos, le sexe...) n'est obligatoire.

La connexion Internet nécessite un accès WIFI via SSID (c'est-à-dire le nom du réseau sans fil) et un mot de passe. Les connexions basées sur des certificats ne sont pas prises en charge. Le SSID et le mot de passe sont stockés avec le programme en texte clair. Il peut donc être judicieux d'utiliser un point d'accès temporaire. L'idéal est d'utiliser un smartphone avec un forfait de données.

2. Travaux pratiques (10 min)

Étape 2 : Travaux pratiques :

Dans cette étape, vous vous familiariserez avec les différents blocs de code des fonctions IA et IoT. Cette étape se compose de trois parties :

4. Configurer une connexion WIFI
5. Programmation et application de la reconnaissance vocale
6. Messagerie et communication en réseau local et en cloud



1. Configurer une connexion WIFI

Ouvrez mBlock et connectez le CyberPi dans l'onglet « Périphériques ». Passez en mode de téléchargement. Une fois connecté, recherchez la catégorie de bloc AI (Artificial Intelligence) dans les blocs de code et cliquez dessus. Vous verrez maintenant les fonctions disponibles, y compris la reconnaissance vocale (si les blocs sont grisés, vous avez oublié de passer en mode téléchargement...).



Bloc de code :



Ce bloc lance les tentatives de connexion au réseau sans fil fourni. Vous avez besoin d'un routeur ou d'un point d'accès qui permette aux appareils de se connecter au réseau et d'accéder à Internet. Certaines écoles n'autorisent que les appareils connus - chaque adaptateur réseau (avec ou sans fil) possède un numéro unique, l'adresse MAC. Si ce numéro est inconnu de ces réseaux sécurisés, la connexion sera refusée même si le mot de passe est correct.

Le moyen le plus simple de contourner ce problème est de créer un point d'accès temporaire, par exemple à l'aide d'un smartphone avec un forfait de données qui n'est utilisé que pour le CyberPi/mBot2 pendant un cours (vérifiez les réglementations légales de votre école).

Tous les blocs qui suivent ce bloc de code seront exécutés immédiatement, de sorte qu'il n'attend pas qu'une connexion soit établie (« non bloquant »). Vous devez coder vous-même une routine d'attente avec le bloc suivant :

Bloc de code :

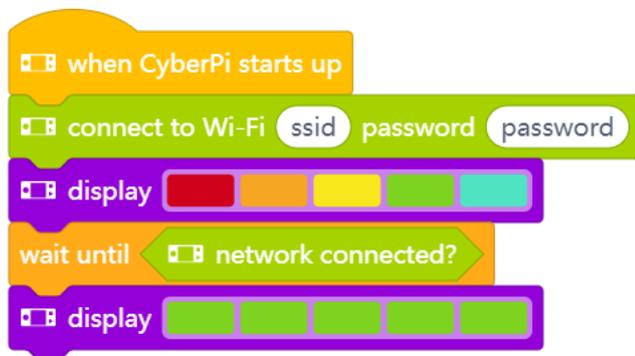


Ce bloc booléen vérifie si la connexion au réseau aurait pu être établie, si c'est le cas, il envoie une valeur "vrai". Vous pouvez l'utiliser avec un bloc d'attente :



Cette combinaison arrête l'exécution de tout autre code dans le même processus jusqu'à ce qu'une connexion soit établie avec succès. Vous souhaitez peut-être rendre cette fonction plus conviviale à un stade ultérieur, en informant sur les tentatives de connexion, en indiquant une connexion réussie, ou même en ajoutant un délai d'attente (comme un redémarrage si aucune connexion ne peut être établie pendant 30 secondes).

Pour utiliser la reconnaissance vocale, vous devez connecter le CyberPi à Internet. Vous trouverez ci-dessous l'exemple de programmation que vous utiliserez pour que le CyberPi se connecte à Internet. Dès que le CyberPi est connecté à Internet, vous le verrez sur l'écran et les LED deviendront vertes.



Ces exemples utilisent une approche basée sur les événements, de sorte qu'un processus du programme tente d'activer la connexion Internet au démarrage. Les autres processus commencent la reconnaissance. Mais pour l'instant, le programme ne vérifie pas si la connexion Internet peut être établie avant de continuer, et il ne donne aucune indication à l'utilisateur sur ce qui se passe, sauf s'il a réussi. Vous pourriez vouloir changer ceci à un stade ultérieur en essayant vos propres idées..

Veillez à saisir correctement le nom (SSID) et le mot de passe du réseau WIFI (l'image montre un exemple !). Ces deux éléments sont sensibles à la casse. Connectez le mBot2 à votre ordinateur avec un câble USB et activez le mode téléchargement. Recréez le code de programmation ci-dessus et essayez si le code fonctionne. Une astuce consiste à utiliser un point d'accès depuis votre téléphone portable, car le réseau de l'école ne permet souvent qu'aux appareils connus de se connecter (adresses MAC en liste blanche). Assurez-vous que le plan de données couvre l'utilisation d'Internet ! Connectez maintenant le CyberPi à votre ordinateur.

Vous avez maintenant connecté le mBot2 à l'Internet. Vous pouvez maintenant communiquer avec le mBot2 via le cloud. C'est ce qu'on appelle l'Internet des objets (IoT).

Si vous utilisez un CyberPi séparé pour afficher le statut (comme la commande envoyée à la cuisine ou le processus de livraison), il doit se connecter au même réseau WIFI. Cela permet de s'assurer que les deux appareils (mBot2-Cyberpi et CyberPi autonome) utilisent le même canal dans le réseau WIFI. Le routeur/point d'accès décide des canaux. Cela garantit que tous les appareils sont sur le même canal pour l'échange de données. Vous pouvez également vous référer à la communication LAN dans la leçon 7.

2. Programmation et application de la reconnaissance vocale

Vous allez maintenant programmer et tester la reconnaissance vocale. Vous allez compléter le code de programmation dans lequel vous laissez le mBot2 se connecter au WIFI par un code pour la reconnaissance vocale.

Bloc de code :



Ce bloc de code sera utilisé dans un exemple ci-dessous, il fait le contraire de la reconnaissance vocale : avec ce bloc, vous pouvez faire en sorte que le CyberPi/mBot2 lise à haute voix un texte fourni. Ici, le texte est transféré vers le service en cloud, et le fichier audio synthétisé est renvoyé. Il possède une détection automatique de la langue utilisée, de sorte que la langue parlée devrait être la même que celle utilisée dans le texte.

Bloc de code :



La reconnaissance vocale est disponible en 12 langues différentes. Si votre langue maternelle n'est pas prise en charge, vous pouvez peut-être utiliser l'anglais. Ce bloc de code démarre un enregistrement audio pour l'heure spécifique et l'envoie au service en cloud pour la détection. Pour les résultats, vous avez besoin du bloc de code suivant :

Bloc de code :



Les résultats de la reconnaissance vocale sont accessibles par le biais de ce bloc rapporteur. Vous pouvez sauvegarder le résultat dans une variable, au cas où vous auriez besoin de le réutiliser, ou l'afficher à l'écran ou le traiter plus avant avec une analyse (comme « le texte contient-il le mot 'oui' ? »).

Bloc de code :



Les fonctions d'IA permettent même de traduire des textes dans une autre langue. Ce bloc de code est un bloc rapporteur qui vous permet de traduire le texte fourni dans une langue (prise en charge) de votre choix. Vous pouvez utiliser ce bloc avec les précédents et les connaissances acquises lors des activités précédentes pour construire un traducteur à commande vocale : appuyez sur un bouton et traduisez 5 secondes de discours dans une langue étrangère... Vous pouvez même en essayer plusieurs et tester vos compétences en langues étrangères 😊.

Dans l'exemple de programmation ci-dessous, le mBot2 est programmé pour que lorsque vous appuyez sur le bouton B, le mBot2 reconnaisse la parole pendant 3 secondes. Le résultat est affiché sur l'écran du CyberPi et la reconnaissance est prononcée. Recréez le code de

programmation ci-dessous et testez-le. Vérifiez ce que vous avez dit et ce qui a été « compris » et traduit en retour.

Est-ce que ça marche ? Ensuite, voyez ce que vous pouvez faire d'autre avec la reconnaissance vocale !

```

when CyberPi starts up
  connect to Wi-Fi ssid password password
  display [red][orange][yellow][green][cyan]
  wait until [network connected?]
  display [ ][ ][ ][ ][ ]
  
```

```

when button B pressed
  recognize (4) English 3 secs
  show label 1 [speech recognition result] at center of screen by middle pixel
  speak auto [speech recognition result]
  
```

Vous pouvez également utiliser la reconnaissance vocale pour contrôler mBot2 par la voix. Vous pouvez attribuer certaines instructions à certains résultats de la reconnaissance vocale, mais faites attention à la spécificité de vos résultats. Par exemple, vous pouvez construire l'instruction de manière à ce qu'elle ne reconnaisse le « oui » que lorsque le bouton B est pressé. Mais que se passe-t-il si vous dites « oui sans hésiter » ou « oui bien sûr » ? Puis une correspondance directe avec..

```

if [speech recognition result] = "yes" then
  
```

... ne fonctionnerait pas... le résultat de la reconnaissance serait différent (la chaîne « oui » n'est pas égale à la chaîne « oui sans hésiter »). Afin d'autoriser un ensemble plus large de réponses signifiant toutes « oui », vous pouvez simplement vérifier si la chaîne de réponse contient le mot « oui » :

```

if [speech recognition result] contains yes ? then
  
```



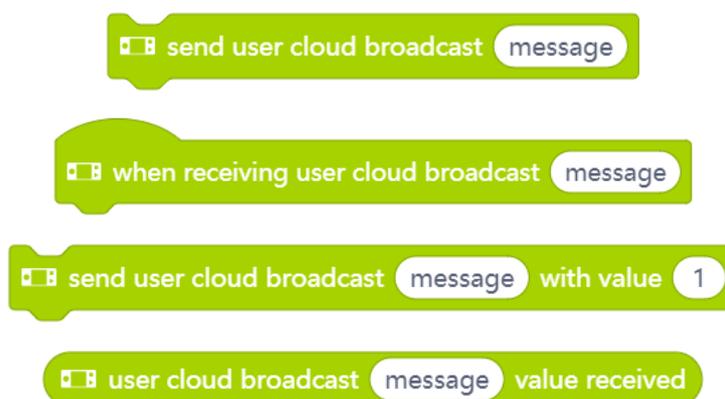
3. Messagerie et communication en réseau local et en cloud

Vous pouvez également utiliser le résultat de la reconnaissance vocale pour interagir avec plusieurs mBot2 ou CyberPi. Dans la leçon 7, vous avez déjà appris à connaître la connexion au réseau local et comment la configurer.

Outre l'envoi d'un message sur le réseau local, qui nécessite que les appareils soient « locaux » les uns par rapport aux autres, vous pouvez utiliser un autre service en nuage pour envoyer des messages à des appareils situés à distance, à condition qu'ils disposent d'une connexion Internet. Ces blocs sont situés dans la catégorie des blocs IoT (Internet of Things).



Ces blocs fonctionnent de manière identique à ceux qui gèrent la communication LAN :



Pour s'assurer que le côté distant ne peut communiquer qu'avec les appareils prévus, vous devez disposer d'un compte utilisateur Makeblock. Cela créera une « clé » pour la communication en cloud - cette clé est cachée dans les blocs de codage et unique à un utilisateur. Pour que deux appareils distants communiquent par Internet, ils doivent être programmés à partir du même compte utilisateur.

Les blocs de diffusion en nuage ont l'avantage de pouvoir communiquer dans le monde entier, mais ils ne sont pas utiles pour une communication rapide - le message peut mettre quelques secondes à arriver. Les diffusions LAN, quant à elles, sont rapides mais ne fonctionnent que localement.

Vous trouverez ci-dessous un exemple de programmation qui résume ce que nous venons de voir. Recréez d'abord le programme et essayez si tout fonctionne avec la reconnaissance et la communication entre le CyberPi et un autre mBot2. Pour cet exercice, vous devez télécharger le même code de programme en mode téléchargement sur les deux appareils.

```

when CyberPi starts up
  connect to Wi-Fi ssid password password
  display [red][orange][yellow][green][cyan]
  wait until [network connected?]
  display [green][green][green][green][green]
  show label 1 Shall the robot drive? at center of screen by middle pixel
  show label 2 press B to answer at middle at bottom by middle pixel
  
```

```

when button B pressed
  recognize (4) English 3 secs
  show label 1 [speech recognition result] at center of screen by middle pixel
  if [speech recognition result contains yes?] then
    broadcast message yes on LAN
  
```

```

when receiving yes broadcast on LAN
  moves forward 10 cm until done
  
```

Après vous être assuré que tout fonctionne bien, vous pouvez modifier et étendre l'exemple - peut-être des commandes directionnelles ? Pensez à la leçon 1 et à l'exercice consistant à faire naviguer manuellement le mBot2 dans un labyrinthe... pouvez-vous faire de même avec des commandes vocales ?

3. Test [25 minutes]

Étape 3 : Test

Vous savez maintenant comment établir une connexion WIFI, programmer la reconnaissance vocale et déployer la messagerie en cloud ou en réseau local. Il est maintenant temps de vous lancer vous-même. Vous allez programmer un robot de service poli : un robot serveur. Il existe

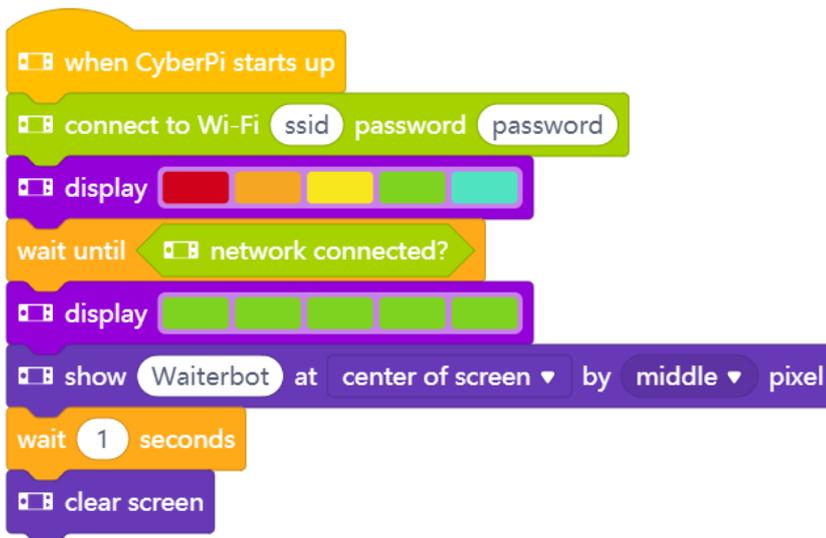
déjà des restaurants dans certains pays où un robot vient prendre la commande et apporte la nourriture depuis les cuisines. Regardez la vidéo suivante :

<https://www.youtube.com/watch?v=FFCPKmLAZb4>

Le robot serveur que vous allez fabriquer doit être capable de faire un certain nombre de choses :

- Rouler en ligne droite
- S'arrêter à une zone rouge (votre table), prendre une commande et l'envoyer aux cuisines
- Option supplémentaire : si un deuxième mBot2 ou CyberPi est disponible, afficher les commandes en cuisine (pour qu'ils puissent déjà commencer à préparer la nourriture)
- Continuer à rouler jusqu'à ce que le mBot2 rencontre une surface verte (les cuisines) et s'arrêter pour prendre la nourriture dans la cuisine
- Facultatif : si un deuxième mBot2 ou CyberPi est disponible, notifier que le plat est en train d'être servi

La séquence de démarrage est presque identique à celle des exemples précédents - n'oubliez pas d'entrer le SSID et le mot de passe corrects :



Le bloc de code suivant est programmé dans le même champ de script que dans l'exemple précédent. Dans ce bloc de code, vous donnez l'ordre au mBot2 de commencer à rouler et de s'arrêter à un repère rouge pour prendre la commande. Après cela, le mBot2 se dirige vers une marque verte. Les marqueurs de couleur indiquent la table et la cuisine. Pour commencer plus facilement, nous supposons qu'ils sont en ligne droite. Si tout fonctionne, vous pouvez aller plus loin et inclure les connaissances acquises lors des leçons précédentes et faire en sorte que le robot suive une ligne jusqu'à la table (la carte fournie comporte des marqueurs rouges et verts sur la piste qui peuvent être utilisés à cette fin). La leçon 5 explique tout sur le suivi de ligne... En complément de la reconnaissance vocale, en restant sur la ligne droite entre la table et la cuisine, vous pouvez utiliser la messagerie cloud ou LAN pour envoyer la commande à un autre mBot2 ou CyberPi pour qu'il l'affiche.

```

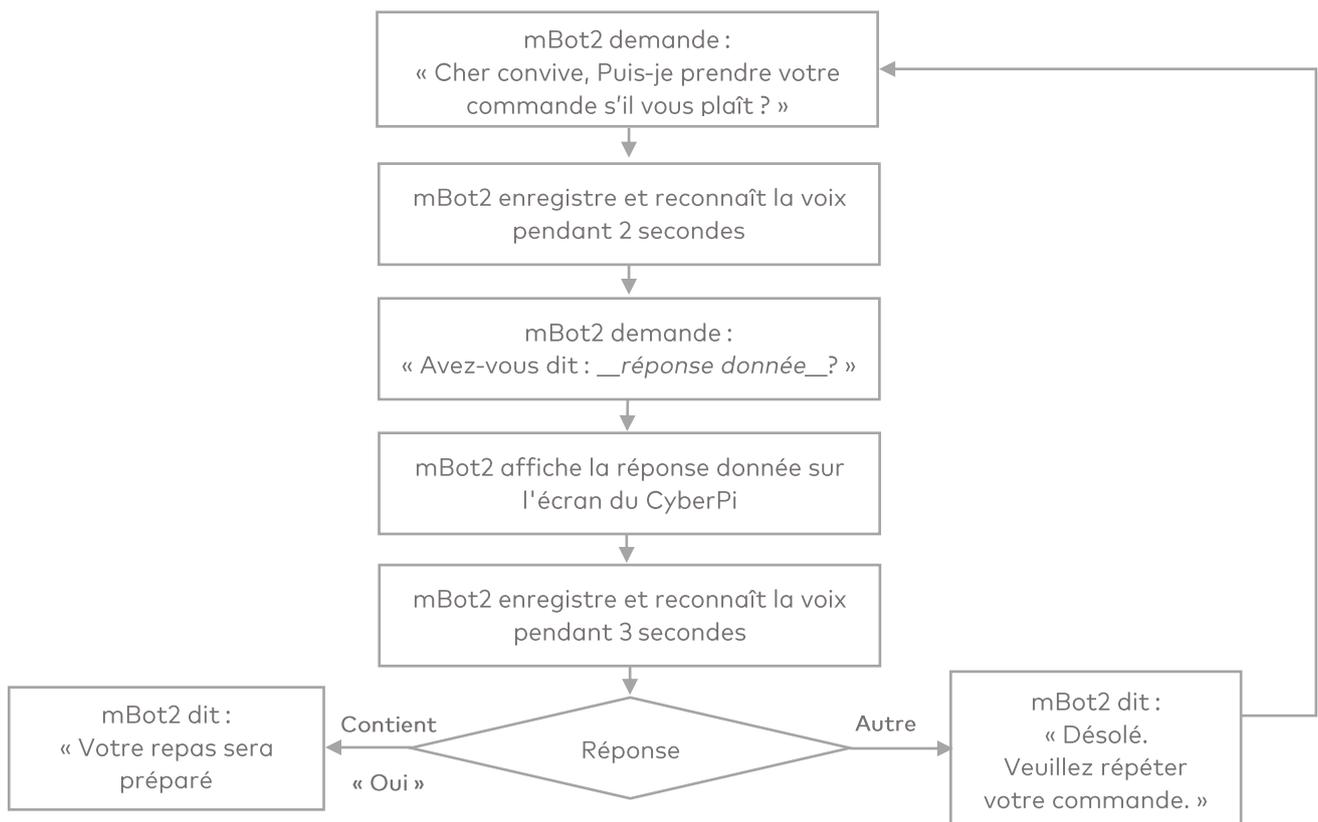
when joystick middle pressed
  clear screen
  moves forward at 50 RPM
  wait until quad rgb sensor 1 probe (3) L1 detected red ? or quad rgb sensor 1 probe (2) R1 detected red ?
  stop encoder motor all
  get_order
  moves forward at 50 RPM
  wait until quad rgb sensor 1 probe (3) L1 detected green ? or quad rgb sensor 1 probe (2) R1 detected green ?
  stop encoder motor all
  broadcast message food in delivery on LAN
  
```

Vous devez créer vous-même le bloc 'get_order' dans 'My blocks'. Cela permet de garder une vue d'ensemble du programme principal - il ne reste plus qu'à conduire jusqu'aux marqueurs de couleur en obtenant l'ordre entre les deux.

Pour définir réellement ce que signifie « get_order », pensons d'abord à un processus de commande :

1. Prendre la commande
2. Reconnaître la réponse
3. Vérifier que la commande comprise est correcte
4. Si oui, passer à la suite (envoyer la commande à la cuisine et terminer la séquence de commande) Si non, demander à nouveau.

Il est sage de procéder à l'étape de vérification pour éviter les mauvaises commandes. Pour traduire ces étapes de manière structurée, vous pouvez utiliser un diagramme de flux comme celui qui suit :



À partir de là, il est beaucoup plus facile de créer les blocs de codage nécessaires : vous commencez par le bloc de code 'obtenir la commande'. Vous créez à nouveau ce bloc de code dans 'My blocks'.

```

define get_order
  set order_status to 0
  LED all displays green
  speak auto Dear Guest! Your orders please?
  repeat until order_status > 0
    LED all displays red
    recognize (4) English 2 secs
    turn off LED all
    set order to speech recognition result
    show order at center of screen by middle pixel
    LED all displays green
    speak auto join Did you say order
    LED all displays red
    recognize (4) English 1 secs
    turn off LED all
    show speech recognition result at center of screen by middle pixel
  if speech recognition result contains yes ? then
    set order_status to 1
    speak auto Your meal will be prepared immediately.
    broadcast message food with value order on LAN
  else
    speak auto Sorry. Please repeat your order
  
```

En outre, vous devez créer deux variables. Dans une variable, nous stockons la réponse donnée pour la commande, de sorte que si tout a été compris correctement, il peut être envoyé à la cuisine (attention : l'étape de vérification par oui/non - la réponse à cette question remplacerait le résultat de la reconnaissance précédente dans le bloc de rapport ; c'est pourquoi nous devons stocker la commande reconnue dans une variable).

La deuxième variable permet de suivre l'ensemble du processus : la commande est-elle complète (c'est-à-dire comprise correctement) ? Dans le cas contraire, le processus de commande recommence. Cette deuxième utilisation de la variable rend compte des données du processus (étape).

Vous pouvez utiliser les connaissances des leçons précédentes pour vous aider dans le programme, par exemple en allumant des LED pour faire comprendre à l'utilisateur que le robot est sur le point de parler (vert) ou d'écouter (rouge), ou vous pouvez utiliser l'écran du serveur-mBot2 pour afficher des données. Utilisez-le à des points de contrôle dans votre programme, afin d'évaluer si votre code fonctionne et de donner un retour aux utilisateurs. En complément, vous pouvez utiliser la messagerie cloud ou LAN pour envoyer des données à un autre mBot2. Pour une meilleure expérience d'apprentissage, il existe des exemples sur la façon de résoudre des tâches spécifiques dans cette leçon. Vous pouvez vous y référer si vous rencontrez des problèmes avec votre propre code ou obtenir des idées sur la façon de traiter une situation spécifique. En général, il est préférable de faire fonctionner une ébauche et d'affiner le code par itérations plutôt que d'essayer de tout faire et de tout programmer parfaitement en une seule fois. Commencez par la séquence de conduite et au lieu d'avoir le bloc 'get_order' prêt, faites un bip à cet endroit pour montrer que tout fonctionne jusqu'à présent. Travaillez ensuite sur le processus de commande, mais ne vous occupez pas de la conduite. De cette façon, vous pouvez itérer plus rapidement sans que le robot ait à passer entre les bornes à chaque fois que vous voulez essayer votre code. Enfin, combinez-les pour résoudre la tâche donnée.

Dans les exemples de programmation ci-dessus, vous avez également envoyé un message à un deuxième mBot2 comme option supplémentaire. Par exemple, dans 'order process', vous avez dit qu'une fois que le client a confirmé la commande, le mBot2 envoie un message via le cloud ou la messagerie LAN à un deuxième mBot2 ou CyberPi (pour afficher les commandes entrantes dans les cuisines). Maintenant vous devez programmer le second mBot2 pour qu'il reçoive le message.



Dans l'exemple de programmation ci-dessous, les cuisines reçoivent la commande et l'affichent. Créez l'exemple de programmation ci-dessous dans le même script. Placez-la à côté du premier exemple de programmation. De cette façon, le scénario reste clair.

Dans l'exemple de programmation « commander un robot », vous avez programmé qu'après avoir détecté la zone verte, un message sur le cloud est envoyé au client pour l'informer de l'arrivée de sa commande. Ceci est également programmé dans le même champ de script à



côté des autres blocs - vous pourriez utiliser un autre CyberPi (comme un dispositif intelligent sur la table du client) pour l'informer de l'état de la commande :

```
when receiving food in delivery broadcast on LAN
  play LED animation rainbow until done
  play yummy until done
  turn off LED all
```

Lorsque vous aurez tout programmé, votre zone de script ressemblera à ceci :

```
when CyberPi starts up
  connect to Wi-Fi ssid password password
  display [red, orange, yellow, green, blue]
  wait until network connected?
  display [green, green, green, green, green]
  show Waiterbot at center of screen by middle pixel
  wait 1 seconds
  clear screen

when joystick middle pressed
  clear screen
  moves forward at 50 RPM
  wait until quad rgb sensor 1 probe (3) L1 detected red ? or quad rgb sensor 1 probe (2) R1 detected red ?
  stop encoder motor all
  get_order
  moves forward at 50 RPM
  wait until quad rgb sensor 1 probe (3) L1 detected green ? or quad rgb sensor 1 probe (2) R1 detected green ?
  stop encoder motor all
  broadcast message food in delivery on LAN

when receiving food broadcast on LAN
  show LAN broadcast food value received at center of screen by middle pixel
  wait 1 seconds
  clear screen

when receiving food in delivery broadcast on LAN
  play LED animation rainbow until done
  play yummy until done
  turn off LED all
```

```

define get_order
  set order_status to 0
  LED all displays green
  speak auto Dear Guest! Your orders please?
  repeat until order_status > 0
    LED all displays red
    recognize (4) English 2 secs
    turn off LED all
    set order to speech recognition result
    show order at center of screen by middle pixel
    LED all displays green
    speak auto join Did you say order
    LED all displays red
    recognize (4) English 1 secs
    turn off LED all
    show speech recognition result at center of screen by middle pixel
    if speech recognition result contains yes ? then
      set order_status to 1
      speak auto Your meal will be prepared immediately.
      broadcast message food with value order on LAN
    else
      speak auto Sorry. Please repeat your order
  
```

Vous devez télécharger ce code sur tous les mBot2s et/ou CyberPi que vous souhaitez utiliser. Il peut s'agir du même code parce qu'ils sont automatiquement connectés via la connexion LAN ou Cloud - et celui qui envoie le message n'y réagit pas.

Créez une ligne droite pour que le mBot2 puisse rouler et assurez-vous qu'il y a un large marqueur rouge et vert. Téléchargez le code et essayez de voir si le robot serveur traite les commandes des clients.

Bien sûr, il est toujours possible de s'améliorer. Ajustez le code là où vous pensez qu'il pourrait être meilleur ou plus joli. Peut-être pourriez-vous commencer avec un suiveur de ligne capable de prendre les commandes ? Ou bien vous approfondissez cet exercice et avez un mBot2 dédié au serveur qui prend les commandes et un mBot2 de livraison qui apporte la nourriture. Vous pouvez coopérer avec d'autres équipes de votre classe et intégrer leurs robots..

Discutez en classe d'extensions possibles comme celles-ci en examinant les progrès réalisés jusqu'à présent et élaborer un cas d'utilisation avec plusieurs robots, le contexte n'est pas obligatoirement un restaurant.

4. Récapitulatif (5 min)

Étape 4 : Récapitulatif

Avez-vous réussi à programmer le robot-serveur ?

Dans cette leçon, vous avez découvert la reconnaissance et la synthèse vocales et comment les appliquer via un service en cloud. Vous avez réussi à connecter le mBot2 à Internet et ainsi envoyer des données localement ou même à distance.

La création d'un dialogue (comme pour commander un repas) faisait partie de l'exercice, et vous avez appris à filtrer la réponse textuelle pour des mots déclencheurs spécifiques afin d'avoir une conversation plus « naturelle » (sans attendre de l'utilisateur qu'il réponde uniquement par des réponses spécifiques).

Il est maintenant temps de procéder à une courte réflexion. Réfléchissez par vous-même puis discutez-en avec le groupe :

- Qu'est-ce qui s'est bien passé, à votre avis ?
- Qu'est-ce qui pourrait être amélioré ?
- Quelles sont les parties de la leçon que vous avez trouvées faciles et celles que vous avez trouvées plus difficiles ?
- Sur quoi souhaitez-vous plus d'explications ?
- Qui pourrait vous aider ?

Leçon 9 : mBot2 dans la nature

Sujet : STEAM
Durée : 45 minutes

Classe(s) : CM2 et plus
Difficulté : débutant

★ Objectifs de la leçon

À la fin de cette leçon, les élèves seront capables de :

- Intégrer l'apprentissage automatique dans la programmation du mBot2
- Faire fonctionner ensemble plusieurs (machine learning) caractéristiques techniques du mBot2
- Établir une communication entre le robot et l'ordinateur, même si le robot fonctionne indépendamment en mode téléchargement pour l'interaction entre le matériel physique (robot) et l'étape sprite du logiciel mBlock

★ Aperçu

L'apprentissage automatique est une forme faible d'intelligence artificielle (IA). L'intelligence artificielle désigne les dispositifs ou les machines qui (machine learning) imitent l'intelligence humaine. Sur la base des données collectées, l'apprentissage automatique permet aux systèmes informatiques ou robotiques de choisir des actions en fonction de l'historique des données et des probabilités, sans qu'il soit nécessaire de formuler explicitement un algorithme à cet effet.

Focus

À la fin de cette leçon, les élèves sauront :

- Ce qu'est l'apprentissage automatique et comment l'appliquer,
- Comment faire fonctionner mBot2 en fonction des conditions,
- Comment les différentes caractéristiques techniques de mBot2 fonctionnent ensemble dans un écosystème,
- Établir un protocole de communication entre le robot exécutant des programmes de manière autonome et l'ordinateur

Liste de contrôle avant la leçon

De quoi avez-vous besoin ?

- PC ou ordinateur portable (avec sortie USB) avec le logiciel mBlock installé, la version web (également pour ChromeBook)
- mBot2 avec un CyberPi
- Câble USB-C ou un dongle Bluetooth Makeblock
- Webcam, intégrée à votre ordinateur portable ou externe
- Imprimante (facultative) ou papier, crayons et marqueurs pour dessiner ou smartphone pour afficher des photos

Plan de cours

Cette leçon se compose de quatre étapes et dure au total 45 minutes.

| Durée | Contenu |
|------------|---|
| 5 minutes | 1. Introduction <ul style="list-style-type: none">• L'apprentissage automatique dans la vie quotidienne• Apprendre à connaître l'apprentissage automatique |
| 25 minutes | 2. Travaux pratiques <ul style="list-style-type: none">• Application de l'apprentissage automatique• L'apprentissage automatique en combinaison avec le mBot2 dans mBlock• Création d'un protocole de communication à partir de programmes en cours d'exécution |
| 10 minutes | 3. Test <ul style="list-style-type: none">• Tâche finale : l'écosystème d'une souris |
| 5 minutes | 4. Récapitulatif <ul style="list-style-type: none">• Showtime : montrez votre écosystème dans un film court et amusant pour en discuter plus tard.• Si votre enseignant le permet, partagez le résultat final sur les médias sociaux avec le hashtag #mBot2inclass• Réflexion: De quoi êtes-vous le plus fier ? Qu'est-ce que vous aimeriez améliorer dans votre robot ? |

1. Introduction (5 min)

Étape 1 : Introduction

Cette étape se compose de trois parties :

1. L'apprentissage automatique dans la vie quotidienne
2. Apprendre à connaître l'extension de l'apprentissage automatique dans mBlock
3. Application de l'apprentissage automatique

1. L'apprentissage automatique dans la vie quotidienne

L'apprentissage automatique fonctionne sur une base statistique : des données d'apprentissage sont fournies à l'algorithme, comme des photos de différents objets accompagnés de leur description. La phase d'apprentissage tente ensuite de distinguer au mieux ces images et de faire correspondre chaque image à la catégorie correcte. Après cette phase, les modèles appris sont testés par rapport à de nouveaux ensembles, et l'apprentissage peut être renforcé. La différence avec les programmes de tri « classiques » est qu'il n'y a pas d'indication claire à l'avance sur la façon de décider entre les différentes catégories, comme on peut différencier les couleurs.

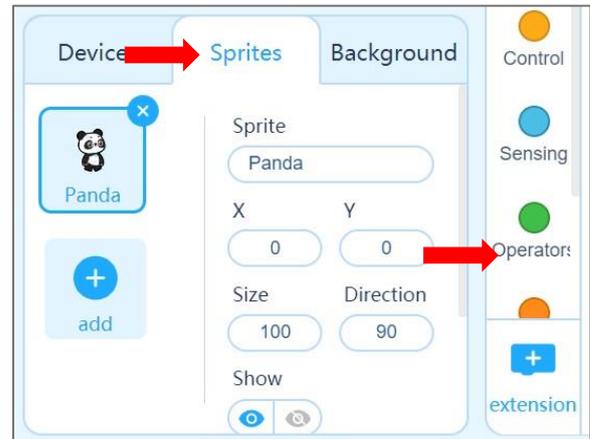
En « apprenant des données passées », le programme informatique peut mieux s'adapter aux situations changeantes. Pensez à la reconnaissance faciale sur votre téléphone : les premières fois qu'il est configuré, il est plus difficile pour le téléphone de vous identifier correctement, mais avec le temps, l'algorithme « apprend » à identifier votre visage dans différents environnements également. L'apprentissage automatique est largement utilisé dans la classification des images (par exemple, sur les médias sociaux) pour analyser automatiquement ce que les photos téléchargées montrent : un chat, un chien, des meubles, des personnes, etc.

L'apprentissage automatique est considéré comme une « intelligence artificielle faible », car le programme informatique n'est pas conscient de lui-même et ne peut pas « penser » comme nous le faisons, mais son comportement actuel est influencé par les données collectées et analysées.

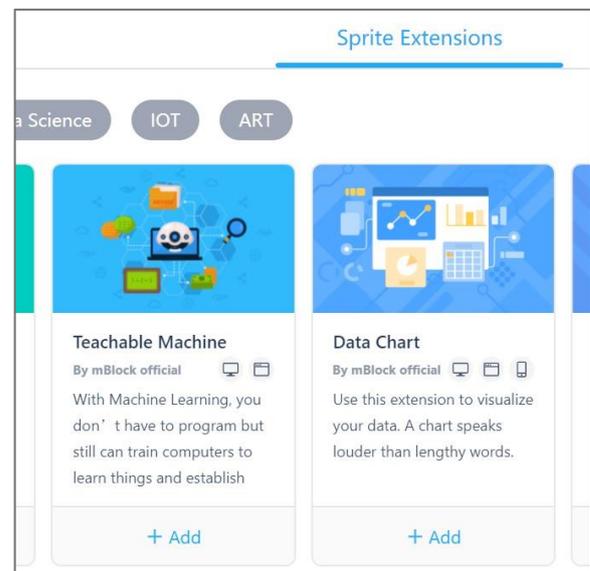
2. Apprendre à connaître l'extension de l'apprentissage automatique dans mBlock

Dans mBlock, il y a un outil d'apprentissage automatique. Vous devez ajouter cet outil à partir de la bibliothèque d'extensions. Procédez comme suit :

- Cliquez sur 'Sprites'.
- Cliquez sur 'Extensions' (en bas au milieu de l'écran).



- Vous allez maintenant entrer dans la bibliothèque d'extension. En haut de l'écran, cliquez sur 'Sprite Extensions'.
- Trouvez l'extension 'Teachable Machine' et ajoutez-la.



- L'extension sera maintenant ajoutée aux blocs de code. Vous pouvez reconnaître l'extension par l'abréviation TM en vert.
- Assurez-vous d'avoir une webcam qui fonctionne.
- Cliquez sur le bouton 'Training model'.



Training model

L'extension consiste en un modèle d'entraînement dans lequel vous pouvez « enseigner » à différencier des séries de photos prises avec une webcam. Le modèle de base que vous voyez

se compose de 3 catégories. Si vous en voulez plus, cliquez sur « construire un nouveau modèle » et choisissez le nombre de catégories.

Quelques conseils :

- Réfléchissez au préalable au nombre de catégories et à la reconnaissance. Vous ne pouvez pas ajouter une catégorie ultérieurement, il vous faudrait tout recommencer.
- Choisissez une catégorie supplémentaire à utiliser comme « arrière-plan ». Ainsi, si vous voulez reconnaître 2 animaux différents grâce aux photos que vous tenez devant la webcam, choisissez 3 catégories : La troisième sera celui où aucune photo d'animal n'est montrée. Sinon, vous « forcez » le modèle à choisir, par exemple, entre un chat et un chien, même si la webcam montre un arrière-plan vide.
- L'arrière-plan doit généralement être simple et statique (pas de salle de classe avec des élèves, webcam face aux élèves, etc.) et présenter un contraste net avec les objets à reconnaître.

Les images seront analysées uniquement sur l'ordinateur et le résultat pourra être stocké. Les photos elles-mêmes ne seront pas conservées ! Ainsi, si vous enregistrez le fichier du programme dans mBlock et le rouvrez plus tard, vous ne verrez pas les images originales sur lesquelles le modèle a été formé. Vous pouvez toujours ajouter de nouvelles images aux catégories existantes, mais pas supprimer les images individuelles (seulement toutes les images et les apprentissages d'une catégorie).

Vous pouvez directement « enseigner » les catégories en utilisant des images prises avec l'appareil photo. Vous utiliserez les 3 catégories standard pour cet exercice :

1. Vous pouvez commencer par le mBot2. Placez-le sur le bureau et alignez la caméra pour obtenir une bonne vue - ou tenez-le dans votre main si vous utilisez un ordinateur portable avec une caméra intégrée. Cliquez ensuite sur le bouton 'Apprendre' . Maintenez le bouton enfoncé jusqu'à ce que la caméra ait enregistré au moins 20 échantillons. Au-dessus de la mosaïque d'échantillons, vous pouvez voir le nombre d'échantillons stockés pour cette catégorie. Donnez un nom à cette catégorie, par exemple 'mBot2'.
2. Choisissez maintenant un autre objet, par exemple un cahier, tenez-le devant la caméra et cliquez sur le bouton 'Apprendre' Maintenez le bouton enfoncé jusqu'à ce que la caméra ait enregistré au moins 20 échantillons. Donnez un nom à cette catégorie, par exemple « cahier ».
3. Enregistrez à présent le « contexte ». Cette option doit être utilisée lorsqu'aucun des autres objets n'est présenté à la caméra. Maintenez le bouton 'Apprendre' enfoncé jusqu'à ce que la caméra ait enregistré au moins 20 échantillons. Donnez un nom à cette catégorie, par exemple « contexte ».

The screenshot displays three distinct machine learning training tasks in a vertical list:

- Task 1:** Labeled 'mBot2' with 36 samples. The progress bar is empty. A 'Learn' button is visible.
- Task 2:** Labeled 'notebook' with 41 samples. The progress bar shows 40.0% completion. A 'Learn' button is visible.
- Task 3:** Labeled 'nothing' with 20 samples. The progress bar is empty. A 'Learn' button is visible.

Vous avez maintenant « entraîné » l'apprentissage automatique à reconnaître différents objets (mBot2, ordinateur portable et arrière-plan) dans l'image de la caméra. Pour chaque catégorie, vous pouvez voir une barre mesurant un pourcentage. Cela représente la confiance que l'algorithme a dans le résultat de la reconnaissance. Vous pouvez augmenter le pourcentage de confiance en augmentant le nombre d'échantillons et en effectuant de légères variations comme le changement d'angle de l'objet. Cela aidera la caméra à reconnaître les objets encore mieux et plus rapidement.

Vous pouvez tester la reconnaissance en direct sans avoir besoin de codage à ce stade. Ajoutez des images jusqu'à ce que la reconnaissance soit rapide et précise (sans passer d'un résultat à l'autre et en affichant un pourcentage élevé >90 %). Êtes-vous satisfait ? Cliquez ensuite sur 'Utiliser le modèle'. Vous allez maintenant revenir à l'écran pour coder au niveau du sprite.

Il y a maintenant trois nouveaux blocs de code. Le premier bloc indique le résultat de la reconnaissance (tel que vous l'avez tapé), le second indique la confiance pour une catégorie donnée et le troisième est le bloc booléen pour les déclarations conditionnelles. Il vous permet de choisir la catégorie par un menu déroulant. Si vous cliquez sur l'un des blocs de code, une nouvelle fenêtre apparaîtra. Il s'agit de la fenêtre de reconnaissance, qui affiche l'image de la caméra et le résultat de la reconnaissance. Votre webcam fonctionne désormais comme un capteur intelligent, identifiant différents objets ou images.



2. Travaux pratiques (25 min)

Étape 2 : Travaux pratiques

Cette étape se compose de trois parties :

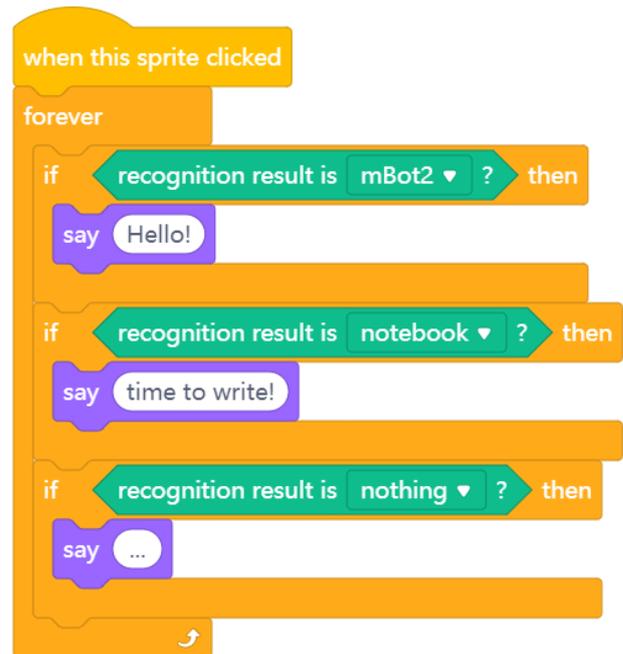
1. Application de l'apprentissage automatique
2. L'apprentissage automatique en combinaison avec le mBot2
3. Établir une connexion LAN

1. Application de l'apprentissage automatique

Vous avez « enseigné » deux objets différents, le mBot2 et un ordinateur portable, ainsi que le contexte du modèle d'apprentissage automatique. Vous pouvez maintenant utiliser la sortie de la reconnaissance comme n'importe quel autre bloc capteur dans votre code de programme. Vous pouvez utiliser des instructions conditionnelles pour différencier les objets et affecter une certaine action séparément à chaque résultat.

Vous allez maintenant essayer de faire parler le sprite (dans ce cas le panda) lorsque la webcam reconnaît un objet appris.

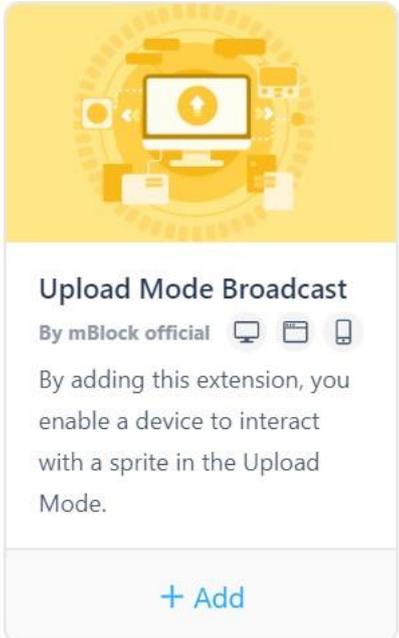
À droite, vous voyez un exemple de programmation. Reproduisez l'exemple de programmation et voyez ce que fait le panda lorsque la webcam reconnaît vos objets appris. Est-ce que cela fonctionne bien ? Ensuite, étudiez les autres possibilités qui s'offrent à vous. Pouvez-vous faire sauter ou marcher le panda ?



```
when this sprite clicked
  forever loop
    if recognition result is mBot2 ? then
      say Hello!
    if recognition result is notebook ? then
      say time to write!
    if recognition result is nothing ? then
      say ...
```

2. L'apprentissage automatique en combinaison avec le mBot2

Vous savez maintenant comment utiliser l'extension d'apprentissage automatique pour la reconnaissance d'images et contrôler un sprite avec elle, par exemple le panda. Vous allez maintenant apprendre à établir une communication entre l'ordinateur et le mBot2 qui exécute son propre programme. Pour cela, le mBot2 doit rester « connecté » à l'ordinateur - soit via un câble USB, soit via le dongle Bluetooth Makeblock. Le dongle permet au robot de se déplacer indépendamment de l'ordinateur ; si vous utilisez un câble USB, assurez-vous de n'effectuer que des mouvements légers du mBot2, comme des petits virages et des distances. Avec cette connexion en place, le sprite (dans ce cas le panda) peut envoyer des données au mBot2. Le mBot2 reçoit ces données et le programme qui tourne sur le robot peut y réagir. Vous devez ajouter une nouvelle extension aux blocs de code pour le sprite et le mBot2, appelée « Diffusion en mode téléchargement ».

| | Extension | Ajouter |
|----------------|---|--|
| Pour le sprite |  | <ul style="list-style-type: none"> • Cliquez sur 'Sprites' • Cliquez sur 'Extensions' • Cliquez sur 'Sprite extensions' • Trouvez l'extension intitulée « Upload Mode Broadcast » et ajoutez-la |
| Pour mBot2 | | <ul style="list-style-type: none"> • Cliquez sur 'Devices' • Cliquez sur 'Extensions' • Cliquez sur 'Device extensions' • Trouvez l'extension intitulée "Upload Mode Broadcast" et ajoutez-la (elle ne se trouve peut-être pas sur la première page de la bibliothèque des extensions) |

De cette façon, l'extension sera ajoutée à la fois pour le mBot2 et le sprite. Dans mBlock, il apparaît en bleu foncé. Cliquez sur la nouvelle extension et voyez les différents blocs de code qui lui sont associés. Fondamentalement, l'extension ajoute un protocole de communication. Il vous permet d'envoyer et de recevoir des messages entre le sprite et le robot lorsque ce dernier est en mode Upload.

3. Établir la connexion avec le mBot2

Vous avez maintenant installé toutes les extensions nécessaires pour que le matériel et le logiciel puissent communiquer entre eux. Vous allez à présent programmer le mBot2 pour qu'il soit contrôlé par l'apprentissage automatique. L'idée est que le mBot2 effectue des actions en fonction des résultats de la reconnaissance. Vous avez programmé une tâche similaire plus tôt juste pour le sprite (mBot2, cahier et contexte).

Maintenant, les résultats de la reconnaissance doivent être envoyés et reçus via la connexion entre la Bot2 et l'ordinateur (soit par câble, soit sans fil via le dongle Bluetooth). Le code de réception utilise des événements, donc le code correspondant est exécuté, une fois le message reçu. Avec les événements, il n'est pas nécessaire de vérifier la présence d'un capteur ou d'un message dans une boucle, ce qui rend le code plus efficace et plus facile à comprendre.

Vous devez créer deux programmes informatiques - un pour le dispositif (mBot2) et un pour le Sprite qui utilise le module complémentaire d'apprentissage automatique :

```

when this sprite clicked
  forever
    if recognition result is mBot2 ? then
      send upload mode message mBot2
    if recognition result is notebook ? then
      send upload mode message notebook
    if recognition result is nothing ? then
      send upload mode message nothing
  
```

```

when receiving upload mode message mBot2
  play yeah until done
  moves forward at 50 RPM for 2 secs

when receiving upload mode message notebook
  play surprised until done
  moves backward at 50 RPM for 2 secs
  
```

Après avoir reprogrammé les blocs de code ci-dessus, assurez-vous que votre ordinateur reste connecté au mBot2 si vous utilisez un câble (ne débranchez pas le câble). L'idéal serait de les connecter via Bluetooth à l'aide de la clé électronique Bluetooth Makeblock, car cela donne au mBot2 plus de flexibilité pour se déplacer.

Si vous utilisez le dongle :



Installez le dongle sur l'ordinateur et allumez le mBot2. Vous devez jumeler le mBot2 avec le dongle Bluetooth. Pour ce faire, cliquez sur le bouton du dongle. La LED se met à clignoter et le dongle essaie de se connecter au contrôleur Makeblock le plus proche (comme mBot2 et autres). Dans une salle de classe, assurez-vous que cela soit fait équipe par équipe, pour assurer la correspondance entre le Dongle et le mBot2. Une fois l'appairage effectué, vous entendrez un bip sonore provenant du mBot2. Cela signifie que l'ordinateur et le mBot2 ont établi une connexion sans fil.

Le dongle Bluetooth Makeblock se souviendra du dernier appareil jumelé et se reconnectera instantanément.

Si vous n'avez pas le dongle Bluetooth de Makeblock, vous pouvez vérifier la compatibilité du Bluetooth dans votre ordinateur via le lien suivant <https://www.yuque.com/makeblock-help-center-en/cyberpi/bluetooth-compatibility>

Passez maintenant mBlock en mode Upload et connectez le mBot2 à mBlock. Cliquez sur le bouton 'Connect' et téléchargez le programme informatique. Vous pouvez maintenant commencer à tester !

3. Test (25 min)

Étape 3 : Test

Vous avez toutes les connaissances et les compétences nécessaires pour utiliser le mBot2 dans n'importe quelle situation. Vous allez maintenant travailler avec l'écosystème d'une petite souris. Une souris vit dans la nature où elle doit chercher de la nourriture et de l'eau. Il existe également de nombreux dangers, comme les prédateurs qui se nourrissent de souris.

Dans ce travail, vous allez programmer le mBot2 pour qu'il agisse comme une petite souris qui cherche de la nourriture et fuit les prédateurs.

Trouvez une image d'un aigle volant et de maïs. Le maïs est la nourriture de la petite souris. L'aigle est le chasseur. Vous pouvez imprimer ces images ou utiliser votre smartphone pour les afficher sur la webcam. Vous pouvez également dessiner à la main des images pour la nourriture et le prédateur si vous le souhaitez. Il suffit d'utiliser la même méthode que celle que vous avez choisie pour les « enseigner » à votre algorithme à l'aide de l'extension d'apprentissage automatique et pour tester le programme.

Programmez le mBot2 de sorte que si vous montrez le maïs à la caméra, le mBot2 se déplace vers l'avant (si vous utilisez le câble, faites un petit mouvement), mais si vous montrez l'image de l'aigle, il recule.

Vous pouvez également ajouter d'autres codes : si la petite souris voit la nourriture, elle dit « miam » et quand elle voit l'aigle, on peut entendre son cœur battre.

Lors de la création du programme informatique pour ce travail, il est utile d'utiliser le plan suivant, étape par étape.

| | Explication |
|---|--|
| Étape 1 : Que voulez-vous faire ? | <ul style="list-style-type: none"> • Que voulez-vous programmer ? |
| Étape 2 : De quoi avez-vous besoin ? | <ul style="list-style-type: none"> • De quoi avez-vous besoin en plus du mBot2 ? |
| Étape 3 : De quels blocs de code avez-vous besoin pour ce faire ? | <ul style="list-style-type: none"> • Comment allez-vous faire pour que le mBot2 se déplace ? • Quels blocs de code utiliserez-vous ? • Faites une brève description du fonctionnement de votre programme (pseudo-code/langage naturel, organigramme ou UML) • Si vous avez besoin de plus d'explications, vous pouvez discuter avec vos camarades, le professeur ou faire une recherche sur le sujet. De l'aide est également disponible pour chaque bloc de code dans mBlock. |



| | |
|---------------------------------|---|
| Étape 4 : Test et mise en œuvre | <ul style="list-style-type: none">• Votre première version est prête ? Testez-la ! Pendant le test, notez les points à améliorer• Travaillez sur les points d'amélioration jusqu'à ce que le mBot2 fasse exactement ce que vous aviez en tête• Vous avez réussi ? Filmez le résultat final et demandez à votre professeur si vous pouvez le publier sur les réseaux sociaux avec le hashtag #mBot2inclass |
|---------------------------------|---|

4. Récapitulatif (5 min)

Étape 4 : Récapitulatif

Avez-vous pu recréer l'écosystème de la souris ?

Dans cette leçon, vous avez appris ce qu'est l'apprentissage automatique et comment l'appliquer. Vous savez également utiliser une communication (et pourquoi de préférence sans fil) entre un sprite et le mBot2, et l'utiliser pour échanger des données entre les deux. En outre, vous avez fait en sorte que diverses caractéristiques techniques du mBot2 fonctionnent ensemble.

Il est maintenant temps de procéder à une courte réflexion. Réfléchissez par vous-même puis discutez-en avec le groupe :

- Qu'est-ce qui s'est bien passé, à votre avis ?
- Qu'est-ce qui pourrait être amélioré ?
- Quelles sont les parties de la leçon que vous avez trouvées faciles et celles que vous avez trouvées plus difficiles ?
- Sur quoi souhaitez-vous plus d'explications ?
- Qui pourrait vous aider ?