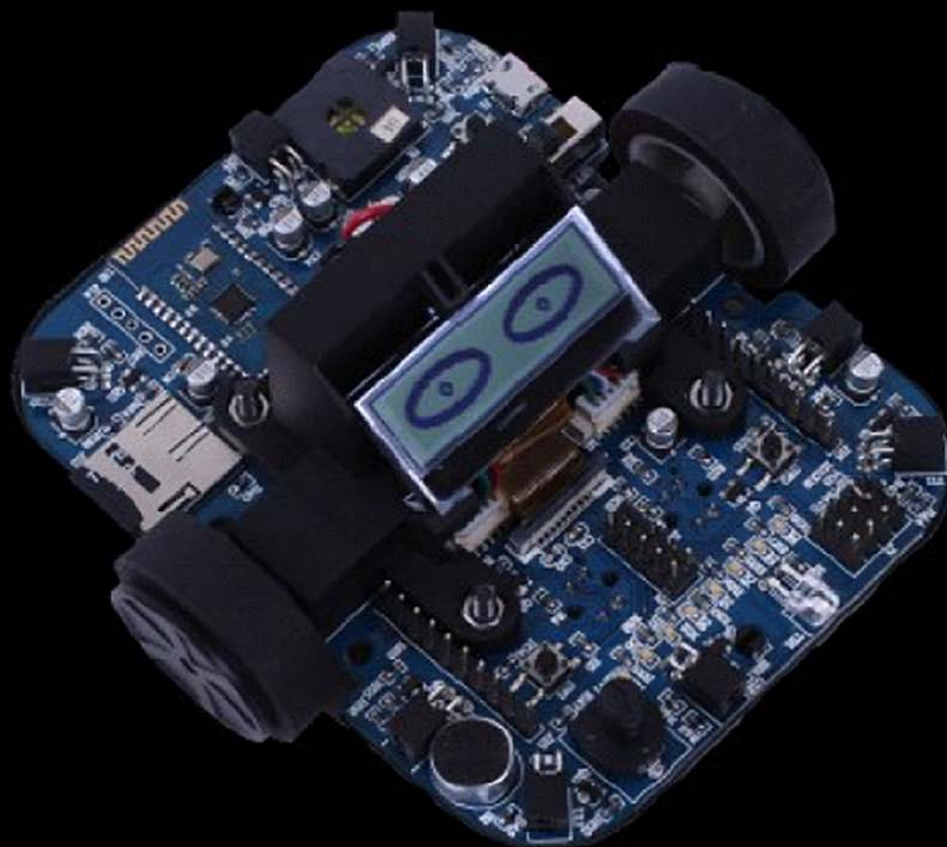


MATRIX | ROBOT ARM

Robotics Course - Instructional Guide



CP5894

MATRIX

www.matrixtsl.com

Copyright © 2016 Matrix Technology Solutions Ltd.

Introduction

Pour commencer	3
Visite guidée - Jetez un coup d'œil aux parties de mon corps	4
Langue neutre - Interface de programme d'application	8
Pseudo-code - Une aide simple à la programmation	10

Configuration du matériel

PC Windows	11
Tablette / Téléphone Android	13
Raspberry Pi et Linux	15

Contrôler le robot

Utilisation de Flowcode	17
Utiliser App Inventor	19
Utilisation de C++ / C# / VB	22
Utilisation de Python	24
Utilisation de LabView	26
Utiliser Scratch	27

Feuilles de travail

Fiche de travail 1 - Le robot fait ses premiers pas	29
Fiche de travail 2 - Attention, ces lumières sont animées	31
Fiche de travail 3 - Allumer, éteindre	33
Fiche de travail 4 - Hello world	35
Fiche de travail 5 - Quelle est la luminosité de cette lumière ?	38
Fiche de travail 6 - À la recherche de la lumière	40
Fiche de travail 7 - Suivez ma ligne	42
Fiche de travail 8 - Les boutons-poussoirs font le travail	45
Feuille de calcul 9 - Panneau d'état	47
Fiche de travail 10 - Incliner et tourner (à l'aide d'un appareil mobile)	49
Fiche de travail 11 - Le gaucher peut naviguer dans un labyrinthe	51
Fiche de travail 12 - Jouer cet air	53
Feuille de travail 13 - Robo-DJ	55
Feuille de travail 14 - Bug mobile	57
Fiche de travail 15 - Robo-Pop	60
Aller plus loin - Défis 16-20	62

Annexe

Tableau de référence de l'API	63
Connexions des broches du microcontrôleur	67
Fréquences des notes de musique	68
Définition du nom du robot	69
Recharger le micrologiciel de l'API AllCode	70

Par où commencer ?

Félicitations. Vous avez maintenant un robot Formula AllCode à la pointe de la technologie qui peut effectuer toutes sortes de manœuvres intéressantes, naviguer dans un labyrinthe, danser, émettre des sons, jouer de la musique, enregistrer et lire de l'audio. La liste est encore longue. Comme il y a tant de choses que vous pouvez faire avec le robot, le premier problème que vous pouvez rencontrer est de trouver le meilleur endroit pour commencer.



Ce guide d'instruction a été préparé pour vous aider à apprendre le plus rapidement possible afin que vous puissiez vous amuser à faire faire au robot les choses que vous voulez qu'il fasse.

S'amuser

L'un des éléments clés de ce guide pédagogique est de veiller à ce que vous vous amusiez. L'apprentissage ou l'acquisition d'une nouvelle compétence est toujours beaucoup plus gratifiant lorsque la tâche est associée à beaucoup de plaisir. Les feuilles de travail de ce guide ont été conçues pour que vous puissiez vous amuser pendant des heures tout en apprenant les bases de la robotique. Faire en sorte que le robot fasse ses premiers mouvements, puis émette ses premiers sons, peut être un défi au départ - mais devrait finir par être un pur plaisir et une satisfaction personnelle lorsque vous y parviendrez.

Créativité

Bien que tu puisses utiliser le robot seul, c'est beaucoup plus amusant lorsque tu fais équipe avec un ou plusieurs de tes amis, car tu verras que chacun d'entre vous a des idées et des suggestions de choses à essayer. Tu seras surpris de voir comment une simple idée peut être transformée par un peu de créativité ou d'imagination collective en quelque chose de vraiment étonnant !

Blocs de construction

Les feuilles de travail de ce guide sont structurées de manière à ce qu'elles commencent par des tâches simples, comme faire fonctionner les DEL ou afficher un message sur l'écran LCD, avant de passer à des tâches plus complexes, comme naviguer dans un labyrinthe ou suivre un chemin. Chaque feuille de travail comporte une barre de couleur qui indique le "niveau de compétence" suggéré - facile, intermédiaire ou avancé.

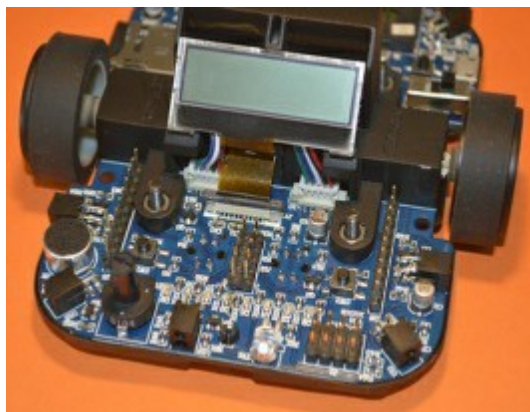
Les exercices de certaines feuilles de travail sont "autonomes", ce qui signifie que vous pouvez vous y plonger et les essayer dans l'ordre que vous souhaitez. D'autres sont "liés" et utilisent une approche modulaire qui incorpore le travail que vous avez rencontré dans une activité précédente. Là encore, cela est indiqué par une barre de couleur en haut de la feuille de travail.

Planifier votre travail

Il est conseillé de commencer par établir un plan de ce que vous souhaitez réaliser chaque fois que vous utilisez le robot. Cela vous permettra de vous concentrer et d'être productif.

Faire le tour

Regardez les parties de mon corps



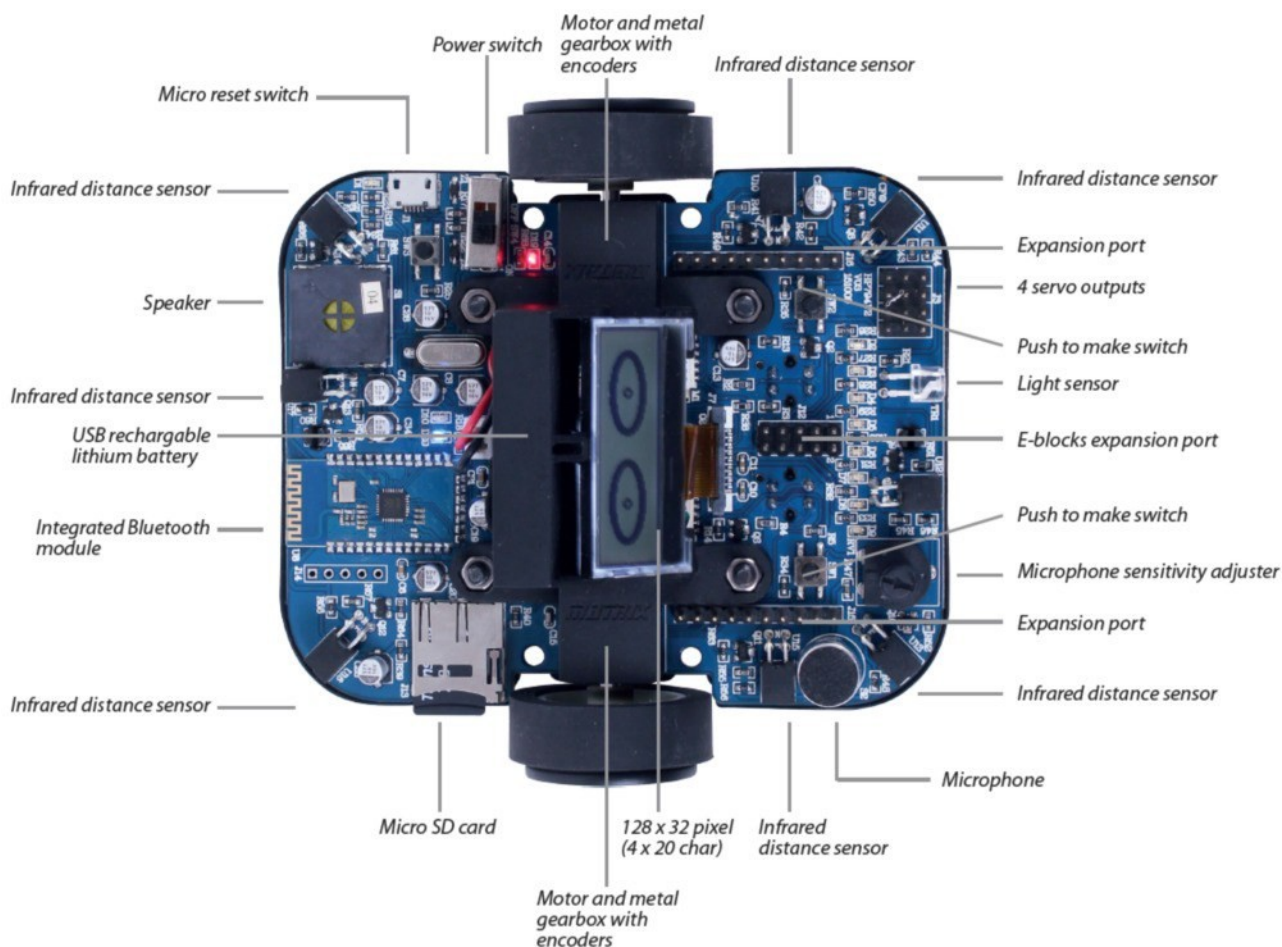
Cette section présente les différentes parties du robot et explique le fonctionnement de base de tous les capteurs et systèmes de contrôle.

Outre les deux moteurs qui peuvent être utilisés pour faire naviguer le robot dans un labyrinthe, il y a des capteurs de lumière et de distance ainsi qu'un équipement d'enregistrement sonore qui vous permettront de vous amuser pendant des heures.

Vous voudrez probablement revenir sur cette page au fur et à mesure que vous lirez ce guide.

Pièces maîtresses du robot

Le schéma fonctionnel ci-dessous montre les éléments importants du robot Formula AllCode.



Faire le tour

Regardez les parties de mon corps

Contrôleur de signal numérique

Au cœur du robot se trouve un contrôleur de signaux numériques (dsPIC®) fabriqué par Microchip Technology Inc. Ce dispositif à haute performance offre une commande de moteur (PWM), des fonctions analogiques avancées, un certain nombre d'interfaces de communication, y compris la capture, le traitement et la lecture de données audio, ainsi qu'un nombre considérable de compteurs et de temporisateurs à grande vitesse.



Moteurs

Le robot est équipé de deux puissants moteurs à aimant permanent, couplés à des réducteurs (utilisés pour augmenter le couple) qui entraînent des roues en caoutchouc (pour assurer l'adhérence sur les surfaces lisses). Un système de contrôle électronique très efficace permet de réduire la consommation d'énergie des moteurs. est aussi faible que possible. Cela signifie que vous pouvez vous amuser beaucoup avec le robot avant que la batterie lithium-ion n'ait besoin d'être rechargée. Les deux moteurs peuvent être commandés indépendamment pour permettre au robot de pivoter sur son propre axe et de s'échapper de l'espace. les situations difficiles ou délicates.

Les encodeurs de rétroaction des moteurs permettent au dsPIC® de "savoir" exactement jusqu'où chacun a tourné - ce qui permet au robot d'avancer ou de reculer d'une distance donnée ou de tourner à un angle donné.

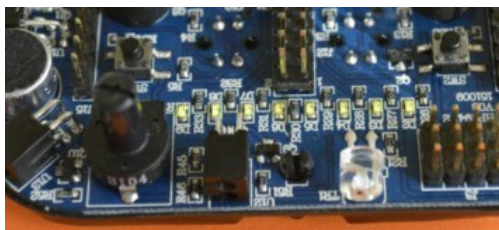


Écran LCD

Au centre du robot se trouve un écran graphique LCD qui peut être utilisé pour afficher des messages et/ou des formes graphiques simples.

LED

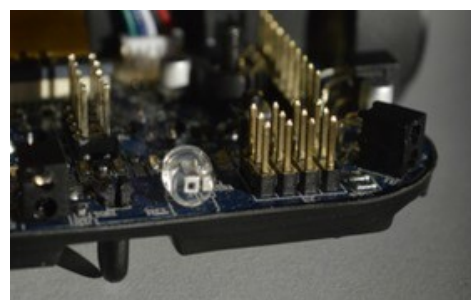
Sur le bord avant du robot se trouve une rangée de DEL vertes qui peuvent être pilotées par des commandes de votre programme.



Capteur de lumière

Un capteur de lumière est placé le long du bord avant du robot. Il peut être utilisé pour mesurer et transmettre à votre programme la quantité de lumière tombant sur le robot.

Si cette fonction est associée au contrôle des moteurs, elle permet au robot de trouver ou d'éviter une source de lumière.

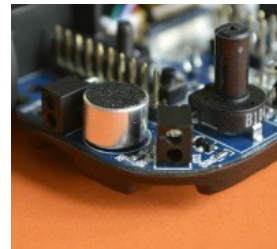


Faire le tour

Regardez les parties de mon corps

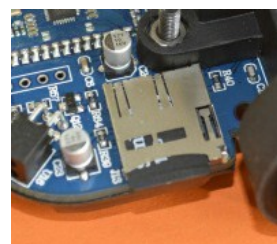
Détecteurs de distance

Autour des quatre côtés et des quatre coins du robot sont montés des émetteurs et des récepteurs infrarouges (IR) qui peuvent mesurer et transmettre à votre programme la distance entre le robot et un éventuel obstacle. Chacun des huit émetteurs IR peut être chargé d'envoyer un faisceau de lumière infrarouge. Si le faisceau rencontre un obstacle, il est réfléchi vers le robot où il est détecté par le récepteur IR.



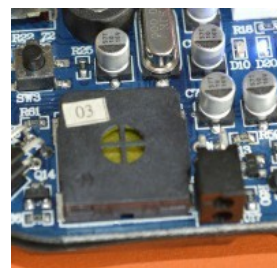
Microphone, emplacement pour carte microSD et un haut-parleur miniature

Ces trois éléments fonctionnent ensemble pour permettre de capturer des sons et des voix (à l'aide du microphone), de les stocker sur la carte microSD et de les restituer à l'aide du haut-parleur embarqué du robot.



Des sons et de la musique préenregistrés, préalablement inscrits sur la carte, peuvent également être joués.

Les notes générées par le robot peuvent être jouées et reliées entre elles pour produire des sons musicaux.



Connecteur micro USB

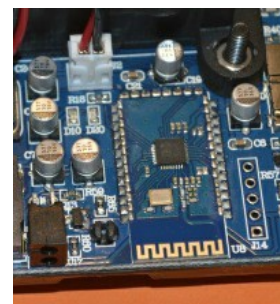
Ce connecteur a deux fonctions. Il peut être utilisé pour recharger la batterie au lithium du robot en y branchant un chargeur comme celui que vous utilisez probablement avec votre téléphone portable. Il peut également être connecté à un ordinateur/portable/tablette (avec une prise USB) pour télécharger et exécuter les programmes que vous avez écrits.



Module Bluetooth embarqué

Il s'agit d'une petite carte fille montée sur le bord arrière du robot. Elle fournit une autre méthode pour télécharger des programmes sur le robot.

L'une de ses principales utilisations est de vous permettre de vous connecter sans fil au robot et d'interagir avec les capteurs embarqués, de contrôler les moteurs, d'écrire des messages sur l'écran LCD, de faire fonctionner les diodes électroluminescentes, de reproduire des sons, etc.

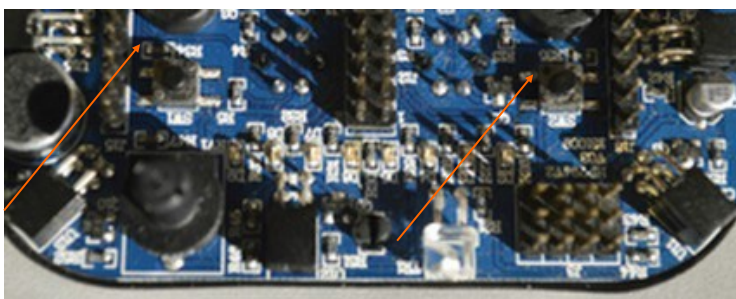


Faire le tour

Regardez les parties de mon corps

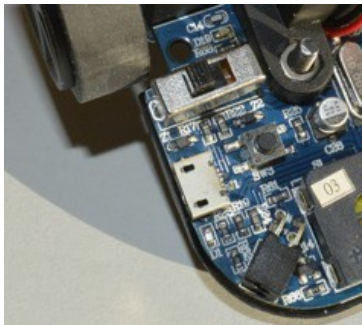
Boutons poussoirs

Le robot comporte trois boutons-poussoirs étiquetés SW1, SW2 et SW3. Les deux premiers boutons peuvent être échantillonnés par votre programme informatique pour déclencher une tâche définie par l'utilisateur. SW3 effectue une réinitialisation matérielle, pour redémarrer votre programme, lorsqu'il est enfoncé.



Interrupteur à glissière

Il s'agit de l'interrupteur principal de mise en marche et d'arrêt du robot.



Détecteurs de ligne

Si vous retournez le robot, vous verrez deux détecteurs de lignes. Chaque unité envoie un faisceau de lumière. En fonction de la couleur et du type de surface (réfléchissante ou non) qui se trouve sous le robot, la quantité de lumière renvoyée est déterminée.

Les détecteurs de lignes sont principalement utilisés pour permettre au robot de suivre une trajectoire dessinée sur une feuille de papier ou imprimée sur un tapis.

Pour ce faire, il suffit d'écrire un programme qui détecte lorsque le robot s'écarte du bord de la trajectoire et qui ajuste la vitesse des deux moteurs pour ramener le robot sur la bonne voie.



Faire le tour

Regardez les parties de mon corps



Indépendant de la langue, agnostique, neutre du point de vue linguistique, indépendant de la plate-forme - que signifient ces termes ?

Cela signifie essentiellement que vous n'êtes pas obligé d'utiliser un certain langage de programmation ou une certaine plate-forme pour contrôler le robot.

En effet, Formula AllCode offre une interface de programme d'application (API) qui vous permet d'interagir avec le robot à l'aide d'un ensemble de routines ou de protocoles simples.

Si vous n'avez jamais entendu parler des API, cette section vous aidera à comprendre comment les utiliser avec le robot.

Une façon d'expliquer cela est de penser au fonctionnement d'une télécommande de télévision. Bien que les téléviseurs modernes soient dotés de boutons tactiles ou de zones tactiles douces sur les bords de l'écran, la plupart des gens trouvent plus pratique d'utiliser une télécommande pour allumer/éteindre le téléviseur, changer de chaîne, régler le volume ou la luminosité, etc.

Quand on y pense, la télécommande d'un téléviseur est un appareil très simple composé d'un clavier et d'un bouton de commande. un faisceau de lumière infrarouge. Lorsqu'un bouton est pressé, sa valeur est codée et utilisée pour envoyer un signal binaire. par le biais du faisceau infrarouge, au téléviseur. Le téléviseur décode le motif reçu et exécute la fonction requise.



Si, pour une raison quelconque, la télécommande du téléviseur tombait en panne, il serait facile de la remplacer par une nouvelle, ou bien même acheter une télécommande universelle (si vous avez un certain nombre d'appareils à contrôler).

Il existe des applications qui permettent de transformer un smartphone en télécommande de télévision, ainsi que du matériel capable d'envoyer des codes de télécommande de télévision. La seule chose à faire est d'envoyer le motif correct (c'est-à-dire la commande) au moment voulu. On pourrait donc dire que le téléviseur possède une API qui permet à une télécommande (quelle que soit sa forme) de contrôler l'intelligence ou les systèmes de contrôle électronique du téléviseur. L'API, telle qu'elle est utilisée sur le robot Formula AllCode, offre la même indépendance de plate-forme et de langage que l'exemple de la télévision décrit ci-dessus. La différence est que le support de transmission du robot est Bluetooth plutôt qu'un faisceau infrarouge. Cela signifie que si vous disposez d'une installation Bluetooth sur votre système, vous avez la liberté d'utiliser votre système de communication préféré.

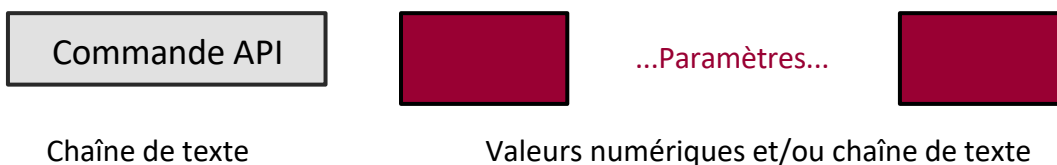
et un langage de programmation pour interagir avec le robot.

Par exemple, vous pouvez choisir d'utiliser un téléphone portable Bluetooth pour contrôler le robot. Vous pouvez également connecter un PC/Mac/Raspberry Pi® ou une carte Matrix Multiprogrammer en y installant un module Bluetooth bon marché pour créer un système de contrôle simple.

Comme l'API réagit à un simple protocole textuel (comme indiqué ci-dessous), vous avez la liberté d'utiliser des langages de programmation formels tels que C, C#, C++ ou Python, ou des langages graphiques ou à base d'icônes tels que Flowcode, App Inventor ou LabView.

L'autre point à noter à propos de l'API est que certaines commandes sont bidirectionnelles. Cela signifie qu'une commande envoyée au robot peut entraîner le retour d'une valeur. Les capteurs de distance infrarouges installés sur le robot en sont un bon exemple. Une commande API peut être envoyée à un capteur spécifique (qui l'interroge), ce qui lui permet de renvoyer une valeur numérique de la distance entre le robot et un obstacle.

Le format général de l'API du robot est présenté ci-dessous.



Chaque commande commence par une chaîne de texte qui identifie ce que le robot doit faire. Elle peut être suivie d'un ou plusieurs paramètres. En fonction de ce que vous essayez de faire avec le robot, ces paramètres peuvent être numériques, textuels ou un mélange des deux.

Par exemple, pour envoyer une valeur aux huit diodes électroluminescentes situées à l'avant du robot, vous utiliserez :

LEDWrite <valeur>

Comme les LED sont regroupées et forment une rangée de 8 bits, elles peuvent être pilotées en leur envoyant un nombre binaire. Le paramètre <valeur> peut donc prendre une valeur comprise entre 0 et 255.

Il convient de noter que les commandes API d'un langage particulier peuvent présenter des différences subtiles. Par exemple, Python utilisera quelque chose comme "fa.LEDWrite(20)" alors que C# ressemblera à "FA_DLL.FA_LEDWrite(20) ;" et pour App Inventor, Flowcode et LabView, l'icône appropriée sera sélectionnée.

Voici un autre exemple qui montre comment contrôler les moteurs du robot.

Avant <distance>

Le paramètre intitulé <distance> peut prendre une valeur comprise entre 1 et 1000 pour définir la distance parcourue par le robot (en millimètres) en ligne droite.

Si vous souhaitez contrôler indépendamment les deux moteurs, vous pouvez utiliser cette commande.

Setmotors <gauche> <droite>

Les paramètres <gauche> et <droite> peuvent prendre une valeur comprise entre -100 et 100 pour définir la vitesse et la direction de chaque moteur, ce qui permet au robot de se déplacer dans différentes directions.

L'API est répertoriée dans l'annexe et les feuilles de travail de ce guide l'expliquent plus en détail.

Pseudo-code

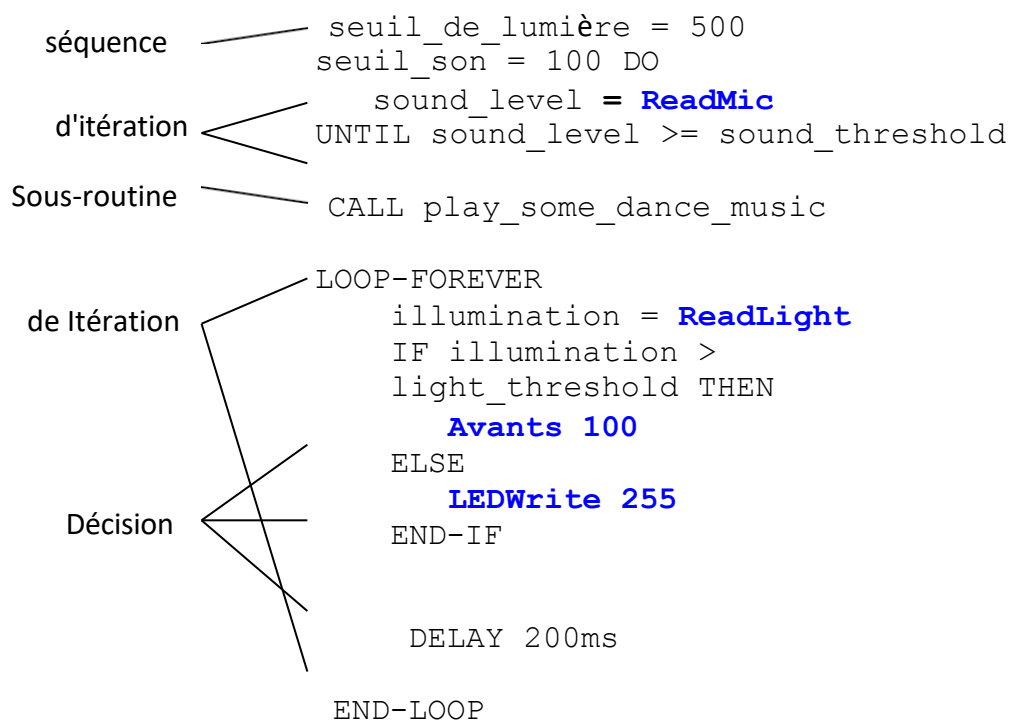
Une simple aide à la programmation

La plupart des exemples figurant dans ce guide pédagogique sont rédigés sous forme de pseudo-code. Vous vous demandez peut-être pourquoi nous n'avons pas utilisé d'organigrammes ou l'un des nombreux langages de programmation qui peuvent être utilisés avec le robot Formula AllCode.

La réponse est double. Premièrement, ce guide serait énorme si tous les exemples étaient rédigés dans tous les langages pris en charge par le robot, et seules certaines parties vous seraient applicables.

Deuxièmement, si vous utilisez des organigrammes, rien ne vous empêche de créer des structures qui ne correspondent pas facilement aux trois principales structures de contrôle (par exemple, séquence, décision, etc.).

itération) que les langages de programmation formels prennent en charge. Voici un exemple de programme, écrit en pseudo-code, qui utilise les trois constructions pour le robot.



Une séquence est un groupe d'une ou plusieurs instructions qui se suivent. Il y a deux exemples d'itération dans ce morceau de pseudo-code : le premier se répète jusqu'à ce qu'un certain événement se produise et le second se répète indéfiniment. Une décision, parfois appelée branche, peut prendre l'une des deux voies suivantes en fonction de la réponse à une question. Cet exemple montre également un `CALL` qui saute à un sous-programme. Il est utilisé pour rendre le plus facile à lire en masquant certains détails inutiles.

Les éléments surlignés en bleu foncé sont des appels API qui vous permettent d'interagir avec le robot. Les noms de ces appels correspondent à toutes les macros disponibles dans Flowcode, App Inventor, Python et autres langages pris en charge par le robot Formula AllCode.

L'utilisation du pseudo-code vous permet de faire le premier pas vers la mise en pratique de vos idées de manière structurée, sans vous heurter à la syntaxe d'un langage de programmation formel. Une fois que vous avez exprimé vos idées à l'aide du pseudo-code, vous pouvez passer à l'écriture du programme proprement dit en utilisant le langage de programmation de votre choix, une tâche de codage assez simple.

Configuration du matériel

PC Windows



De nombreux appareils Windows, en particulier les ordinateurs portables et les tablettes, sont dotés de la fonctionnalité Bluetooth intégrée. Si ce n'est pas le cas de votre PC, vous devrez utiliser un dongle USB Bluetooth 2.

Les différentes versions du système d'exploitation Windows utilisent des méthodes légèrement différentes pour connecter les périphériques Bluetooth, mais elles suivent toutes les mêmes étapes.

Vous ne devrez effectuer cette opération qu'une seule fois, car Windows se souviendra des appareils appariés.

1) Activer le Bluetooth

Souvent, le Bluetooth est activé par défaut et vous pouvez généralement ignorer cette étape. Toutefois, si ce n'est pas le cas, il peut être activé dans les paramètres de Windows et/ou dans le panneau de configuration. Très occasionnellement, Bluetooth doit être activé à l'aide d'un commutateur spécial ou d'une touche de fonction. Veuillez consulter l'aide de votre PC ou de Windows pour plus d'informations.

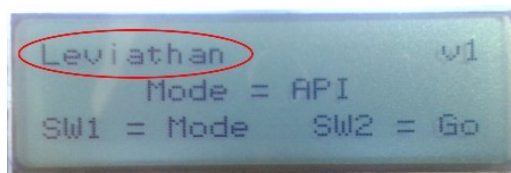
Manage Bluetooth devices

Bluetooth
 On

2) Appairer le robot

Allumez d'abord le robot - son nom s'affiche en haut à gauche de l'écran.

Là encore, l'appariement fonctionne légèrement différemment selon les versions de Windows et il est donc difficile de donner des instructions spécifiques ici. L'aide et le site web de Windows contiennent des guides qui expliquent comment procéder.



Lors de l'appairage, un écran ou une liste des appareils Bluetooth disponibles s'affiche.

Sélectionnez l'appareil portant le nom de votre robot et cliquez sur Suivant ou Appairer.

Il vous sera demandé de saisir le code d'appariement. Le robot Formula AllCode utilise le code par défaut de 1234, mais vous pouvez le remplacer par un autre code si vous voulez vous assurer que personne d'autre ne peut s'appairer avec votre robot.

Une fois le code saisi, Windows confirme qu'il a été apparié avec le robot.

Configuration du matériel

PC Windows

3) Déterminer le numéro du port COM

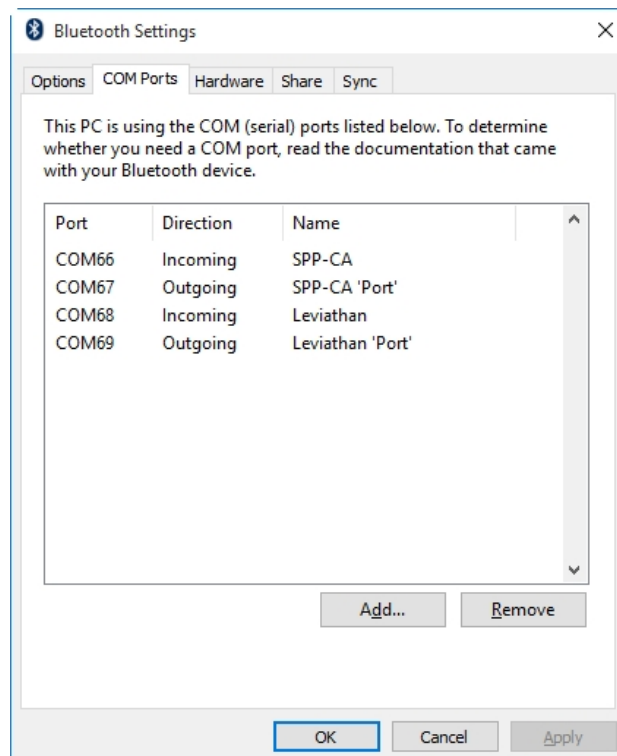
Vous devriez obtenir une fenêtre contextuelle dans la barre des tâches indiquant que l'appareil est prêt à être utilisé. Si vous cliquez sur cette bulle avant qu'elle ne disparaisse, vous pouvez connaître le port COM attribué au robot.

Vous utilisez ce numéro de port COM lorsque vous communiquez avec le robot Formula AllCode et ce numéro de port COM restera le même tant que vous ne retirerez pas ou ne désassemblerez pas le robot de Windows.

Si vous n'avez pas vu le port COM lors de l'appairage du robot, vous pouvez le retrouver dans le menu Bluetooth. La fenêtre Paramètres est illustrée à droite.

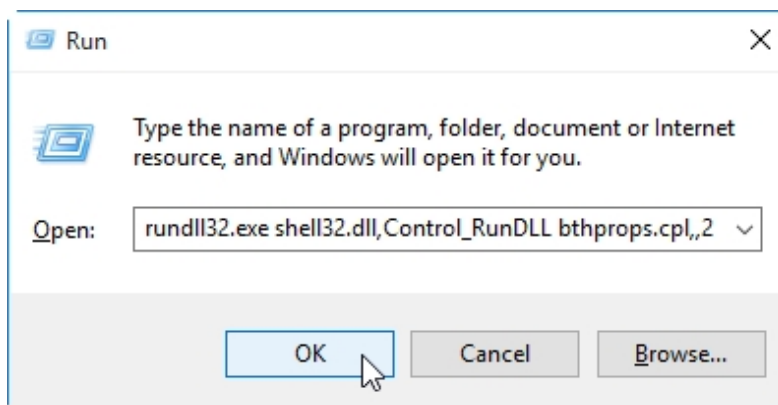
Deux ports COM sont indiqués pour chaque robot. Veillez à toujours utiliser le numéro de port "sortant".

Cette fenêtre peut être un peu difficile à trouver sur certaines versions de Windows. Par exemple, sous Windows 10, vous pouvez la trouver via le lien "Plus d'options Bluetooth" sur l'écran des paramètres Bluetooth.



Heureusement, il existe un moyen garanti d'ouvrir ce dossier dans toutes les versions de Windows à partir de la version 7. Ouvrez la fenêtre "Exécuter..." en maintenant la touche Windows enfoncée et en appuyant sur R, puis sur tapez (ou copiez et collez) la commande suivante dans la boîte et appuyez sur "OK" :

rundll32.exe shell32.dll,Control_RunDLL bthprops.cpl,,2



Maintenant que vous avez apparié le robot et déterminé le numéro du port COM, vous pouvez utiliser l'un des nombreux langages de programmation disponibles sous Windows pour contrôler le robot Formula AllCode.



Les téléphones et tablettes Android, lorsqu'ils sont utilisés avec un logiciel de programmation intuitif tel que App Inventor, constituent une plateforme motivante pour contrôler le robot Formula AllCode.

Ces appareils intègrent presque toujours la technologie Bluetooth.

Comme pour les autres appareils, le robot Formula AllCode doit être apparié au téléphone ou à la tablette avant de pouvoir être utilisé.

Si la fonctionnalité Bluetooth est déjà intégrée à votre téléphone ou à votre tablette, vous devez d'abord l'activer en cliquant sur Paramètres -> Connexions.

Bluetooth désactivé



Bluetooth



Bluetooth activé



Bluetooth



Une fois le Bluetooth activé, vous devez appairer le robot Formula AllCode à votre téléphone pour permettre aux applications de voir l'appareil.

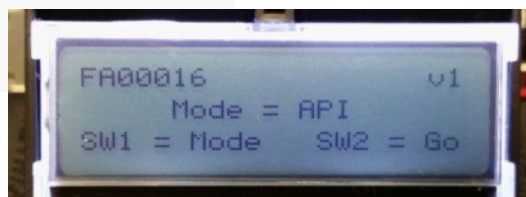
Commencez par cliquer sur l'option Bluetooth dans Paramètres -> Connexions.

Ensuite, assurez-vous que votre robot est allumé et cliquez sur le bouton Scan de votre appareil Android pour rechercher de nouveaux appareils Bluetooth. Notez qu'il se peut que vous deviez faire défiler l'écran vers le bas pour voir les résultats de l'analyse.

Available devices



FA00016



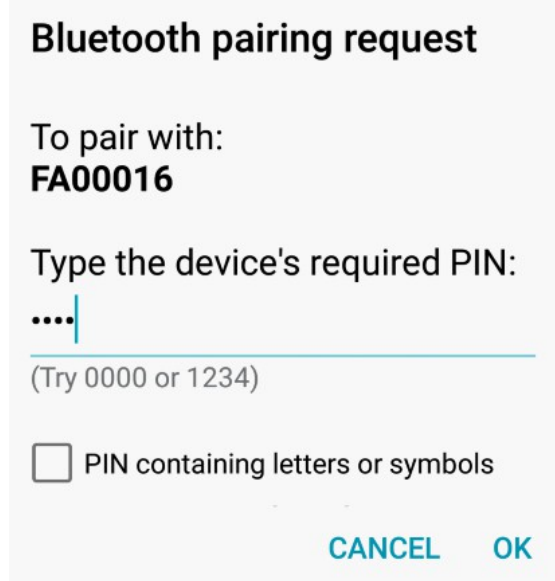
Le nom du Formula AllCode apparaît en haut à gauche de l'écran LCD pour vous indiquer le nom Bluetooth du robot auquel vous souhaitez vous connecter.

Configuration du matériel

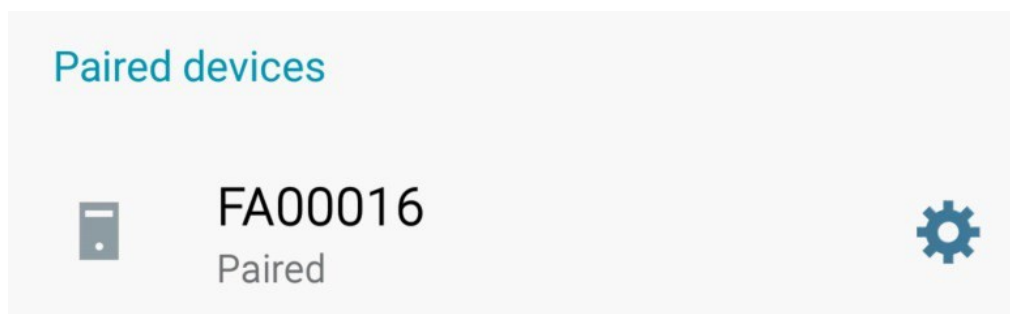
Tablette / téléphone Android

Lorsque le nom de l'appareil apparaît, cliquez sur le nom de l'appareil et vous serez invité à entrer la clé de la paire.

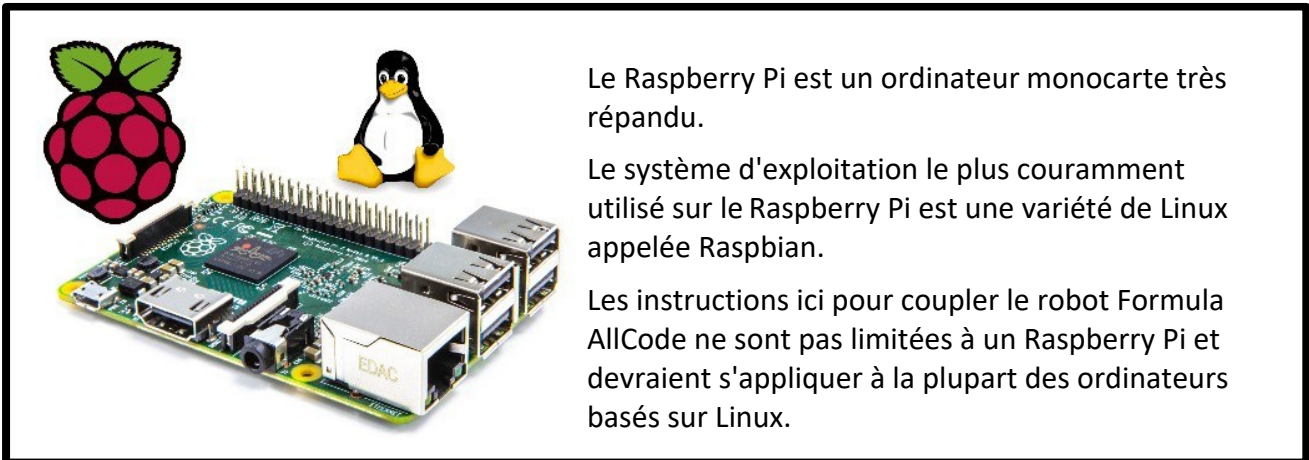
La clé par défaut est 1234.



Une fois l'appareil apparié, il sera listé avec tous les autres appareils Bluetooth appariés que vous pourriez avoir et sera prêt à être utilisé avec toutes les applications Formula AllCode que vous téléchargez ou créez.



Remarque : cette procédure peut être légèrement différente sur votre appareil Android. Pour plus de détails sur votre Téléphone ou tablette, veuillez consulter la procédure d'appairage des appareils Bluetooth pour votre appareil spécifique.



La configuration de Bluetooth est relativement facile sur un Raspberry Pi et peut être réalisée de plusieurs manières. Les étapes suivantes sont peut-être plus complexes, mais elles devraient fonctionner dans toutes les situations. Notez que le Pi a besoin d'un dongle USB Bluetooth.

Étape 1 - Paramètres Bluetooth

Ouvrez un terminal de ligne de commande et tapez la commande "hciconfig". Vous obtiendrez ainsi une liste des périphériques Bluetooth disponibles sur votre RPi. La chose importante à noter est l'identifiant du module Bluetooth - dans mon cas, il s'agit de "hci0" :

```
pi@raspberrypi / $ hciconfig
hci0: Type: BR/EDR Bus: USB
      BD Address: 00:15:83:15:A3:10 ACL MTU: 339:8 SCO MTU: 128:2
      UP RUNNING PSCAN ISCAN
      RX bytes:1420 acl:0 sco:0 events:53 errors:0
      TX bytes:452 acl:0 sco:0 commands:46 errors:0
```

Étape 2 - Détecter la formule AllCode

Allumez le robot et tapez "hcidtool scan". Lorsque j'ai fait cela, deux appareils sont apparus. Le mien était le second ("API_B") et vous devrez prendre note des 6 paires de nombres hexadécimaux qui constituent l'adresse MAC, un identifiant unique pour le robot - dans mon cas, "00:BA:55:23:1C:20".

```
pi@raspberrypi / $ hcidtool scan
Scanning ...
00: BA: 55: 23: 1C: 16      FormAllCode
00: BA: 55: 23: 1C: 20      API_B
```

Étape 3 - Jumeler le robot avec le RPi

Pour l'appariement, vous pouvez utiliser la commande "bluezsimple-agent" en utilisant "hci0" et "MAC" trouvés dans les étapes précédentes.

```
pi@raspberrypi / $ sudo bluez-simple-agent hci0 00:BA:55:23:1C:20
RequestPinCode (/org/bluez/2342/hci0/dev_00_BA_55_23_1C_20)
Enter PIN Code: 1234
Release
New device (/org/bluez/2342/hci0/dev_00_BA_55_23_1C_20)
```

Étape 4 - Rendre le changement permanent

La dernière étape consiste à faire en sorte que cet appairage se produise automatiquement lors de la prochaine utilisation du RPi. Cela peut être fait en éditant le fichier `/etc/bluetooth/rfcomm.conf` (par exemple en utilisant nano) et en entrant le code suivant. Encore une fois, vous devrez vous assurer que vous utilisez la bonne adresse MAC qui a été trouvée précédemment.

```
pi@raspberrypi / $ sudo nano /etc/bluetooth/rfcomm.conf
```

Vous devrez ajouter à ce fichier `rfcomm.conf` une section similaire à la suivante :

```
rfcomm1 {  
# Lier automatiquement l'appareil au  
démarrage bind yes ;  
  
# Adresse Bluetooth de l'appareil  
00:BA:55:23:1C:20 ;  
  
# Canal RFCOMM pour le canal de  
connexion 1 ;  
  
# Description de la connexion  
commentaire "Formula AllCode" ;  
}
```

Les trois parties de texte en rouge peuvent être personnalisées - vous utiliserez l'adresse MAC trouvée à l'étape 2, et peut utiliser son nom dans le champ "commentaire".

Si vous avez plus d'un robot, vous pouvez ajouter plusieurs sections - il suffit de nommer chacune d'entre elles "rfcomm1", "rfcomm2", etc.

Étape 5 - Test de la connexion

Une fois que vous êtes appariés, vous pouvez tester la connexion en utilisant la commande suivante dans le terminal de ligne de commande :

```
echo "PlayNote 100,100\n" > /dev/rfcomm1
```

Si tout se passe bien, vous devriez entendre un bip de la Formula AllCode.

Si cela ne fonctionne pas et que vous obtenez le message "permission denied", il se peut que vous deviez vous ajouter au groupe "dialout". Pour vérifier si c'est le cas, utilisez la commande "id" avec votre nom d'utilisateur comme paramètre pour vérifier à quels groupes vous appartenez. Si le groupe "dialout" n'est pas listé, vous pouvez vous ajouter au groupe en utilisant la commande suivante (n'oubliez pas de remplacer "nom d'utilisateur" par votre nom d'utilisateur !)

```
sudo usermod -a -G dialout nom d'utilisateur
```

Vous devrez ensuite vous déconnecter et vous reconnecter pour voir ce changement et la "PlayNote"

FC FLOWCODE

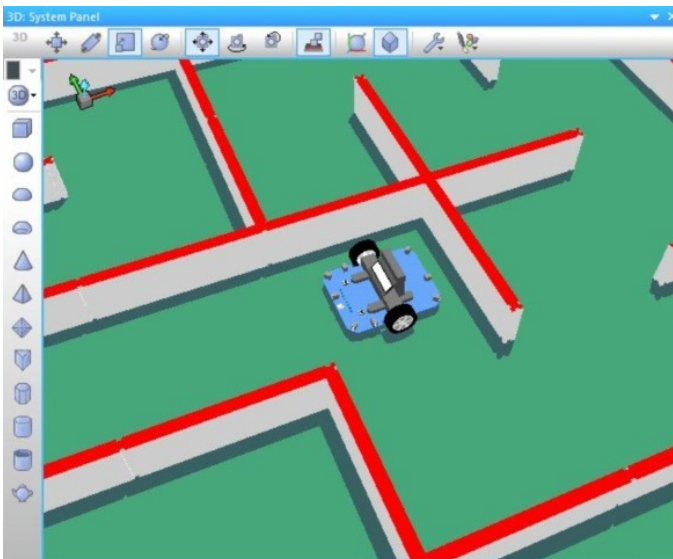
Cette section explique comment utiliser Flowcode pour contrôler le robot.

Comme vous le savez probablement, Flowcode fournit des macros de composants pour tous les dispositifs complexes tels que le bus CAN, ZigBee et le robot. Cela signifie que vous pouvez commencer à apprendre la robotique et comment contrôler le robot Formula AllCode très rapidement et très facilement.

Le site web de Matrix propose des programmes pour vous inspirer et vous aider.

Cette section suppose que vous connaissez les bases de l'utilisation de Flowcode. Il existe deux façons d'utiliser Flowcode pour contrôler le robot Formula AllCode. 1) reprogrammer le firmware du robot et 2) utiliser la fonctionnalité API intégrée.

Dans Flowcode 6.1.3 et les versions ultérieures, deux composants sont disponibles dans le menu mécanique pour vous permettre de sélectionner le mode de fonctionnement souhaité, comme illustré à droite.



Téléchargement du code sur le robot

Le composant Formula AllCode nous permet de créer un code qui s'exécutera sur le microcontrôleur embarqué dans la Formula AllCode.

Le composant est livré avec une simulation opérationnelle permettant de créer des programmes pour suivre des lignes ou résoudre des labyrinthes sans avoir à compiler et à télécharger le code en permanence au robot.

Pour réduire le nombre de macro-fonctions dans le composant Formula AllCode, l'option pour piloter les sorties des servomoteurs, la carte SD et l'accéléromètre n'ont pas été incluses. Il existe des composants séparés qui vous permettent de le faire.

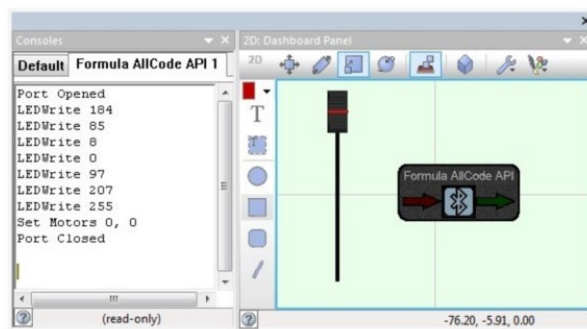
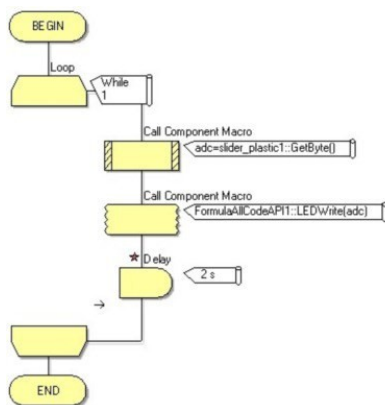
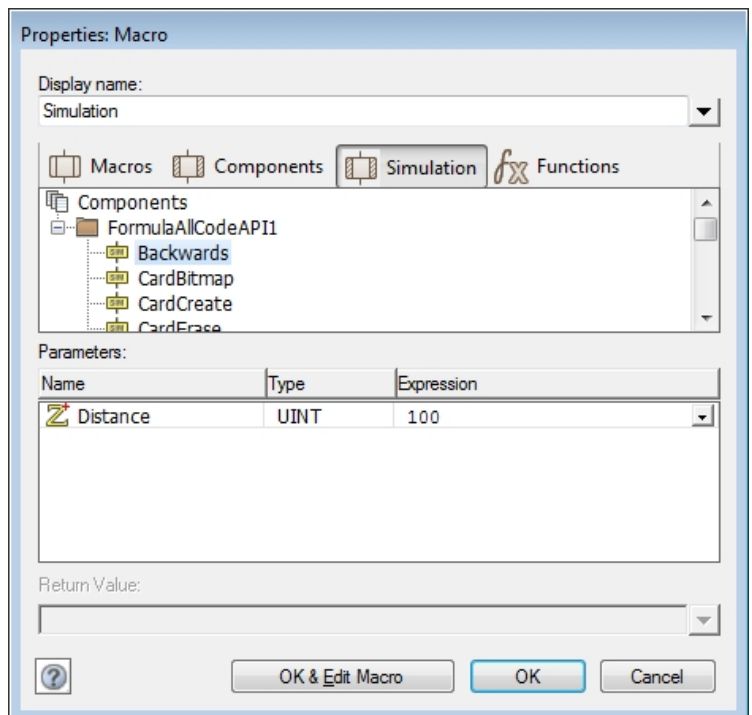
Remarque : le téléchargement de code à l'aide de ce composant supprimera la fonctionnalité API du robot. Instructions pour restaurer l'API afin que le robot retrouve son état d'origine.

Contrôle direct à partir de Flowcode à l'aide de l'API

L'autre composant ("Formula AllCode API") nous permet de contrôler un robot Formula AllCode à partir de Flowcode, sans téléchargement.

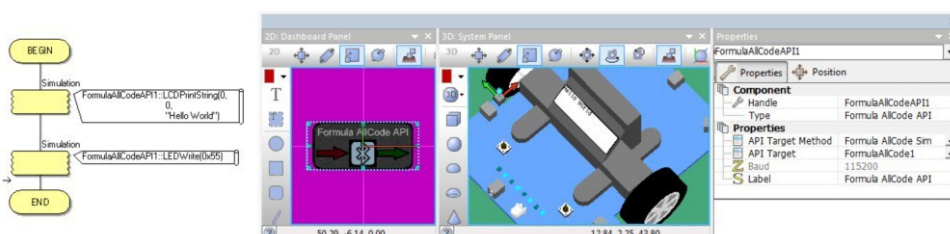
Notez que les fonctions de la macro-composant apparaissent dans l'onglet Simulation plutôt que dans l'onglet habituel des composants. Cela souligne le fait que le code n'est pas téléchargeable sur le site web de la Commission européenne. Formule AllCode.

Ici, nous contrôlons la valeur des DEL en lisant une valeur provenant d'un appareil analogique simulé, potentiomètre à glissière. La console fournit maintenant une liste des appels de fonction de l'API, des paramètres et des valeurs de retour.



Le port de communication sélectionné est automatiquement ouvert lorsque vous démarrez la simulation. La vitesse des moteurs est automatiquement mise à 0 avant de fermer le port à la fin de la simulation.

En faisant également glisser un composant Formula AllCode sur le panneau, vous pouvez également contrôler le robot simulé via le composant API macros.





[Inventeur
d'application
s](#)



[Modèle](#)

Cette section explique comment commencer à coder. appelé App Inventor qui vous permettra d'utiliser un appareil Android pour contrôler le robot.

Ces codes QR et ces hyperliens vous aideront à accélérer votre travail. afin que vous puissiez commencer à vous amuser à coder.

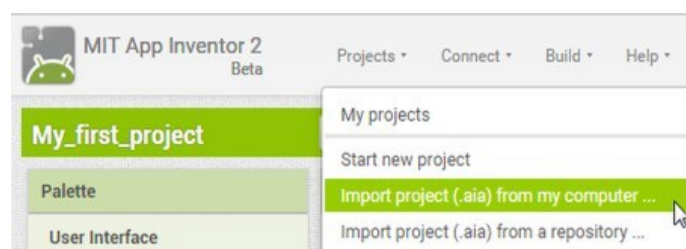
Inventeur d'applications

App Inventor est un langage de programmation graphique disponible gratuitement et hébergé sur l'un des systèmes de stockage et d'informatique en nuage du Massachusetts Institute of Technology (MIT) aux États-Unis. Tout ce dont vous avez besoin pour commencer à développer des applications pour un téléphone mobile ou une tablette Android, c'est d'un navigateur web et d'un compte Google. App Inventor utilise des icônes codées par couleur, en forme de pièces de puzzle, pour créer une application en assemblant les pièces. Le système vous évite de faire des erreurs en veillant à ce que seules certaines formes de la même couleur puissent être assemblées. Cette technique encourage les personnes de tous âges à s'amuser à "coder" et à développer leur confiance et leurs capacités en matière de programmation informatique.

Configuration d'App Inventor

Les éléments clés dont vous avez besoin sont un ordinateur de bureau ou un ordinateur portable (équipé d'un navigateur moderne tel que Chrome ou Firefox) et un téléphone ou une tablette fonctionnant avec le système d'exploitation Android. Vous aurez également besoin d'un lecteur QR ; il serait donc judicieux d'en télécharger un à ce stade sur votre téléphone portable. Il suffit de suivre les étapes suivantes pour être rapidement opérationnel.

1. Créez un compte Google (si vous n'en avez pas déjà un).
2. Rendez-vous sur le site web d'App Inventor en scannant le code QR ou en cliquant sur le lien hypertexte ci-dessus, puis connectez-vous à l'aide de votre compte Google.
3. Suivez les instructions en ligne, y compris l'installation de l'application "MIT AI2 Companion App" sur votre appareil Android.
4. Vous devez relier l'application Web App Inventor à votre téléphone ou à votre tablette. Pour ce faire, sélectionnez "AI Companion" dans le menu "Connect" d'App Inventor.
5. Téléchargez le modèle Formula AllCode sur votre ordinateur en scannant le code QR. ou en cliquant sur le lien hypertexte. Souvenez-vous de l'endroit où vous les avez sauvegardés sur votre bureau/ordinateur portable.



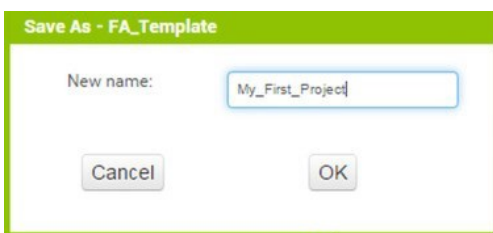
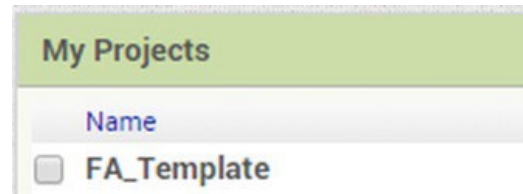
Contrôler le robot

Utiliser App Inventor

Votre premier programme

Chaque fois que vous souhaitez lancer un nouveau projet, suivez les étapes suivantes :

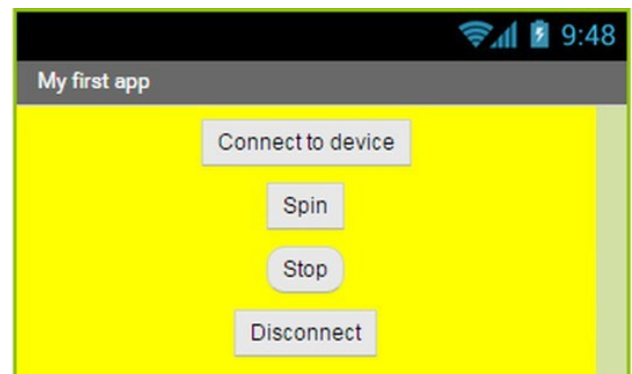
1. Chargez le fichier modèle en cliquant sur "Mes projets" dans le menu App Inventor et en sélectionnant le projet "FA_Template".



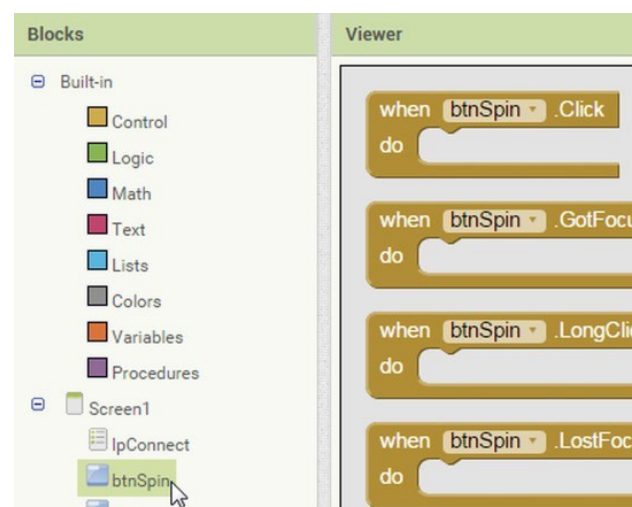
2. Enregistrez ce modèle dans un nouveau fichier en sélectionnant "Enregistrer le projet sous..." dans le menu "Projets", puis en saisissant un nom approprié pour votre projet.

3. Cliquez sur "Screen1" dans le volet "Composants" et définissez le "AppName" et le "Title" dans les "Propriétés" à quelque chose de convenable.

4. Faites glisser un bouton du panneau Interface utilisateur sur l'écran du visualiseur et modifiez son texte pour qu'il se lise "Spin". Renommez également le bouton pour qu'il se lise "btnSpin". Procédez de la même manière pour créer un autre bouton appelé "Stop" et nommé "btnStop".



5. Passez en mode "Blocs" et cliquez sur l'objet "btnSpin" - une liste d'icônes apparaît. Faites glisser l'icône "when btnSpin.Click" dans votre programme.



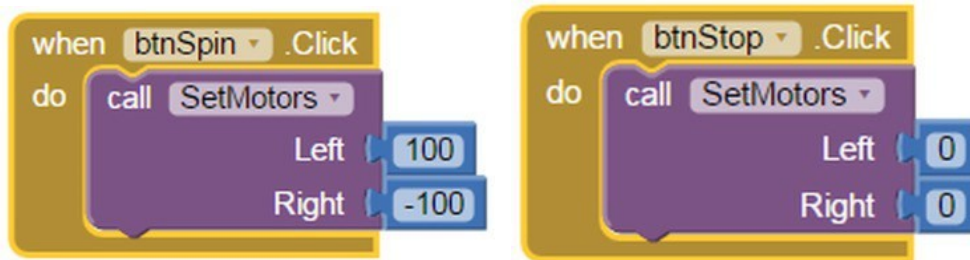
Répétez l'opération, mais cette fois-ci, cliquez sur le bouton Objet "btnStop".

Contrôler le robot

Utiliser App Inventor

6. Cliquez sur "Procedures" dans la liste "Built-in" et faites glisser l'icône "call SetMotors" au milieu de vos icônes "when btnSpin.Click" et "when btnStop.Click". Ajoutez deux valeurs littérales de "Math" comme paramètres gauche et droit des blocs de code SetMotors.

Vos deux blocs devraient ressembler à ceci :



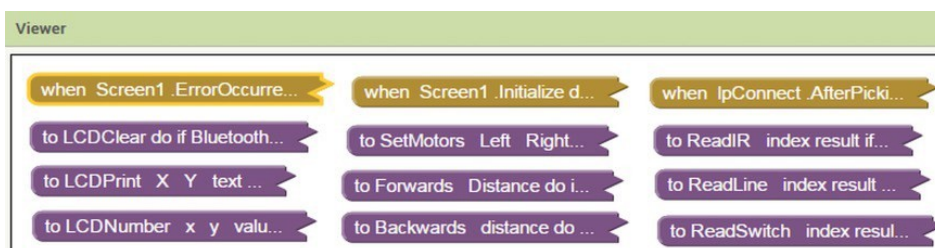
7. Vous devez maintenant construire le projet. Sélectionnez "App (fournir le code QR pour .apk)" dans le menu "Build". Une fois l'opération terminée, lancez l'application "MIT AI2 Companion", puis scannez le code QR sur votre appareil Android à l'aide du bouton "Scan QR Code".

Remarque :

Si vous visitez le site Web d'App Inventor, vous trouverez des instructions sur les autres méthodes disponibles pour transférer votre programme sur votre appareil Android.

8. Vous pouvez maintenant exécuter votre programme sur votre appareil Android. Cliquez sur "Connecter à l'appareil" et sélectionnez le robot Formula AllCode dans la liste. En cliquant sur le bouton "Spin", le robot tourne et en cliquant sur "Stop", il s'arrête.

Vous avez peut-être remarqué un certain nombre d'icônes en haut de l'écran dans App Inventor. Ces icônes définissent les procédures de communication avec le robot Formula AllCode et certains éléments de l'application des fonctions standard pour permettre l'établissement de la liaison Bluetooth.



Les icônes de couleur beige représentent des **événements** tels que le clic sur un bouton, le déclenchement d'une minuterie ou l'apparition d'une erreur.

Les icônes de couleur mauve se rapportent à un ensemble de **procédures** ou de **sous-programmes** qui ont été conçus pour effectuer certaines tâches pour vous. Vous ne



Un outil de programmation courant est Visual Studio de Microsoft.

Dans cette section, nous présentons différentes méthodes pour communiquer avec le robot Formula AllCode en utilisant certaines des méthodes les plus courantes.

des langages de programmation très connus tels que C++, C# et Visual Basic.

L'utilisation de Formula AllCode avec Visual Studio via les langages de programmation Visual C++, Visual C# ou Visual Basic est assez simple et consiste à utiliser une bibliothèque DLL et des fichiers associés fournis par MatrixTSL pour communiquer avec le robot.

Comme pour les autres langages, vous devez utiliser le numéro de port COM auquel le robot est connecté. Vous trouverez des exemples sur les pages Formula AllCode du site web Matrix TSL à l'adresse suivante : <https://www.matrixtsl.com/allcode/resources/>

Utilisation de C#

Le programme ci-contre présente un programme de base en C#.

Vous devez utiliser l'espace de noms "FormulaAllCode" et placer le fichier de bibliothèque FA_DLL dans le même dossier que votre projet. La DLL doit se trouver dans le même dossier que l'EXE que vous créez.

Vous remarquerez que les commandes de l'API Formula AllCode sont les suivantes précédé des caractères "FA_" et doivent également recevoir à chaque fois le port COM comme premier paramètre.

N'oubliez pas de modifier l'API dans l'annexe lors de leur utilisation.

N'oubliez pas non plus de fermer la COM à la fin de votre programme !

```
en utilisant System ;

namespace FormulaAllCode
{
    classe MonProgramme
    {
        static void
        Main(string[] args)
        {
            //Affecter le numéro de
            port char iPort = (char) 9 ;

            //Ouvrir le port COM
            FA_DLL.FA_ComOpen(iPort)
            ;
            //jouer un air
            FA_DLL.FA_PlayNote(iPort, 523,
            400) ;
            FA_DLL.FA_PlayNote(iPort, 659,
            400) ;
            FA_DLL.FA_PlayNote(iPort, 784,
            800) ;

            //Fermeture du port COM
            FA_ComClose(iPort) ;
        }
    }
}
```

Contrôler le robot

Utilisation de C++ / C# / VB

Utilisation de VB

Le même programme est présenté à droite, cette fois en Visual Basic.

Vous constaterez que le programme est très similaire au programme C#, avec seulement quelques différences mineures dans la syntaxe. Les appels à l'API Formula AllCode sont identiques.

Le fichier FA_API.vb doit être ajouté à votre projet.

N'oubliez pas de placer le fichier "FASlave.DLL" dans le même dossier que l'EXE que vous avez créé.

Utiliser le C++

Un programme légèrement différent est présenté pour C++, en dessinant cette fois un triangle équilatéral.

Pour utiliser la DLL avec C++, vous avez besoin de pour référencer les fonctions en incluant le fichier d'en-tête "FA_API.h". Vous devez également ajouter le fichier "FASlave.lib" à votre projet Visual Studio.

Placez également la DLL dans le même dossier que l'EXE que vous créez.

Comme pour les autres langues, les appels à l'API Formula AllCode sont très similaires, ce qui signifie qu'il est très facile d'utiliser le robot avec différentes langues - à condition de connaître les bases de la langue en question !

```
Module Module1
```

```
    Sous Main()
```

```
        'Attribuer le numéro de  
        port Dim iPort As Byte  
        iPort = 9
```

```
        Ouvre le port COM  
        FA_ComOpen(iPort)
```

```
        Jouer un air  
        FA_PlayNote(iPort, 523, 400)  
        FA_PlayNote(iPort, 659, 400)  
        FA_PlayNote(iPort, 784, 800)
```

```
        Ferme le port COM FA_ComClose(iPort)
```

```
    End Sub
```

```
End Module
```

```
#include "stdafx.h"  
#include "FA_API.h"
```

```
int main()
```

```
{
```

```
    //Assigner et ouvrir le port COM char  
    iPort = 9 ;  
    FA_ComOpen(iPort) ;
```

```
    /Dessiner un triangle  
    for (int i=0 ; i<3 ; i++)
```

```
    {  
        FA_Forwards(iPort, 500) ;  
        FA_Left(iPort, 120) ;
```

```
    }
```

```
    //Fermeture du port COM
```

```
    FA_ComClose(iPort) ;
```

```
    retour 0 ;
```

```
}
```



Python est un langage de programmation informatique largement utilisé et disponible sur de nombreux systèmes. Il est gratuit, facile à apprendre et amusant à utiliser.

Cette section vous montrera comment configurer Python pour l'utiliser avec Formula AllCode. Il est supposé que vous avez une connaissance de base de Python.

même. Si ce n'est pas le cas, il existe de nombreuses ressources de qualité sur Internet si vous souhaitez apprendre cette langue.

Mise en place

La première chose à faire est de vous assurer que Python est installé sur votre ordinateur. Il est généralement installé par défaut sur un Raspberry Pi, mais pour Windows et d'autres appareils, vous devrez probablement le télécharger et l'installer à partir de <http://www.python.org>.

Il existe deux versions de Python, 2 et 3, et chacune peut être utilisée pour contrôler le robot Formula AllCode, mais il est préférable de s'assurer que la dernière version est installée.

En plus de Python lui-même, vous devrez également installer la bibliothèque PySerial. Celle-ci peut être trouvée sur GitHub : <https://github.com/pyserial/pyserial> ou peut être téléchargé sur un appareil basé sur Linux en utilisant la commande suivante dans une fenêtre de terminal :

```
sudo apt-get install python-serial
```

Maintenant que Python et la bibliothèque PySerial sont installés, vous devez télécharger la bibliothèque Formula AllCode Python à partir d'ici : <https://www.matrixsl.com/allcode/resources/>

Vous trouverez sur cette page des exemples et d'autres ressources qui vous aideront à contrôler le robot en Python et dans de nombreux autres langages.

Mon premier programme Python

```
importation FA          # Importer la bibliothèque Formula
fa = FA.Create()        # AllCode Créer une instance de l'API
fa.ComOpen(5)           # Ouvrir le port COM
fa.Forwards(100)        # Avancer de 100mm #
fa.ComClose()           # Fermer le port COM
```


Contrôler le robot

Utilisation de Python

Ce programme très simple fait avancer le robot de 10 cm à l'aide de la commande API `Forwards`, mais il y a un certain nombre d'autres lignes de code avant et après cette commande qui peuvent nécessiter plus d'explications.

Les trois premières lignes de code importent la bibliothèque afin que vous puissiez utiliser les commandes de l'API, puis une instance de l'API est créée et un canal de communication avec elle est ouvert. Le nombre "5" représente le port COM qui a été créé lors de l'appairage du robot.

Il est important de fermer le port COM et de le faire à la fin du programme. à l'aide d'un appel à la commande API `ComClose`.

Contrôler plusieurs robots

En créant plusieurs instances de l'API, nous pouvons en fait en contrôler plusieurs en même temps. Le programme ci-contre montre comment cela est possible.

Tout comme le premier programme, nous commençons par importer la bibliothèque FA (notez que nous importons également la bibliothèque "time"). Nous créons ensuite deux instances de l'API et ouvrons leurs ports COM.

La routine pour dessiner le carré devrait être explicite.

Enfin, les deux ports COM sont fermés.

Théoriquement, de nombreux robots peuvent être contrôlés simultanément. Malheureusement, il y a une limite pratique due à la capacité du dispositif Bluetooth de l'ordinateur. J'ai trouvé que 3 ou 4 était le maximum réaliste.

Aller plus loin

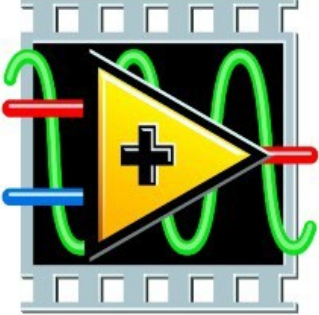
Nous n'avons montré que quelques brefs exemples de la manière de contrôler le robot Formula AllCode à l'aide de Python. Si vous consultez la référence de l'API à la fin de ce document, vous trouverez

```
# Importer les bibliothèques import FA
temps d'importation

#Créer et ouvrir 2 robots
fa1 = FA.Create()
fa2 = FA.Create()
fa1.ComOpen(5)
fa2.ComOpen(6)

#Tracer un carré
boucle = 0
while loop > 0 :
    fa1.En avant(100)
    fa2.En avant(100)
    time.sleep(1)
    fa1.A droite(90)
    fa2.A droite(90)
    time.sleep(1)
    loop = loop - 1

# Fermer les ports COM
fa1.ComClose()
fa2.ComClose()
```



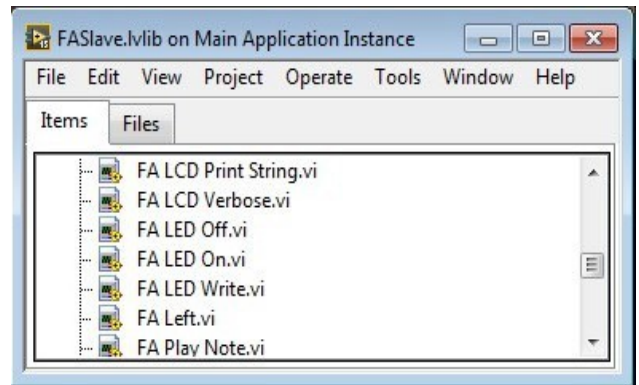
LabVIEW est un environnement de développement permettant de créer des applications personnalisées qui interagissent avec des données ou des signaux du monde réel dans des domaines tels que la science et l'ingénierie.

Il peut également être utilisé pour contrôler le robot Formula AllCode. Cette section explique comment démarrer

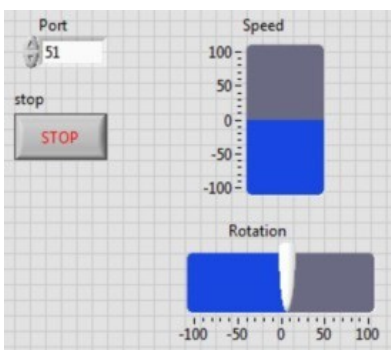
L'utilisation du robot avec Labview est assez simple et consiste à utiliser une bibliothèque fournie par MatrixTSL.

Tout d'abord, téléchargez la bibliothèque (qui se compose d'une DLL et d'un fichier de bibliothèque LabView) à partir du site Web de Matrix TSL : <https://www.matrixtsl.com/allcode/resources/>

Pour commencer, créez un nouveau VI vierge, puis ouvrez l'application "FASlave.lvlib" qui a été téléchargé précédemment. Ce fichier contient tous les appels de fonction à l'API Formula AllCode, comme indiqué à droite.



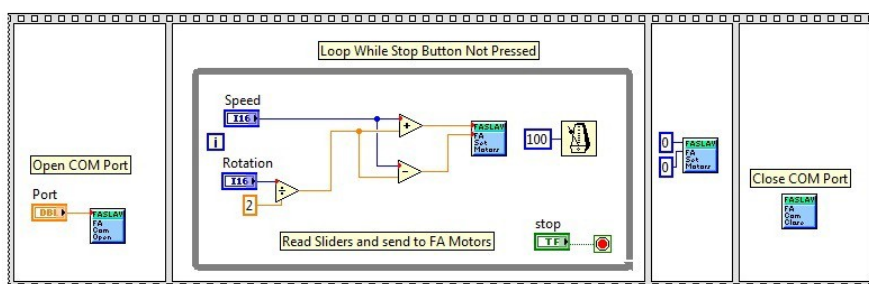
Un exemple de programme est présenté ci-dessous, utilisant deux curseurs pour contrôler le robot. Réglez le numéro de la case "Port" sur le numéro de port COM créé pour votre robot lors de son appairage.



Le programme en bas de page montre une séquence plate qui garantit que les différentes parties du programme sont appelées à tour de rôle.

La fenêtre de gauche s'exécute en premier et ouvre le port COM. La fenêtre du milieu tourne en boucle jusqu'à ce que l'on appuie sur "stop". Elle prend la valeur des curseurs et l'envoie à la commande SetMotors de l'API chaque fois que 100ms.

La fenêtre suivante arrête les moteurs lorsque la boucle est terminée et la dernière fenêtre à droite ferme le port COM.



Contrôler le robot

Utiliser Scratch



Scratch est un outil de programmation très populaire qui permet d'initier les enfants d'âge scolaire aux techniques de codage.

Bien qu'il soit principalement utilisé pour créer des programmes qui contrôlent des personnages à l'écran, vous pouvez également l'utiliser pour contrôler du matériel externe comme le robot Formula AllCode.

Cette section décrit ce dont vous avez besoin pour commencer.

À l'heure où nous écrivons ces lignes, il existe de nombreuses versions et dérivés de Scratch - en ligne et hors ligne. Les instructions données ici concernent l'éditeur en ligne Scratch 2.

Scratch ne supporte pas nativement le matériel externe, nous avons donc écrit une application externe "helper" qui traduit la communication entre Scratch et le robot Formula AllCode. Il y a également un fichier modèle pour vous aider à commencer à utiliser le robot. Vous pouvez télécharger cette application d'aide et le fichier modèle ici :

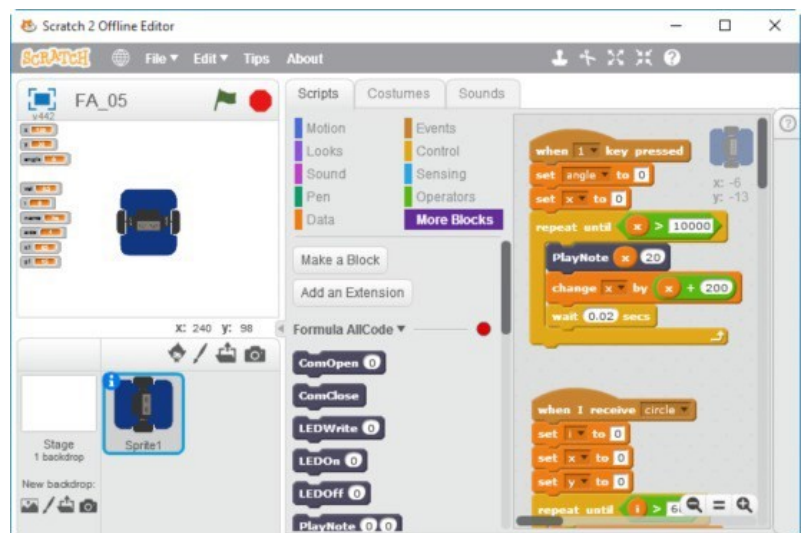
<https://www.matrixsl.com/allcode/resources/>

Avant de démarrer Scratch, lancez l'application d'aide et sélectionnez le robot que vous souhaitez connecter. à partir de la liste.

Vous pouvez maintenant ouvrir Scratch. Ouvrez d'abord le fichier modèle pour commencer votre nouveau projet. Cela chargera toutes les API dans la zone "More Blocks" de Scratch. Il vous fournira également un sprite Formula AllCode et un programme d'exemple de base.

La création de vos propres programmes devrait maintenant être simple.

Par exemple, pour déplacer le robot lorsque l'on appuie sur les touches fléchées, les icônes ci-dessous peuvent à utiliser.



Contrôler le robot

Utiliser Scratch

Il existe de nombreux autres programmes que vous pouvez écrire en Scratch. Par exemple, voici trois morceaux de code :

Le premier produit un son intéressant, le deuxième fait une petite danse et le troisième joue un air connu !

Bien entendu, vous pouvez également établir des liens avec des activités à l'écran, comme indiqué à droite.

Ici, l'ordinateur vous demande votre nom et l'affiche sur l'écran. l'écran LCD du robot Formula AllCode.

Ensuite, il fait clignoter le rétroéclairage de l'écran LCD pour attirer votre attention.

Des programmes plus complexes peuvent également être écrits. Le programme de gauche dessine une équation trigonométrique intéressante.

Vous pouvez également lire les valeurs des capteurs en retour



Cette feuille de travail explique comment fabriquer le robot déplacer en avant et en arrière, et tourner dans un sens ou dans l'autre.

La position angulaire de l'appareil lui permet de tracer une forme géométrique, comme un carré, un triangle ou une maison.

C'est probablement la toute première chose que vous voudrez faire

Activité

Autonome

Niveau de compétence

Facile

Contrôle des moteurs

Il existe deux façons de contrôler les moteurs du robot. La première méthode utilise la commande API **SetMotors** pour définir la vitesse et le sens de rotation de chacun des deux moteurs. Cette commande particulière sera abordée en détail dans une feuille de travail ultérieure. La seconde méthode de contrôle est basée sur les mouvements du logo.

Mouvements du logo

Logo est un langage conçu il y a de nombreuses années pour enseigner la programmation informatique aux étudiants, afin qu'ils puissent contrôler des appareils tels qu'un petit robot ou une tortue à l'aide d'une série de commandes simples.

En avant et en arrière

Il existe deux commandes API pour contrôler le mouvement en ligne droite.

En avant <distance> et **en arrière <distance>**

La valeur de <distance> indique la distance en millimètres que le robot doit parcourir et peut varier de 0 à 1000. Voici un programme très simple pour le faire avancer et reculer. vers l'arrière de 100 mm.

La chose importante à noter à propos de ces deux commandes est que le robot se déplacera en ligne droite. La ligne d'alimentation en électricité, parce qu'il y a un retour d'information sur la rotation de chaque moteur, permet de contrôler et de faire varier la puissance. appliquée aux moteurs. Maintenant, faisons en sorte que le robot se déplace latéralement.

```
// Un exemple de programme écrit en pseudo-code LOOP-FOREVER
```

```
Avance 100 //avance de 100 mm DELAY 250ms
```

```
Vers l'arrière 100 //mouvement vers l'arrière 100 mm DELAY 250ms
```

```
END-LOOP
```

Gauche et droite

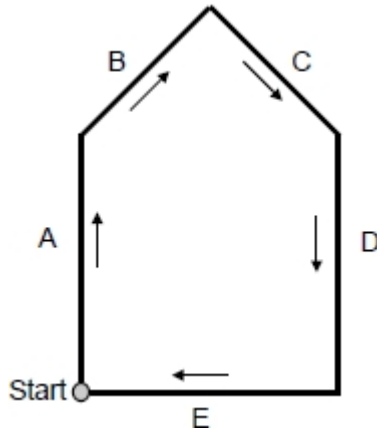
Il s'agit des deux commandes API permettant de contrôler le mouvement angulaire.

Gauche <angle> et **Droite <angle>**

La valeur de <angle> spécifie l'angle en degrés que le robot doit tourner et peut varier.

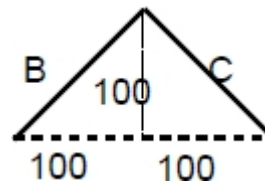
Dessiner la forme d'une maison

Soyons un peu plus aventureux et demandons au robot de tracer la façade d'une maison.



La longueur suggérée pour les côtés A, D et E est de 200 mm.

Comme le montre le schéma ci-dessous, la structure du toit est constituée de deux triangles rectangles dont la base mesure 100 mm de long et la base, 100 mm de large, perpendiculaire.



À vous de jouer

(a) En utilisant le théorème de Pythagore, vous pouvez calculer la longueur des côtés B et C. Vous devriez également être en mesure de déduire les angles.

Ensuite, en utilisant le programme ci-dessous comme guide, entrez vos valeurs calculées (là où vous voyez les points d'interrogation) et voyez si le robot dessine la forme correcte.

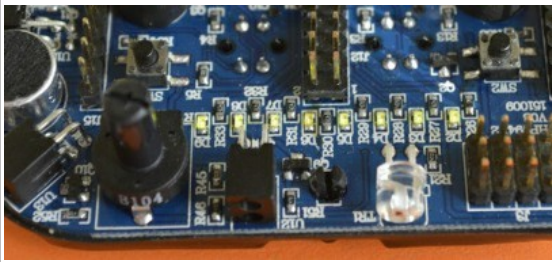
```
// Un programme d'exemple écrit en pseudo-code
Vers l'avant 200 //Tirer le côté A
D'accord ?
Vers l'avant ? ?? /Tirer l'apex B
Droit 90
Vers l'avant ? ?? /Tirer l'apex C
D'accord ?
Vers l'avant 200 //Côté tirage D
Droit 90
Vers l'avant 200 //Dessiner la base de la maison, côté E
```

(b) Essayez de faire dessiner au robot les formes géométriques suivantes : triangle équilatéral, triangle rectangle, hexagone, parallélogramme. Quelles autres formes pouvez-vous dessiner ?

(c) Défie l'un de tes amis de dessiner une forme et demande à ton robot de la copier.

Résumé

Cette section a expliqué comment utiliser les commandes API pour contrôler les moteurs du robot afin qu'ils se déplacent avec précision vers l'avant ou vers l'arrière, ou qu'ils tournent à gauche ou à droite d'un angle donné.



Cette feuille de travail explique comment utiliser les huit indicateurs verts situés sur le bord avant du robot Formula AllCode.

Ils peuvent être utilisés pour produire des effets visuels intéressants et créatifs qui vous permettront d'apprécier davantage les émissions de

Activité

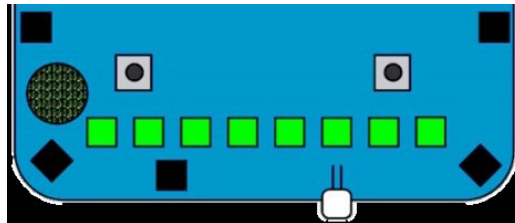
Autonome

Niveau de
compète

Facile

LED

À l'avant du robot se trouve une rangée de huit diodes électroluminescentes. Chacune porte un numéro de référence (de LED-7 à LED-0), comme le montre ci-dessous.



Les deux commandes API suivantes vous permettent d'activer ou de désactiver une DEL individuelle.

LEDOn <index> et **LEDOff <index>**

<Index> peut prendre une valeur entre 0 et 7 pour définir la LED que vous voulez manipuler. Par exemple, pour allumer la LED-7, vous devez utiliser : **LEDOn 7** et pour l'éteindre : **LEDOff 7**

Remarque :

Ces deux commandes n'affectent qu'une seule DEL.

L'état des sept autres DEL reste inchangé.

Ce programme fait clignoter la LED-7 4 fois par seconde.

// exemple de pseudo-code

```
LOOP-FOREVER
  LEDOn 7
  DELAY 125ms
  LEDOff 7
  DELAY 125ms
END-LOOP
```

Si vous souhaitez définir l'état de toutes les DEL en même temps, vous pouvez utiliser la commande API **LEDWrite <valeur>** où <valeur> peut prendre une valeur comprise entre 0 et 255.

L'ensemble des DEL peut être représenté par un nombre binaire de 8 bits (c'est-à-dire une valeur comprise entre 0 et 255). La DEL-0 est le bit le moins significatif et a une valeur de 1, tandis que la DEL-7 est le bit le plus significatif et a une valeur de 128.

Ainsi, pour régler l'état de toutes les LED en même temps, il suffit de calculer la valeur globale et d'utiliser **LEDWrite <valeur globale>**.

Feuille de travail 2

Attention, ces lumières sont animées

Par exemple, pour allumer uniquement les DEL-2, DEL-1 et DEL-0, vous devez envoyer la valeur 7 (c'est-à-dire $4 + 2 + 1 = 7$) en tant que paramètre. Pour ce faire, vous devez utiliser la commande API suivante.

`LEDWrite 7`

À vous de jouer

Vous avez peut-être entendu parler de *Knight Rider*, une série télévisée américaine qui mettait en scène une voiture dont les phares étaient utilisés pour créer une séquence animée. Dans la séquence la plus connue, les phares se déplaçaient vers le centre, puis revenaient vers l'extérieur. Vous devriez être en mesure de créer le motif *Knight Rider* en utilisant les huit DEL du robot. La première étape consiste à déterminer les valeurs numériques qui doivent être écrites sur les DEL. Dans le premier cas, les DEL-7 et les DEL-0 sont allumées en même temps. Vous devez déterminer la valeur que cela représente. Dans le schéma suivant, les DEL-6 et les DEL-1 sont allumées. Là encore, vous devez déterminer sa valeur. Continuez ainsi jusqu'à ce que vous ayez calculé toutes les valeurs, puis entrez ces valeurs dans votre programme sous forme de commandes API. Vous constaterez probablement qu'il est nécessaire d'ajouter un délai entre les commandes API pour assurer une transition en douceur et obtenir l'effet *Knight Rider* souhaité.

Autres effets à tester

Il existe de nombreux autres effets que vous pouvez essayer. Voici deux suggestions :

Le motif du *serpent* - il commence par l'allumage de la première DEL (valeur 1), puis se déplace d'une place vers la gauche, de sorte que la deuxième DEL s'allume (valeur 2). Cette séquence se poursuit jusqu'à ce que la DEL la plus importante soit allumée (valeur 128). Ensuite, la séquence revient sur ses pas jusqu'au début. L'effet obtenu doit ressembler à un serpent qui déplace ou tisse sa tête ou son corps d'un côté à l'autre.

Un autre effet que vous pouvez essayer est le motif *snake*. Ce motif commence comme l'effet *serpent*, mais lorsqu'il atteint le chiffre le plus élevé (valeur 128), la DEL reste allumée pendant que le motif revient au début. L'effet est celui d'un "œuf" pondu à l'extrémité. La séquence se répète ensuite jusqu'à ce que la DEL 6 soit atteinte. Ce voyant reste allumé (comme ainsi que LED-7) alors que le modèle revient sur ses pas.

Certains modèles que vous souhaitez créer peuvent suivre une séquence mathématique, ce qui signifie que vous pouvez sortir une valeur d'une variable nommée, comme indiqué ci-dessous.

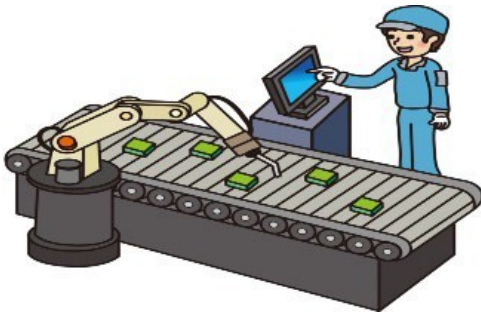
`LEDWrite <variable_nommée>`

Résumé

Cette section vous a montré comment utiliser les commandes de l'API pour manipuler la rangée de DEL sur le robot en envoyant une valeur pour contrôler les huit DEL ensemble ou individuellement.

Feuille de travail 3

L'allumer, l'éteindre



Cette feuille de travail explique comment les boutons-poussoirs embarqués et les DEL afin que vous puissiez comprendre comment les entrées et les sorties fonctionnent dans un système numérique simple.

Activité

Autonome

Niveau de compétence

Facile

Boutons poussoirs

Vers l'avant du robot, à côté des fixations des moteurs, se trouvent deux boutons-poussoirs étiquetés SW1 et SW2 qui peuvent être détectés par votre programme et utilisés pour déclencher une tâche.

Il y a également un autre bouton-poussoir étiqueté SW3, à côté du connecteur micro-USB, qui exécute une fonction de réinitialisation chaque fois qu'il est enfoncé.

La commande API pour lire l'état de SW2 ou SW1 est la suivante :

ReadSwitch <index>

L'appel API ReadSwitch utilise un seul paramètre "index" (0 pour le bouton gauche, 1 pour le bouton droit) et renvoie une valeur de 1 (pressé) ou 0 (non pressé).

La façon dont cet appel serait utilisé est d'assigner le résultat à une variable nommée (par exemple SW_1).

SW_1 = ReadSwitch 0

Si vous souhaitez simplement effectuer une action ou une tâche simple lorsque vous appuyez sur un bouton, il vous suffit de tester si la valeur de la variable nommée est supérieure à zéro.

```
// Un exemple écrit en pseudo-code
LOOP-FOREVER
  SW_1 = ReadSwitch 0
  IF SW_1 > 0 THEN // Détecter si SW1 est
    enfoncé CALL SW_1_TASK // Routine utilisateur
  END-IF
  SW_2 = ReadSwitch 1
  IF SW_2 > 0 THEN // Détecter si SW2 est pressé
    CALL SW_2_TASK // Routine utilisateur
  END-IF
END-LOOP
```

Feuille de travail 3

L'allumer, l'éteindre

À vous de jouer

Maintenant que vous savez comment vérifier les boutons SW1 et SW2, vous devriez essayer d'écrire un programme pour effectuer une série de tâches l'une après l'autre.

(a) Lorsque le bouton SW1 est enfoncé, le motif Knight Rider est envoyé aux DEL du robot, et lorsque le bouton SW_2 est enfoncé, le motif Snake est envoyé.

Cela signifie que vous devrez écrire du code à l'intérieur des routines utilisateur intitulées *SW_1_Task* et *SW_2_Task* pour produire les motifs souhaités.

Pour simplifier les choses à ce stade, vous constaterez que votre programme exécutera la séquence ci-dessus une seule fois, à moins que le bouton approprié ne soit encore enfoncé.

Lorsque vous relâchez enfin le bouton, le motif en cours s'arrête à la fin de sa séquence.

Dans une feuille de travail ultérieure, vous découvrirez comment écrire un programme qui se souvient du bouton sur lequel vous avez appuyé, de sorte que vous n'avez pas besoin de garder votre doigt sur le bouton.

(b) Le bouton gauche permet au robot de dessiner un carré et le bouton droit de dessiner un cercle.

(c) Jetez un coup d'œil aux commandes de l'API dans l'annexe et voyez si vous pouvez faire en sorte que d'autres choses se produisent lorsque vous appuyez sur un bouton.

Résumé

Cette section vous a montré comment lire l'état des boutons SW1 et SW2 et effectuer une certaine tâche.

Une autre utilisation possible de SW1 ou SW2 est de servir de déclencheur, de sorte que lorsqu'un bouton est pressé, il envoie un signal à un autre programme lui demandant de commencer à fonctionner.

Par exemple, vous pouvez faire en sorte que le son (précédemment enregistré sur la carte micro SD) soit et le robot exécute une danse lorsqu'on appuie sur un bouton.

Bonjour le monde



Cette feuille de travail explique comment envoyer un texte sur l'écran LCD situé dans la zone centrale du robot.

La deuxième partie de cet exercice concerne l'envoi d'une série de messages multi-lignes qui changent après un certain temps.

Activité **Autonome** Niveau de compétence **Facile**

Le panneau graphique LCD (gLCD), situé au centre du robot, est capable de dessiner des éléments graphiques tels que des lignes et des rectangles, et de manipuler des pixels individuels.

Note : Ces aspects seront traités plus en détail dans une autre feuille de travail.

L'utilisation du panneau en mode non graphique permet d'afficher des messages (texte et chiffres), par exemple les valeurs numériques obtenues à partir des capteurs de lumière ou de distance, ou simplement un message simple pour envoyer un retour d'information "visuel" à l'utilisateur. Vous pouvez bien sûr mélanger les deux modes en plaçant des lignes de séparation, des blocs de forme ou des symboles graphiques créés par l'utilisateur (près de (ou texte) pour attirer l'attention de l'utilisateur sur certaines parties du panneau.

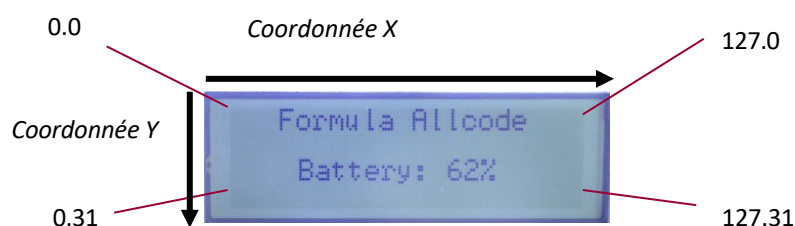
La luminosité du rétro-éclairage de l'écran peut également être modifiée par le biais d'un programme de contrôle.

Toutefois, il convient de noter que l'allumage complet du rétroéclairage entraîne un déchargement plus rapide de la batterie.

Disposition de l'écran LCD

Le diagramme ci-dessous montre comment les coordonnées X et Y du panneau sont disposées. La coordonnée X traverse le haut du panneau (de gauche à droite) et peut prendre une valeur comprise entre 0 et 127 pour définir une colonne sur l'écran. La coordonnée Y va du haut vers le bas du panneau et peut prendre une valeur comprise entre 0 et 31 pour définir une ligne sur l'écran.

Lorsque vous travaillez dans un environnement graphique, il est d'usage de spécifier les paramètres X et puis la coordonnée Y (c'est-à-dire la colonne puis la ligne), de sorte que la position en haut à gauche serait la suivante est connue sous le nom de 0,0 et la position inférieure droite sous le nom de 127,31. Une erreur fréquente consiste à confondre les



Bonjour le monde

Maintenant que vous savez comment le panneau est présenté, vous pouvez écrire un message à l'aide de l'appel API.

```
LCDPrint <x> <y> <texte>
```

Bonjour le monde

Voici un exemple du message que l'on trouve dans la plupart des livres sur la programmation.

```
// Un exemple écrit en pseudo-code  
LCDPrint 0 0 "Hello World" (Bonjour le monde)
```

Voici un exemple de message multi-lignes centré sur l'écran LCD.

```
// Un exemple écrit en pseudo-code  
  
LCDPrint 0 0 " Bonjour et bienvenue  
" LCDPrint 0 8 " à la "  
LCDPrint 0 16 " Robotique "  
LCDPrint 0 24 " Guide d'instruction "
```

Chaque ligne du message est centrée en ajoutant des espaces de part et d'autre de la chaîne de texte. Le nombre d'espaces nécessaires est calculé en soustrayant la longueur de chaque message de 21 (la largeur de caractère de l'écran LCD) et en divisant le résultat par deux. Parfois, le résultat est un nombre impair, ce qui signifie qu'un côté du message comporte un espace supplémentaire.

À vous de jouer

Transformez le programme ci-dessus en une macro ou un sous-programme en rassemblant le pseudo-code et lui donner un nom approprié comme *MESSAGE_1*. Une fois que vous avez fait cela, créez d'autres messages multilignes, leur donner des noms de macro appropriés et les combiner pour faire des annonces dans une séquence chronométrée (comme indiqué ci-dessous).

```
// Un exemple écrit en pseudo-code  
  
LOOP-FOREVER  
  LCDClear  
  APPEL MESSAGE_1  
  DELAY 5s  
  LCDClear  
  APPEL MESSAGE_2  
  DELAY 5s  
END-LOOP
```

Toutes les informations affichées à l'écran peuvent être effacées à l'aide de la commande API **LDClear** . avant d'envoyer un nouveau message à l'écran LCD.

Affichage de valeurs numériques

Si vous avez besoin d'afficher la valeur numérique d'une variable nommée, vous pouvez le faire en utilisant la commande API.

```
LCDNumber <x> <y> <numéro>
```

Bonjour le monde

Voici un exemple d'affichage de la valeur de la variable nommée "distance".

```
Numéro LCD 0 0 distance
```

Pour rendre la présentation plus conviviale, un texte pourrait être utilisé pour indiquer la valeur numérique afin que vous sachiez ce qu'elle représente.

```
// Un exemple écrit en pseudo-code
```

```
LCDClear  
LCDPrint 0 0 "Distance = "  
LCDNumber 66 0 distance
```

Luminosité du rétro-éclairage

Comme indiqué à la page précédente, la luminosité du rétro-éclairage de l'écran peut être modifiée pour s'adapter aux conditions d'observation. La commande API permettant d'effectuer cette opération est la suivante : **LCDBacklight <valeur>** où <valeur> peut varier de 0 (rétro-éclairage éteint) à 100 (pleine luminosité).

LCDOptions en mode non graphique

L'appel API suivant peut être utilisé pour contrôler la manière dont le texte est affiché sur l'écran LCD.

```
LCDOptions <premier plan> <arrière plan> <transparent>
```

Les deux premiers paramètres peuvent prendre une valeur de 0 ou de 1 pour définir respectivement le blanc ou le noir. Cela signifie que vous pouvez afficher un texte blanc sur un fond noir ou un texte noir sur un fond blanc. Le troisième paramètre peut également prendre une valeur de 0 (affichage du texte au premier plan et à l'arrière-plan) ou de 1 (affichage du texte au premier plan uniquement).

Commandes LCD associées

Les commandes API suivantes affectent l'aspect graphique de l'écran LCD et sont traitées dans une feuille de travail ultérieure.

```
LCDLine, LCDRect et LCDPixel
```

Ce qu'il faut retenir

- la disposition de l'écran LCD est identique à celle d'une feuille de papier millimétré
- l'axe X traverse la partie supérieure et l'axe Y descend sur le côté
- les axes X et Y sont numérotés à partir de zéro (et non d'un)

A vous (encore)

Cette section vous a montré comment utiliser les différentes commandes API pour afficher et formater des informations (texte et valeurs numériques) sur l'écran LCD.

(a) Demandez au robot d'afficher le texte d'une blague. Vous devrez faire une pause avant d'afficher la chute !

(b) Affichez les résultats de votre équipe de football préférée. Possibilité de faire une pause entre les résultats jusqu'à ce que

Quelle est la luminosité de cette lumière ?



Cette feuille de travail explique comment utiliser le capteur de lumière situé à l'avant du robot.

Le premier exercice mesure le niveau de lumière et l'affiche sur l'écran LCD.

Le deuxième exercice consiste à ajuster l'éclairage des LED en fonction de la luminosité ambiante.

Activité

Lié

Niveau de
compétence

Facile

Au centre du bord avant du robot se trouve un capteur de lumière. Bien qu'il ressemble à une simple LED, il s'agit en fait d'un photo-transistor qui réagit à la quantité de lumière qui lui parvient. La quantité de courant qui peut passer à travers le dispositif augmente en fonction de la luminosité.

Ainsi, en plaçant le dispositif dans un simple circuit de diviseur de potentiel, une tension variera en fonction de l'intensité lumineuse. Le robot peut détecter cette tension et la convertir en valeur.

La commande API permettant de lire le niveau de luminosité est : **ReadLight** et la façon de l'utiliser est d'affecter le résultat à une variable nommée (par exemple, illumination).

```
illumination = ReadLight
```

La première chose à faire est de déterminer la plage des valeurs numériques produites par le photo-transistor dans différentes conditions d'éclairage. Cet exercice simple montre comment mesurer le niveau de lumière et afficher la valeur sur l'écran LCD.

```
// Un exemple écrit en pseudo-code
LDClear
LCDPrint 0 0 "Valeur lumineuse = "

LOOP-FOREVER
    illumination = ReadLight
    LCDNumber 84 0 illumination
    DELAY 100ms
END-LOOP
```

Bien que le photo-transistor puisse théoriquement produire des valeurs comprises entre 0 et 4095, il se peut que vous ne puissiez pas les obtenir si vous ne placez pas le robot dans une enceinte totalement noire ou si vous ne le tenez pas à côté d'une ampoule électrique brillante.

Notez certains des résultats que vous obtenez lorsque vous placez le robot dans différentes parties d'une pièce ou lorsque vous l'emmenez à l'extérieur. Notez s'il s'agit d'un jour ensoleillé ou couvert. Vous pouvez également essayer d'orienter une lampe de poche vers le capteur de lumière et voir quelles lectures vous obtenez en vous rapprochant ou en vous éloignant du robot.

L'exercice suivant est un système de contrôle ou de rétroaction très simple. L'objectif est de mesurer le niveau de lumière ambiante et d'utiliser cette valeur pour ajuster le nombre de DEL allumées à l'avant du robot. L'idée est d'allumer davantage de DEL lorsque le niveau de lumière diminue et d'en éteindre davantage lorsque la lumière augmente.

Si cette idée était appliquée à une situation pratique (en utilisant des lampes plus puissantes), il devrait être possible de maintenir l'éclairage d'une pièce à un niveau constant, quels que soient les niveaux de luminosité. Le pseudo-code ci-dessous montre comment cela est possible.

```
// Un exemple écrit en pseudo-code

LOOP-FOREVER
    illumination = ReadLight
    illumination = Round(illumination / 16)
    LEDs = 255 - illumination
    LEDWrite = LEDs
    DELAY 100ms
END-LOOP
```

Le programme commence par lire la valeur du capteur de lumière dans la variable nommée *'illumination'* et divise ensuite cette valeur par 16. Cela permet d'ajuster la valeur de l'éclairage pour qu'elle se situe dans la fourchette suivante la plage de 0 à 255 au lieu de 0 à 4091.

Pour s'assurer que le résultat est un entier, nous utilisons la fonction Round(). Le résultat que vous avez choisi Le langage de programmation dispose de cette fonction, mais elle peut s'appeler autrement et nécessiter un paramètre supplémentaire pour spécifier le niveau d'arrondi.

La partie suivante du programme crée l'effet d'éclairage inversement proportionnel en soustrayant la valeur d'éclairage de 255. Ainsi, lorsque la valeur de l'éclairage augmente, la valeur de la variable nommée "LEDs" diminue.

La dernière partie du programme écrit la valeur de la variable nommée "LEDs" sur les LEDs à l'avant du robot, puis, après un court délai, toute la séquence est répétée.


À vous de jouer

Cette section vous a montré comment utiliser l'appel API pour lire la valeur du capteur de lumière. Essayez les deux exercices décrits ici pour acquérir de l'expérience dans l'utilisation du capteur de lumière et pour renforcer vos connaissances sur le fonctionnement de l'écran LCD et des DEL.

Le deuxième exercice ne fonctionnera pas comme prévu - si la valeur de la variable "LEDs" est de 127, 7 LEDs s'allumeront, mais si elle est de 128, une seule s'allumera. Essayez de modifier le programme pour que le nombre de DEL qui s'allument augmente lorsque le niveau de luminosité baisse.

Fiche de travail 6

A la recherche de la lumière

	<p>Cette feuille de travail rassemble le travail effectué au cours d'une session de formation. Quelques exercices précédents pour résoudre un problème de la vie réelle. Le problème à résoudre consiste à faire en sorte que le robot suive une source de lumière provenant d'une torche tenue à la main. Cela semble plus simple que vous ne le pensez !</p>
Activité	Lié
Niveau de compétence	Intermédiaire

Stratégie

La première chose à faire, pour résoudre le problème décrit ci-dessus, est d'élaborer une stratégie pour faire face aux situations possibles qui peuvent se produire. Par exemple, si le robot détecte le faisceau lumineux de la torche, il doit avancer d'une certaine distance et vérifier à nouveau. C'est assez évident, mais que doit faire le robot s'il perd de vue le faisceau ? Il doit s'arrêter et essayer de trouver la source de lumière.

Une façon de déterminer ce que le robot doit faire est de se mettre dans la même situation et de penser à ce que l'on ferait. Avec vos yeux, vous pourriez probablement voir le faisceau lumineux. Il se peut que vous deviez déplacer légèrement vos yeux vers la gauche ou la droite pour "voir" à nouveau le faisceau. Si le faisceau a changé radicalement de position, vous devrez peut-être bouger la tête à gauche ou à droite parce que la distance que peuvent parcourir les yeux est physiquement limitée.

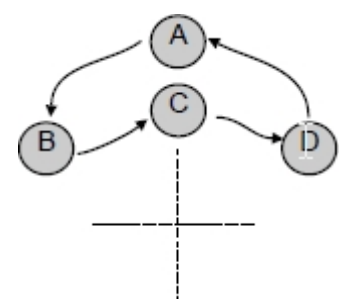
L'anatomie du robot est un peu différente de la nôtre. Il n'a peut-être qu'un seul œil (c'est-à-dire un capteur de lumière) dans une position fixe (sur le bord avant du robot), mais il est capable de bouger sa tête sur une rotation complète de 360 degrés. Une chose que nous ne pouvons pas faire !

Voici la stratégie à adopter pour tenter de trouver le faisceau lumineux. Actuellement, le robot est immobile, il doit donc tourner vers la gauche d'un certain angle, disons 5 degrés, et vérifier s'il y a un faisceau. S'il parvient à trouver le faisceau, il commence à avancer. S'il n'y parvient pas, il déplace sa tête vers la droite du même angle et vérifie à nouveau. Si ces deux manœuvres ont échoué, l'angle est augmenté, par exemple jusqu'à 10 degrés, et la procédure est répétée.

Voici un diagramme de séquence qui montre de manière imagée comment la stratégie pourrait être mise en œuvre.

Le point A est le robot pointant droit devant lui. La flèche de A à B indique un mouvement angulaire vers la gauche. Le point B vers C représente le robot qui revient vers l'avant. La flèche de C à D indique un mouvement angulaire vers la droite, un mouvement angulaire vers la droite, tandis que D à A est le mouvement de retour vers l'avant, le point avant.

L'existence du faisceau lumineux est vérifiée aux quatre points. La transition de D à A est l'endroit où l'angle de détection est augmenté avant que la séquence ne soit répétée.



Feuille de travail 6

À la recherche de la lumière

Voici un exemple de la structure du programme de recherche de lumière. Pour vous aider à lire le programme, les commandes de l'API ont été mises en évidence en **bleu**.

```
// Un programme d'exemple écrit en pseudo-code

seuil = 2000 //Valeur expérimentale
pos = "A"
mouvement = 5 //Initiale 5 degrés
LOOP-FOREVER
  illumination = ReadLight
  SI l'éclairement > seuil ALORS
    pos = "A"
    mouvement = 5
    Avances 100
  ELSE
    SI pos == "A" ALORS
      pos = "B"
      Mouvement de gauche
    ELSE-IF pos == "B" ALORS
      pos = "C"
      Mouvement de droite
    ELSE-IF pos == "C" ALORS
      pos = "D"
      Mouvement de droite
    ELSE //doit être au point D
      pos = "A"
      Mouvement de gauche
      SI mouvement < 180 ALORS
        mouvement = mouvement + 5
      ELSE
        mouvement =
        5 END-IF
    END-IF
  END-IF
  DELAY 50ms //Délai court (voir
texte) END-LOOP
```

Bien que la structure corresponde étroitement à la description textuelle donnée à la page précédente, il y a quelques points à noter. La valeur seuil a été obtenue par en expérimentant avec une petite lampe de poche dans une pièce à l'éclairage tamisé. Vous devez viser une situation où la quantité de lumière disponible plus la lumière de la torche ne dépasse pas la limite supérieure du capteur de lumière (par exemple 4095).

La boucle principale comprend un court délai pour atténuer la synchronisation des mouvements gauche/droite.

Vous pouvez essayer de modifier la valeur de la variable nommée *mouvement* et voir comment cela affecte le comportement du robot et sa capacité à localiser le faisceau lumineux.



Cette feuille de travail rassemble certaines tâches couvertes dans les sections précédentes, pour résoudre un problème réel.

Le problème à résoudre est le suivant : le robot doit utiliser les capteurs embarqués pour suivre une ligne noire sur un fond blanc. Cela semble plus simple que vous ne le pensez !

Activité

Lié

Niveau de
compétence

Intermédiaire

Si vous n'avez pas encore regardé sous le robot, c'est le moment de le faire. Vous devriez pouvoir repérer les deux capteurs qui détectent la quantité de lumière réfléchiée par une surface. Chaque capteur comprend un émetteur infrarouge et un phototransistor, qui peuvent identifier la différence entre les surfaces blanches et noires en fonction du contraste et des propriétés de réflexion d'un objet.

La commande API permettant de lire le capteur de ligne est la suivante : `ReadLine <index>` : **ReadLine <index>** et la façon de l'utiliser est d'affecter le résultat à une variable nommée (par exemple `line_reading`).

Le paramètre **<index>** peut prendre une valeur de 0 ou 1 pour sélectionner le capteur approprié.

Voici un exemple de lecture du *capteur de ligne 0*, qui est le capteur de gauche lorsque l'on regarde le robot depuis la partie supérieure, la face du panneau gLCD étant orientée dans la direction opposée.

```
left = ReadLine 0
```

En guise d'exercice, vous pourriez écrire un programme simple pour lire les capteurs de ligne et afficher leurs valeurs sur l'écran LCD. Cela vous permettra d'expérimenter et de voir quelles sortes de valeurs peuvent être affichées sur l'écran LCD.

```
// Un exemple écrit en pseudo-code
LCDClear
LCDPrint 0 0 " Gauche Droite "

LOOP-FOREVER
  gauche = ReadLine 0
  droite = ReadLine 1
  LCDNumber 24 0 gauche
  LCDNumber 72 0 droite
END-LOOP
```

Le programme ci-dessus devrait produire un écran clairement formaté avec les valeurs du capteur de ligne affichées sous les titres Gauche et Droite. Cela vous permettra d'enregistrer les valeurs de réflexion pour différents types de surfaces. Notez les relevés pour une surface noire et une surface blanche.

Les valeurs que vous devriez obtenir sont probablement 0 pour le noir et 200 pour le blanc.

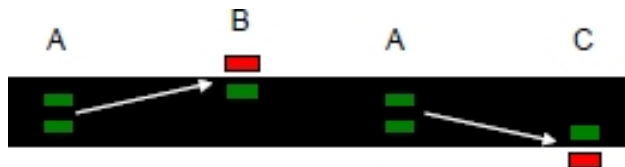
Fiche de travail 7

Suivez ma ligne

Mise en place d'une piste d'essai et résolution du problème

Si vous avez le tapis fourni avec votre robot, vous pouvez l'utiliser, ou vous pouvez fabriquer votre propre piste à l'aide d'un rouleau de ruban adhésif noir de 25 mm de large collé sur un morceau de carton blanc rigide. Une piste ovale ou en forme de huit sont de bonnes formes pour commencer. Veillez à ce que les virages ne soient pas trop prononcés ou trop serrés, sinon le robot aura du mal à les négocier.

Lorsque vous utilisez les mouvements Logo, le robot doit se déplacer en ligne droite. des facteurs tels que le dérapage ou des traces de graisse sur la surface qui pourraient entraîner le robot à s'arrêter progressivement.



Le diagramme ci-dessus montre ce qui se passe lorsque le robot dévie de sa trajectoire. Aux points B et C, les relevés des capteurs de gauche et de droite, respectivement, sont passés à indique que le robot est à cheval sur le bord de la piste.

Le diagramme ci-dessous montre le robot se déplaçant en ligne droite lorsque la direction de la voie change. Comme dans le cas précédent, cette situation peut être détectée par le changement de la direction de la voie.



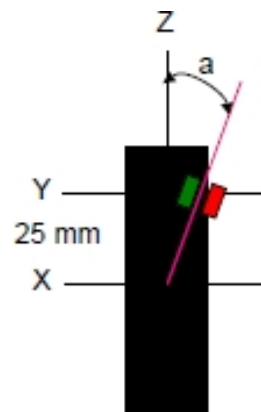
Pour remettre le robot sur la bonne voie, il doit effectuer un mouvement vers la droite au point B et vers la gauche.

L'angle selon lequel le robot doit se déplacer peut être obtenu par tâtonnement ou en appliquant quelques règles simples de géométrie et de trigonométrie, comme indiqué ci-dessous.

Le point X représente l'axe central du robot, où les deux roues sont montées. Le point Y est situé à 25 mm vers l'avant du robot et constitue le point d'appui normal des capteurs de ligne lorsque le robot est en mouvement tout droit. Le point Z est l'axe central avant-arrière du robot et le point médian de la bande noire.

La distance entre les deux capteurs est de 17 mm, ce qui convient parfaitement à la bande noire de 25 mm de large.

L'angle "a" peut être calculé par trigonométrie simple ou en réalisant un dessin à l'échelle. Dans les deux cas, vous obtiendrez un résultat d'environ 10 à 12 degrés. Il s'agit du facteur de correction angulaire



Fiche de travail 7

Suivez ma ligne

Si vous réunissez tous ces faits, vous pouvez créer un programme pour résoudre le problème.

```
// Un programme d'exemple écrit en pseudo-code
black = 0 //Valeur de réflexion sur une surface noire
white = 200 //Valeur de réflexion sur une surface blanche
correction = 10 //10 degrés
LOOP-FOREVER
  sensueur_gauche =Ligne de
  lecture          0
  sensueur_droit  =Ligne de
  lecture 1 IF sensueur_gauche
  >= blanc ALORS
    Correction à droite
  ELSE-IF sensueur_droit >= blanc ALORS
    Correction à gauche
  ELSE
    Avance de 10 //Les deux capteurs détectent une
    surface noire END-IF
END-LOOP
```

La construction IF-THEN-ELSE vérifie où se trouve le robot. S'il se trouve au-dessus de la bande noire, il doit avancer. S'il est partiellement au-dessus de la zone blanche, le facteur de correction doit être appliqué pour ramener le robot au centre.

À vous de jouer

En utilisant le programme ci-dessus comme guide, vous devez essayer de faire en sorte que votre robot suive la ligne en utilisant les mouvements du logo.

Le programme utilise des variables globales qui sont définies au début du programme.

Cela vous permettra de modifier les valeurs si les propriétés de réflexion de votre surface sont différentes.

Vous devriez également essayer de modifier la valeur de correction pour voir comment cela affecte la précision du suivi. Vous pouvez ensuite explorer l'autre façon de contrôler le robot en utilisant l'appel API **SetMotors**. Cet appel actionne continuellement les deux moteurs de la manière suivante.

SetMotors <gauche> <droite>

Les paramètres <gauche> et <droite> peuvent prendre une valeur comprise entre -100 et 100, où -100 correspond à la vitesse de rotation maximale dans le sens inverse des aiguilles d'une montre, et 100 à la vitesse maximale.

dans le sens des aiguilles d'une montre. Par exemple, la commande API **SetMotors 50 50** permet de piloter le robot en avant à la moitié de la vitesse, et **SetMotors -50 -50** le ferait reculer, toujours à la moitié de la vitesse, et **SetMotors -50 -50** le ferait reculer, toujours à la moitié de la vitesse.

Modifiez l'exemple de programme ci-dessus pour le faire fonctionner à l'aide de la commande SetMotors. Le réglage de la vitesse des moteurs est en quelque sorte un compromis entre le fait de se déplacer sur la voie à un bon rythme et le fait de ne pas sortir trop loin de la voie avant que les capteurs de ligne ne détectent et ne rectifient la situation.

En utilisant les programmes ci-dessus comme point de départ, faites quelques expériences pour voir à quelle vitesse vous pouvez faire faire un tour de piste à votre robot. Défiiez vos amis pour savoir qui a le robot le plus rapide.

Feuille de travail 8

Les boutons-poussoirs font le travail



Cette feuille de travail explique comment les boutons-poussoirs embarqués peuvent être utilisés pour exécuter une tâche spécifique.

La feuille de travail comprend un exemple de création d'un compteur binaire Up/Down piloté par les boutons-poussoirs.

Activité

Lié

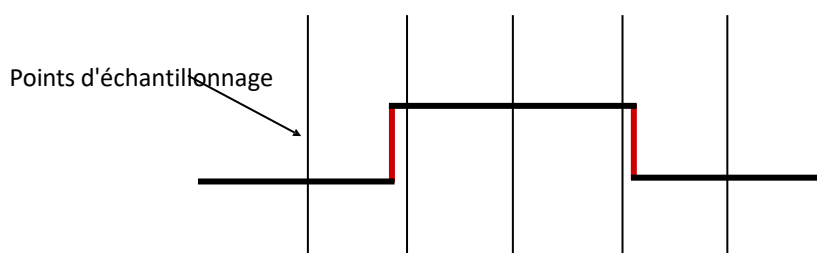
Niveau de compétence

Intermédiaire

Une feuille de travail précédente expliquait comment utiliser l'appel API **Read <switch>** pour détecter l'état de SW1 ou SW2 et exécuter une tâche si un bouton était pressé. Le programme étant très simple, il présentait un certain nombre de limitations. La principale étant qu'il fallait garder le doigt sur un bouton si l'on voulait que la tâche ou la séquence se répète continuellement. Et si vous vouliez qu'une tâche soit exécutée une seule fois, quelle que soit la durée de la pression sur le bouton ?

C'est comme appuyer sur une touche d'un téléphone portable ou d'une calculatrice. Si vous maintenez cette touche enfoncée, vous ne voulez qu'elle soit détectée qu'une seule fois, et non de manière répétée, sinon vous obtiendrez une série de chiffres répétés - comme une action de mitrailleuse.

Ce qu'il faut faire, c'est détecter le point réel où un bouton est enfoncé ou relâché, plutôt que le fait qu'un bouton est enfoncé. Ces points sont colorés en rouge sur le



Bien qu'il existe de nombreuses façons de détecter ces bords, la méthode décrite ici est basée sur la prise d'une série d'échantillons ou d'instantanés dans le temps. Chaque échantillon est comparé au précédent. S'ils sont identiques, c'est que le bouton n'a pas bougé, et s'ils sont différents, c'est que le bouton n'a pas bougé.

```
// Un exemple écrit en pseudo-code pour détecter le front montant de SW1
SW1_previous = 0

LOOP-FOREVER
  SW1 = ReadSwitch 0
  IF SW1 > SW1_previous THEN// Détection du front montant
    CALL specific_task //SW1 tâche ascendante
  END-IF
  SW1_previous = SW1
END-LOOP
```

Feuille de travail 8

Les boutons-poussoirs font le travail

À vous de jouer

Vous disposez maintenant d'un programme simple pour détecter le bord ascendant lorsque vous appuyez sur SW1. Modifiez le programme pour que le robot tourne à droite de 90 degrés à chaque fois que vous appuyez sur SW1.

Une fois cette tâche accomplie, étendez le programme de manière à ce que le robot tourne de 90 degrés vers la gauche chaque fois que vous appuyez sur le bouton SW2.

Si vous pensez aux produits/équipements que vous utilisez et qui sont dotés de boutons, d'un pavé numérique ou d'un clavier, vous constaterez que la plupart d'entre eux effectuent une action lorsque vous appuyez sur un bouton. Et si vous vouliez qu'un dispositif fasse quelque chose lorsque vous relâchez un bouton ?

La solution est très simple. Tout ce que vous avez à faire est de modifier votre programme d'origine pour détecter les

```
// Un exemple écrit en pseudo-code pour détecter le front descendant de
SW1 SW1_previous = 0

LOOP-FOREVER
  SW1 = ReadSwitch 0
  IF SW1 < SW1_previous THEN // Détection du front de chute

    CALL specific_task      //Tâche de front de chute SW1

  END-IF
  SW1_previous =SW1        // Copier le nouvel échantillon sur le précédent
END-LOOP
```

Vous savez maintenant comment écrire un programme pour détecter le moment où un bouton est enfoncé et celui où il est relâché. En fonction de votre objectif, vous pouvez vouloir détecter les deux fronts ou un seul. La situation la plus courante consiste à détecter le bord ascendant.

Un autre défi pour vous

Ecrivez un programme qui utilise les LEDs sur le bord avant du robot pour se comporter comme un compteur binaire Up/Down. Le programme est contrôlé par SW1 et SW2. Chaque fois que l'on appuie sur SW1, le compteur binaire est incrémenté, et lorsque l'on appuie sur SW2, il est décrémenté.

Vous pouvez décider, puisqu'il s'agit de votre programme, quel bord sera le bord actif. Voici quelques éléments dont vous aurez besoin dans votre programme :

Une variable nommée (par exemple, compteur) qui sera utilisée pour contenir la valeur du compteur binaire. Un bloc IF pour détecter le moment où SW1 est pressé.

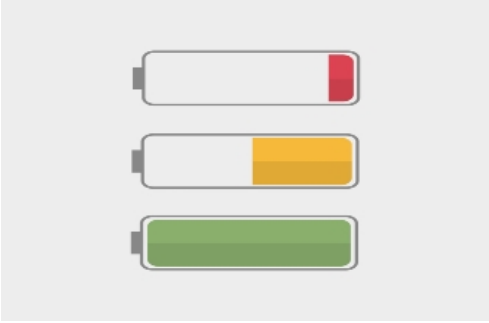
Un bloc IF pour gérer la détection de l'appui sur le bouton SW2.

La commande API pour écrire la valeur du compteur sur les LEDs du robot.

Après avoir lu certaines des autres fiches de travail de ce guide pédagogique, vous pourriez

Feuille de travail 9

Panneau d'état



Cette feuille de travail regroupe un certain nombre de tâches couverts dans les sections précédentes du guide.

L'objectif est de détecter l'état des capteurs du robot et d'afficher les valeurs sur le panneau graphique LCD (gLCD) du robot ou sur un téléphone portable.

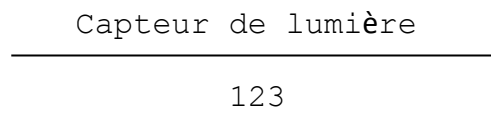
Activité **Lié** Niveau de compétence **Intermédiaire**

Stratégie de conception

Le gLCD étant assez compact, il sera difficile d'afficher beaucoup d'informations en même temps. Une solution consiste donc à afficher les informations relatives à un type de capteur particulier pendant quelques secondes, puis à passer au capteur suivant. Cela signifie que vous pouvez planifier chaque disposition, puis les combiner de manière à les afficher de façon cyclique. À la fin de cette feuille de travail, vous trouverez des instructions sur la manière de rendre le processus piloté par l'utilisateur plutôt que libre.

Planification de votre première mise en page

Le premier panneau d'état à traiter est l'affichage de la valeur du capteur de lumière. Voici une esquisse de comment cela pourrait être présenté en utilisant un mélange de lignes graphiques, et une valeur numérique.



La première ligne est un simple en-tête de texte statique composé de 5 lettres, d'un espace suivi de 6 lettres, soit un total de 12 caractères. Cela équivaut à 72 pixels (c'est-à-dire $12 * 6$ pixels). Pour centrer l'en-tête, il faut soustraire 72 de 128 et diviser le résultat par 2 (soit 28 pixels). Cela signifie que l'en-tête commencera à la coordonnée X,Y 28,0. Voici l'appel API qui permet d'effectuer cette opération.

LCDPrint 28 0 "Capteur de lumière"

Il faut ensuite une ligne de séparation qui s'étende sur toute la longueur du panneau. Plutôt que de la placer directement sous le texte, il est conseillé de laisser une rangée de pixels vides pour que le texte ressorte. Voici la commande API permettant de placer la ligne sur la ligne 9 (la dixième ligne).

La valeur obtenue par le capteur de lumière est affichée sur la ligne suivante. La valeur pouvant aller jusqu'à 4095 (soit 4 chiffres), cela signifie que la largeur nécessaire à l'affichage du nombre est de 24 pixels.

Une fois encore, vous pouvez centrer ce nombre sur l'écran en soustrayant 24 de 128 et en divisant le résultat par 2 (soit 52 pixels). Voici la commande API permettant d'afficher la valeur d'une variable nommée.

LCDNumber 52 11 <variable_nommée>

Feuille de travail 9

Panneau d'état

À vous de jouer

La première chose à faire est de créer ce programme simple pour l'écran d'état du capteur de lumière.

```
// Un exemple écrit en pseudo-code
// Panneau d'état - Capteur de lumière
LCDClear
LCDPrint 28 0 "Light Sensor"
light_level = ReadLight
LCDLine 0 9 127 9
Numéro LCD 52 11 light_level
LCDLine 0 19 127 19
```

Vous devriez mettre de l'ordre en plaçant le programme ci-dessus dans une macro. (Par exemple, *panel_1*) Puis modifiez le code que vous venez de créer pour *panel_1* afin de créer *panel_2* pour les deux capteurs de ligne. Continuez et créez des panneaux supplémentaires pour l'ensemble des capteurs de distance. Combinez-les ensuite avec un délai de 5 secondes pour obtenir un affichage en continu, comme vous l'avez fait avec *Knight Rider*. Le programme que vous venez de créer affichera chaque panneau d'état pendant 5 secondes, puis passera automatiquement au panneau suivant. Bien que ce programme soit acceptable, il peut être rendu beaucoup plus utile et convivial en y ajoutant quelques touches supplémentaires.

Capteur de ligne	
Gauche	Droit
10	46

Par exemple, vous pouvez utiliser les deux boutons-poussoirs (SW1 et SW2) pour avancer ou reculer dans l'affichage des informations sur les différents panneaux. Cet effet peut facilement être Cette opération est réalisée à l'aide de la technique de détection du front de montée abordée dans une feuille de travail précédente.

L'avantage de cette approche est qu'une fois qu'un panneau d'état est sélectionné, ses informations restent affichées à l'écran jusqu'à ce que l'utilisateur appuie sur un bouton-poussoir. Ceci est très utile lorsqu'un grand nombre de des informations, telles que les données des huit capteurs de distance, doivent être affichées.

Un autre moyen de naviguer parmi les panneaux est de détecter un son émis par le microphone du robot. Par exemple, si vous frappez dans vos mains, cette action peut être détectée et utilisée pour sélectionner le panneau suivant dans une séquence de rotation.

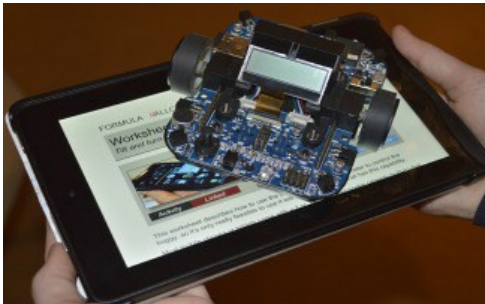
Une feuille de travail à la fin de ce guide explique comment cela peut être réalisé.

Résumé

Cette section a rassemblé un certain nombre de sujets abordés dans les feuilles de travail précédentes afin de créer un panneau d'état vraiment utile. Vous avez également découvert comment afficher des informations sur le panneau gLCD sur une base de temps fixe ou en utilisant les boutons poussoirs du robot.

Feuille de travail 10

Inclinaison et rotation (à l'aide d'un appareil mobile)



Cette feuille de travail regroupe un certain nombre de tâches couverts dans les sections précédentes du guide.

L'objectif est de développer une application qui contrôlera le robot en utilisant le capteur d'inclinaison du téléphone portable ou de la tablette.

Activité

Lié

Niveau de compétence

Intermédiaire

Cette feuille de travail décrit comment utiliser les sorties d'un accéléromètre pour contrôler le robot. Il n'est donc possible de l'utiliser qu'avec un appareil (mobile) doté de cette capacité.

La plupart des téléphones portables ou des tablettes sont équipés d'un accéléromètre à 2 ou 3 axes (déjà intégré) qui peut détecter si vous inclinez votre appareil d'un côté à l'autre ou si vous l'inclinez vers l'avant ou vers l'arrière.

Il s'agit des axes X et Y. Le troisième axe, l'axe Z, indique le mouvement lorsque l'appareil est levé ou abaissé. Ce que l'appareil mesure en réalité, c'est le taux de variation du mouvement, appelé accélération.

Ce programme utilise les valeurs d'accélération X et Y pour contrôler le robot. Par exemple, lorsque vous inclinez votre appareil mobile d'un côté à l'autre, le robot se déplace vers la droite ou la gauche. Lorsque vous l'inclinez vers l'avant ou vers l'arrière, vous contrôlez la vitesse et la direction des moteurs. L'inclinaison vers l'avant augmente la vitesse de déplacement, et l'inclinaison vers l'arrière augmente la vitesse de déplacement dans le sens inverse. Maintenir l'appareil mobile à l'horizontale arrête les moteurs.

Caractéristiques de l'accéléromètre

La sortie X de l'accéléromètre (appelée *xAccel*) donne une valeur de 0 lorsque l'appareil est au repos sur une surface plane, une valeur positive lorsqu'il est incliné vers la droite (c'est-à-dire que son côté gauche est surélevé) et une valeur négative lorsqu'il est incliné vers la gauche (c'est-à-dire que sa taille droite est surélevée). La sortie Y (appelons-la *yAccel*) donnera une valeur de 0 lorsque l'appareil est au repos sur une surface plane, une valeur positive lorsque son bas est soulevé et une valeur négative lorsque son haut est soulevé.

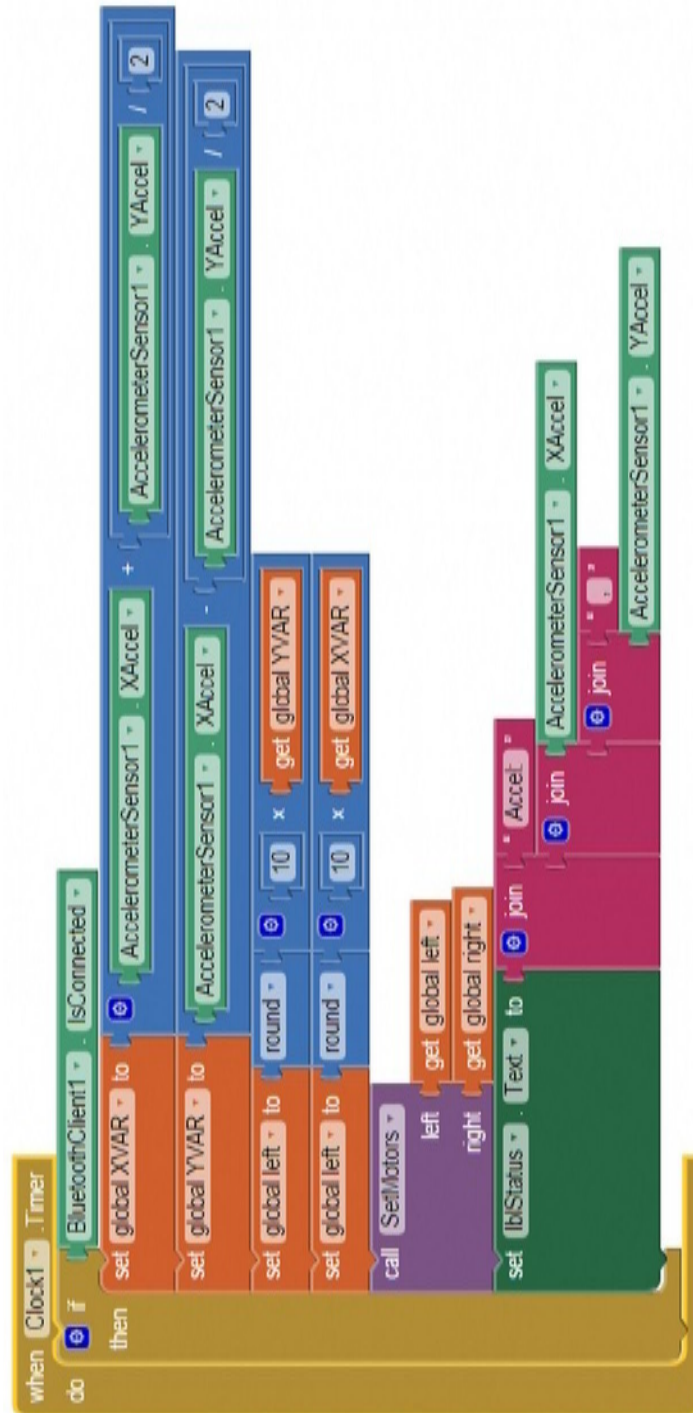
Calculs de vitesse et de direction

Une fois les valeurs d'accélération X et Y obtenues, elles doivent être converties en valeurs de vitesse et de direction pour les deux moteurs du robot. Vous pouvez faire des recherches sur Internet et étudier différentes méthodes pour effectuer ce calcul. Cependant, pour rester simple, à ce stade, la méthode utilisée ici consiste à trouver la moyenne de la somme et de la différence des valeurs X et Y.

Le programme App Inventor ci-dessous applique la technique de calcul de la moyenne et utilise les valeurs résultantes comme paramètres pour l'appel API à [SetMotors](#) `<left>` `<right>`.

Feuille de travail 10

Inclinaison et rotation (à l'aide d'un appareil mobile)



À vous de jouer

Une fois que vous aurez essayé ce programme, vous pourrez vous amuser à piloter le robot à distance à partir d'un appareil mobile. Pourquoi ne pas créer une course d'obstacles à franchir ?

Feuille de travail 11

Le gaucher peut s'orienter dans un labyrinthe



Cette feuille de travail rassemble un certain nombre de tâches couvertes par le programme les sections précédentes du guide.

L'objectif est de développer un programme permettant au robot de naviguer dans un labyrinthe simple en utilisant la main gauche. la technique du suivi de mur.

Activité

Lié

Niveau de compétence

Intermédiaire

Cet exercice est probablement l'un des plus difficiles et des plus gratifiants que vous puissiez faire avec le robot. Le faire se déplacer et naviguer dans un labyrinthe sans qu'il soit touché est un jeu qui plaît à la plupart des gens, quel que soit leur âge. Il offre également la possibilité de s'engager dans des défis compétitifs avec des amis pour voir qui a le robot le plus agile et le plus intelligent.

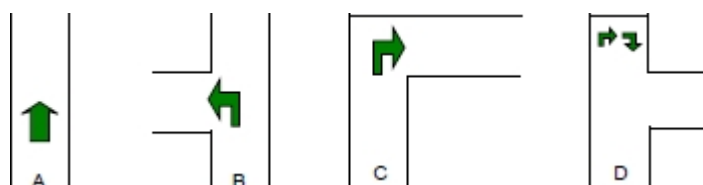
Si vous faites quelques recherches sur l'internet, vous découvrirez quelques-unes des nombreuses méthodes différentes que les gens ont développées pour s'échapper du centre d'un labyrinthe ou pour s'y déplacer. La méthode la plus courante est l'algorithme de suivi du mur de gauche. La méthode la plus courante est l'algorithme de suivi du mur de gauche.

Un algorithme est une procédure ou une formule permettant de résoudre un problème. Il peut s'écrire de manière informelle de mots ou d'énoncés, ou d'une manière structurée, comme un morceau de pseudo-code ou un code formel. langage de programmation.

Commençons par supposer que le robot a été placé à l'entrée d'un labyrinthe et voyons comment un algorithme viable pourrait être développé. À ce stade, il peut être judicieux de partager cet exercice avec un ami afin de pouvoir échanger des suggestions.

Le robot dispose d'un ensemble de capteurs de distance (placés aux quatre coins et aux quatre bords) qui peuvent être utilisés pour détecter la présence d'obstacles. Le robot doit vérifier la présence du mur sur le capteur de distance de gauche. Il doit également détecter le capteur de distance situé à l'avant du robot pour s'assurer que la voie d'accès est libre, avant de se déplacer.

avancer
comme le montre le



. Situation (A)
schéma ci-dessous.

Si le mur (du côté gauche) disparaît, indiquant un vide dans le labyrinthe, le robot doit alors tourner vers la gauche et continuer à suivre le mur. Situation (B) dans le diagramme.

Feuille de travail 11

Sortie d'un labyrinthe par la gauche

Si le robot se trouve dans une impasse, comme dans la situation (D) au verso, il doit s'arrêter et tourner de 90 degrés dans le sens des aiguilles d'une montre. Lorsque le robot réévalue la situation, il constate que sa trajectoire est toujours bloquée et il doit donc effectuer une nouvelle rotation de 90 degrés dans le sens des aiguilles d'une montre

Le robot a effectivement pivoté de 180 degrés et peut revenir sur ses pas pour sortir de l'impasse.

Pour résoudre la situation (D), il faut appliquer deux fois la situation (C).

L'algorithme est presque complet, il ne reste plus qu'à l'encapsuler dans une boucle pour que le robot répète les étapes ci-dessus de manière cyclique. Cette solution est en fait un contrôle tout comme les programmes utilisés pour contrôler les processus industriels.

Voici un exemple de la structure du programme pour la navigation dans un labyrinthe.

Pour vous aider à lire le programme, les commandes de l'API ont été mises en évidence en **bleu**.

Comme indiqué précédemment, si le robot se trouve dans une impasse, le programme appliquera la règle pour la situation (C) deux fois dans des boucles consécutives à travers le programme.

```
// Un exemple écrit en pseudo-code
// Naviguer dans un labyrinthe à l'aide des capteurs de distance DS-0 et DS-2

clear = 100 //La voie est libre ou il n'y a pas de mur
wall = 10 //Mur détecté

LOOP-FOREVER
left = ReadIR 0 //Lire le capteur de distance DS-0 (capteur
gauche) front = ReadIR 2 //Lire le capteur de distance DS-2
(capteur avant) IF left <= wall AND front == clear THEN
//Situation A
    SetMotors 50 50 //Voir note ci-dessous
ELSE-IF left == clear THEN //Situation B
    SetMotors 0 0
    Gauche 90
ELSE-IF gauche <= mur ET avant <= mur ALORS //Situation C
    SetMotors 0 0
    Droit 90
END-IF
END-LOOP
```

À vous de jouer

Créez le programme décrit ci-dessus afin d'acquérir de l'expérience dans l'utilisation des capteurs de distance pour contrôler la vitesse et la direction des deux moteurs.

Une fois que vous avez réussi à faire naviguer votre robot dans un labyrinthe, vous devez.. :

(a) Essayez de modifier les valeurs de **SetMotors** pour voir à quelle vitesse le robot peut se déplacer dans le labyrinthe sans s'écraser contre un mur.

(b) Essayez de trouver d'autres algorithmes de contrôle pour la navigation dans le labyrinthe. Par exemple, vous pourriez utiliser le capteur de distance situé sur le côté droit du robot.

Feuille de travail 11

Sortie d'un labyrinthe par la gauche



Cette feuille de travail explique comment utiliser l'ordinateur de bord, haut-parleur et processeur de signal numérique pour jouer des notes de musique.

La deuxième partie de cet exercice montre comment réaliser un simple interprète musical pour transcrire des partitions et s'amuser à faire jouer le robot

Activité

Autonome

Niveau de
compétence

Facile

Dispositif sonore

Le robot peut créer des tonalités audio numériques qui sont diffusées à l'aide du petit haut-parleur embarqué. La fréquence des sons peut varier de 1 à 10 000 Hz et leur durée de 1 à 10 000 ms. Cela signifie que le robot couvre une bonne partie de la gamme audio que les humains peuvent entendre. Pour ce faire, vous devez utiliser la commande API suivante.

PlayNote <note> <temps>

Par exemple, pour jouer le do central, qui a une fréquence de 262 Hz, pendant 1 seconde, il faut utiliser :

PlayNote 262 1000

Pour jouer une série de notes, il suffit de répéter l'appel API et de remplacer les valeurs appropriées. Si vous vouliez jouer un morceau de musique, cette méthode deviendrait vite fastidieuse. Voici donc une méthode pour réaliser un interprète musical très simple. Elle est basée sur l'utilisation de lettres pour représenter les notes de musique et de chiffres pour signifier la durée de chaque note.

En commençant par le do central, la séquence de lettres serait C, D, E, F, G, A, B. Le chiffre 1 correspondrait à une unité de temps, le chiffre 2 au double, etc. Vous pouvez décider du nombre de millisecondes que représente chaque unité de temps. À ce stade, les notes fractionnaires, les dièses et les bémols sont ignorés et seule la première octave est couverte. Vous pourrez étendre les capacités de l'interprète ultérieurement.

La première étape consiste à trouver la fréquence des notes de l'octave à partir du do central. Le do moyen est à 262 Hz, le ré à 294 Hz. Vous pouvez trouver les fréquences des autres notes.

Interprète musical

Pour jouer un morceau de musique, vous devez prendre une page de partition et calculer le nom et la durée de chaque note. Par exemple, "D" 2 signifie que la note D est jouée deux fois plus longtemps.

Vous pouvez ensuite entrer les détails dans le programme à la page suivante.

Le programme consiste en une macro qui décode le nom de chaque note et sa durée. Ainsi, par exemple, la note "D" est remplacée par la valeur de fréquence de 294 Hz. De la même manière, les valeurs numériques sont remplacées par un nombre qui définit combien de millisecondes il faut jouer la note "D".

Jouez cet air

```
// Un exemple écrit en pseudo-code
// Interprète musical

play(mynote duration) //Macro pour décoder votre musique
  SI note == "C" ALORS note = 262
  ELSE-IF mynote == "D" ALORS note =
  294 ELSE-IF mynote == "E" ALORS note
  = ? ?? ELSE-IF note == "F" ALORS note
  = ? ?? ELSE-IF note == "G" ALORS note
  = ? ?? ELSE-IF note == "A" ALORS note
  = ? ?? ELSE-IF note == "B" ALORS note
  = ? ?? END-IF

  IF duration == 1 THEN time = 200 //Time unit is 200ms ELSE-IF
  duration == 2 THEN time = 400
  ELSE-IF durée == 3 ALORS temps = 600
  ELSE-IF durée == 4 ALORS temps = 800
  END-IF
```

PlayNote temps de note

END-play

```
APPEL play("A" 1) /Pouvez-vous trouver le de cette musique ?
                   titre
```

```
APPEL play("G" 1)
APPEL play("A" 1)
APPEL play("C" 1)
APPEL play("A" 1)
```

```
APPEL play("G" 1)
APPEL play("A" 1)
APPEL play("C" 1)
APPEL play("C" 1)
```

```
APPEL play("D" 1)
APPEL play("E" 1)
APPEL play("D" 1)
APPEL play("E" 1)
```

```
APPEL play("D" 3)
APPEL play("E" 1)
```

À vous de jouer

(a) Bien qu'il s'agisse d'un interprète musical très simple, il se prête à des extensions et à des améliorations. Complétez d'abord le programme pour les notes énumérées.

(b) Ensuite, ajoutez un décodage supplémentaire pour gérer les "dièses", les "bémols", les "silences" et les autres octaves.

(c) Enfin, ajoutez du code pour gérer le "tempo". Ainsi, au lieu d'avoir un temps de base de 200 ms, vous pourriez utiliser le tempo comme facteur d'échelle pour rendre la musique plus réaliste.

Cette section a expliqué la simple commande **PlayNote** et a montré qu'avec un peu de

Jouez cet air



Cette feuille de travail explique comment utiliser l'ordinateur de bord, haut-parleur et carte microSD pour lire des documents préenregistrés.

La deuxième partie de l'exercice montre comment vous pouvez construire une simple machine à reproduire les sons.

Activité

Autonome

Niveau de compétence

Facile

Cet exercice simple consiste à copier un morceau de votre musique préférée sur une carte microSD et à demander au robot de vous le restituer. Plutôt facile. Très amusant.

La première étape consiste à trouver une carte microSD préformatée. Il s'agit des minuscules cartes mémoire qui s'insèrent dans les téléphones intelligents et les tablettes. L'étape suivante consiste à insérer temporairement la carte dans votre appareil mobile et à y copier un morceau de musique. Si votre musique se trouve sur un ordinateur de bureau ou un ordinateur portable, vous aurez peut-être besoin d'un adaptateur de carte SD (à moins que votre équipement ne soit très récent).

Le format d'enregistrement pris en charge par le robot est le format Waveform Audio File Format, communément appelé WAV en raison de son extension de fichier. Assurez-vous que votre logiciel de copie sonore est configuré pour produire des sorties au format .wav en utilisant l'un des paramètres de taux d'échantillonnage/profondeur de bits suivants.

8KHz @ 8-bit, 8KHz @ 16-bit, 16KHz @ 8-bit ou 16KHz @ 16-bit

Assurez-vous que le robot est éteint, puis retirez la carte microSD de votre équipement de copie et insérez-la dans le logement de la carte microSD du robot.

Vous êtes maintenant prêt à faire jouer de la musique au robot à l'aide des commandes API suivantes.

CardInit et **CardPlayback <fichier>**

La première commande initialise l'électronique embarquée du robot qui contrôle la carte SD. Le paramètre <filename> définit le nom du fichier .wav à jouer. Cette commande renvoie effectivement une valeur (0 signifie OK, 239 signifie que le fichier n'a pas été trouvé et 255 signifie une erreur).

Voici un exemple de lecture d'un fichier "wav" préenregistré.

status = CardPlayback track1.wav

Dans un programme complet, vous pourriez faire en sorte qu'un test soit effectué pour voir si la variable nommée *status* a une valeur différente de zéro, et si c'est le cas, signaler une erreur en conséquence.

À vous de jouer

Vous en êtes maintenant au stade où vous pouvez enregistrer une série de pistes "wav" sur une carte microSD, puis écrire un programme très simple pour lire une seule piste par le biais de l'interface utilisateur du robot.

Machine à reproduire le son

L'exemple ci-dessous montre une méthode de sélection des pistes en utilisant SW1 pour faire défiler les pistes disponibles et SW2 pour lancer la lecture de la piste sélectionnée. Si vous prévoyez d'utiliser un appareil mobile pour effectuer ces tâches, vous pouvez utiliser des boutons programmables sur l'écran de l'appareil.

```
// Un exemple écrit en pseudo-code SW1_prev
= 0
SW2_prev = 0
piste = 1
LCDPrint 0 0 "Piste = "
LCDNumber 54 0 track
CardInit

LOOP-FOREVER

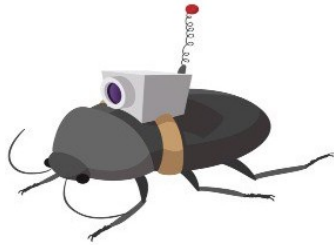
SW1 = ReadSwitch 0 // C'est le bouton de sélection de
la piste IF SW1 > SW1_prev THEN // Détecter le front
montant
    SW1_prev = SW1
    SI piste < 10 ALORS
        piste = piste + 1
        Numéro LCD 54 0
        piste
    ELSE track = 1
    END-IF
ELSE-IF SW1 < SW1_prev THEN // Détection du front
descendant SW1_prev = SW1
END-IF

SW2 = ReadSwitch 1 // C'est le bouton d'exécution
IF SW2 > SW2_prev THEN // Détection du front
montant
    SW2_prev = SW2
    SI piste == 1 ALORS
        CardPlayback track1.wav
    ELSE-IF piste == 2 ALORS
        CardPlayback track2.wav
    ELSE-IF piste == 3 ALORS
        CarteLecture track3.wav
    ELSE-IF piste == 4 ALORS
        CarteLecture track4.wav
    etcetera
    END-IF
ELSE-IF SW2 < SW2_prev THEN // Détection du front
descendant SW2_prev = SW2
END-IF

END-LOOP
```

A vous (encore)

Bien qu'il s'agisse d'un appareil de lecture sonore très simple, il se prête à des extensions et à des améliorations. Ajoutez un décodage supplémentaire pour gérer la sélection des pistes "en avant" et "en arrière" au lieu de devoir parcourir les pistes dans une seule direction.



Cette feuille de travail rassemble des tâches qui ont été présentées dans les sections précédentes, pour résoudre un problème typique du monde réel.

Le problème à résoudre est de créer un dispositif d'espionnage. Le robot se rend à un endroit, enregistre un discours, revient et rejoue le discours enregistré. Il s'agit d'un dispositif d'espionnage.

Activité

Lié

Niveau de compétence

Intermédiaire

L'objectif de cet exercice est de développer une application qui contrôlera le robot à l'aide du capteur d'inclinaison et qui utilisera également la fonction audio pour enregistrer et reproduire des paroles. Bien que cet exemple utilise App Inventor, un logiciel équivalent pourrait être utilisé pour s'adapter à votre développement et aux plates-formes cibles.

Ne vous laissez pas décourager. Cet exercice semble plus difficile qu'il ne l'est en réalité. Si vous analysez le problème, vous découvrirez rapidement que vous avez déjà abordé certaines tâches. Par exemple, la feuille de travail *Incliner et tourner* expliquait comment piloter le robot à l'aide d'un appareil mobile. Vous avez également appris à utiliser la fonction de lecture du son dans la feuille de travail *Robo-DJ*, de sorte que le seul élément manquant à la solution est l'apprentissage de l'enregistrement du son.

L'enregistrement du son à l'aide du robot est un peu plus complexe, mais pas difficile, car l'enregistrement du son se fait à l'aide de l'appareil.

Les appels d'API suivants sont disponibles pour vous simplifier la tâche.

La première chose à faire est d'initialiser l'électronique embarquée du robot qui contrôle la carte microSD. Cette tâche est réalisée avec l'appel API **CardInit**. Cet appel renvoie un indicant l'état du système. Le chiffre 0 indique que le système est OK, le chiffre 254 indique une erreur et le chiffre 255 indique qu'aucune carte n'a été détectée dans le logement de la carte microSD.

L'étape suivante consiste à commencer l'enregistrement à partir du microphone embarqué sur la carte microSD. La commande API suivante est utilisée pour effectuer cette tâche.

CardRecordMic <bitdepth> <samplerate> <time> <filename>

Les paramètres <bitdepth> et <samplerate> doivent être réglés pour correspondre à l'un des quatre modes d'enregistrement pris en charge par le robot. Ces modes sont les suivants : 8-bit@8KHz, 16-bit@8KHz, 8-bit@16KHz ou 16-bit@16KHz. Le premier réglage permet d'obtenir la plus petite taille de fichier sur la carte SD, tandis que le quatrième réglage permet d'obtenir la meilleure qualité d'enregistrement (mais occupe une taille de fichier plus importante). Plutôt que d'entrer les valeurs réelles, un code numérique est utilisé, comme indiqué dans les tableaux ci-dessous.

	8 bits	16 bits
profondeur de bits	0	1

	8KHz	16KHz
échantillonner	0	1

Le paramètre <time> définit la durée de l'enregistrement. Il peut prendre une valeur comprise entre 1 et 65535 secondes. Le dernier paramètre spécifie le nom de l'enregistrement.

Voici un exemple pour réaliser un enregistrement au format .wav, à la meilleure qualité, pendant 10 secondes.

`CardRecordMic 1 1 10 track1.wav`

Cet appel de l'API renvoie une valeur indiquant l'état de la commande. Par exemple renvoie le chiffre 0 pour indiquer que le système est OK, le chiffre 1 indique que le fichier existe déjà sur la carte et le chiffre 255 signifie qu'une erreur s'est produite.

Si vous souhaitez supprimer un enregistrement de la carte, vous pouvez utiliser la commande suivante.

`CardDelete <nom du fichier>`

L'étape suivante consiste à créer une interface utilisateur pour l'appareil mobile.

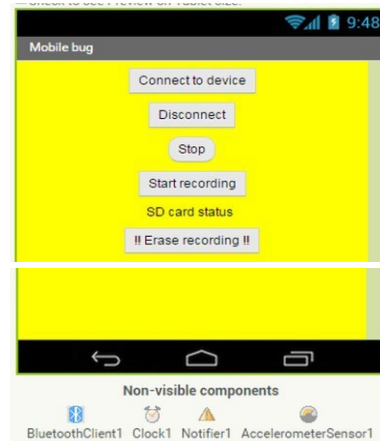
Voici un exemple de mise en page simple telle qu'elle apparaîtrait dans la vue Designer de l'application Inventeur.

Les deux premiers boutons programmables concernent la fonction de communication Bluetooth du robot.

Le bouton *Stop* arrête le mouvement du robot en mettant la vitesse des moteurs à zéro.

Le bouton *Démarrer l'enregistrement* émet la commande API `CardRecordMic` (comme indiqué ci-dessus).

Le bouton *Effacer l'enregistrement* supprime l'enregistrement de la carte. Au même moment, vous pourriez afficher un message sur l'appareil mobile ou sur le gLCD du robot qui indique ...



"Cet enregistrement s'autodétruit au bout de 15 secondes."

Pour vous aider à vous familiariser avec le codage par blocs, vous trouverez à la page suivante quelques exemples montrant les événements déclenchés dans la vue Bloc d'App Inventor.

À vous de jouer

Vous devez modifier l'application *Tilt and turn* pour qu'elle devienne *Mobile bug*. Une fois que vous l'aurez fait, vous pourrez envisager d'ajouter des fonctionnalités supplémentaires.

Par exemple, écrivez du code pour que chaque fois que vous appuyez sur le bouton *Start recording*, l'enregistrement se fasse sous un nom de fichier différent. Par exemple, *track1.wav*, puis *track2.wav*, etc.

Un autre dispositif utile serait une sorte de verrouillage empêchant la lecture d'un enregistrement tant que le robot n'est pas rentré à la base en toute sécurité : étendez votre programme de manière à ce qu'un certain code doive être saisi avant que le dispositif de lecture puisse être activé.

Capture d'écran des événements déclenchés par l'appui sur le bouton *Démarrer l'enregistrement*.

```

when BTN_start_recording .Click
do
  if 0 =
    call CardRecordMic
      bitdepth 0
      samplerate 0
      time 10
      filename "track1.wav"
    then set LBL_sd_card_status .Text to "Recording"
    else set LBL_sd_card_status .Text to "!! Card problems !!"
  
```

Capture d'écran
des

```

when BTN_erase_recording .Click
do
  if 0 =
    call CardDelete
      filename "track1.wav"
    then set LBL_sd_card_status .Text to "Deleting sound track"
    else set LBL_sd_card_status .Text to "!! Card problems !!"
  
```

événements

déclenché lorsque la touche *!!! Effacer l'enregistrement !!!* est pressée.

Feuille de travail 15

Robo-Pop



Cette feuille de travail rassemble certaines tâches, couvertes dans

Les sections précédentes permettent de faire danser le robot sur de la musique lorsqu'un son (comme des battements de mains) est détecté par le microphone.

Au lieu de danser, vous pourriez programmer le robot pour qu'il

faire une "célébration" il s'agit d'un événement marqué par un football

Activité

Lié

Niveau de compétence

Intermédiaire

Ne vous laissez pas décourager. Cet exercice semble plus difficile qu'il ne l'est en réalité. Si vous analysez le problème, vous découvrirez rapidement que vous avez déjà abordé la plupart des tâches. Pour

Par exemple, la feuille de travail *Let's move it, move it* expliquait comment contrôler les moteurs du robot. Vous avez également appris à utiliser la fonction de lecture du son dans la fiche *Robo-DJ*, de sorte que le seul élément manquant à la solution est l'apprentissage de la détection d'un son.

L'appel de l'API pour détecter un son est **ReadMic**. Cet appel renvoie une valeur, de sorte que la façon de l'utiliser est de l'affecter à une variable nommée (par exemple, `sound_level`), comme indiqué ci-dessous.

```
sound_level = ReadMic
```

La valeur renvoyée peut être comprise entre 0 et 4095, ce qui indique l'amplitude du son. Vous devrez peut-être effectuer quelques expériences pour déterminer la valeur seuil correspondant au "son" que vous envisagez d'utiliser comme "déclencheur".

Cet

```
exemple // Un exemple écrit en pseudo-code
montre // Robo-pop déclenché par la détection du son
comme  threshold = 100                this according to your situation DO
nt      sound_level = ReadMic//Clap your hands to make
combiner a noise UNTIL sound_level >= threshold
un son
détecteur CALL play_some_dance_music //Utiliser le code de "Robo-DJ"
avec      CALL perform_some_groovy_moves //Utiliser le code de "Let's move it"
autres
le travail
que vous
avez
effectué.
```

Feuille de travail 15

Robo-Pop

Votre robot danse et joue de la musique. Que manque-t-il donc ? Des lumières, des rayons laser - quelque chose pour faire vibrer le public. Bien que le robot n'ait pas de rayons laser, il possède un écran LCD capable d'afficher des formes graphiques - lignes, rectangles ou carrés - et de manipuler des pixels individuels. En utilisant la capacité du robot à dessiner des rectangles, vous pouvez créer un spectacle de faisceaux lumineux.

Spectacle de faisceaux lumineux

La commande API suivante demande à l'écran graphique LCD de dessiner un rectangle.

```
LCDRect <x1> <y1> <x2> <y2>
```

Les paramètres <x1> et <y1> définissent la partie supérieure gauche du rectangle et <x2> et <y2> la partie inférieure droite. Voici un exemple pour dessiner un carré de 10 par 10 pixels aux coordonnées 0,0.

```
LCDRect 0 0 10 10
```

A vous (encore)

Maintenant que vous avez vu comment dessiner un carré noir, mettons-y un peu d'animation. Votre défi consiste à faire en sorte que l'un des carrés noirs, ou les deux, illustrés dans le diagramme ci-dessous, se déplace sur l'écran LCD et revienne en arrière (comme un faisceau lumineux sur la piste de danse d'une discothèque).



Si le point de départ du carré A est situé à la coordonnée 0,4 et celui du carré B à la coordonnée 116, 18, les deux carrés auront de la place pour glisser l'un sur l'autre.

La prochaine chose à prendre en compte est la manière de faire bouger les carrés.

La méthode la plus simple consiste à utiliser quelques variables nommées qui agissent comme des compteurs pour donner le décalage de la position de chaque carré, et à les utiliser dans l'appel à `LCDRect`.

```
LCDRect counterA 0 bottom_right_A 13
```

Par exemple, le compteur du carré A commence à 0. Pour déplacer le carré vers la droite, il faut incrémenter le compteur (d'une valeur appropriée, disons 5) et appeler à nouveau `LCDRect`. `Bottom_right` est une autre variable nommée qui prend la valeur : `bottom_right_A = counterA + 10`

Vous devrez insérer un court délai avant que la séquence ne soit répétée afin que le carré noir se déplace en douceur sur l'écran. Dans votre programme, vous devrez vérifier que le compteur A atteint l'extrémité droite de l'écran LCD et décider s'il faut le ramener à son point de départ ou simplement retracer son mouvement (comme dans le diagramme).

Pour aller plus loin

Défis 16-20



Les 5 derniers défis sont rassemblés ici pour ceux qui veulent aller plus loin.

Nous ne vous donnons pas beaucoup d'indications ici car la majeure partie du travail s'appuie sur les feuilles de travail précédentes et si vous êtes arrivé jusqu'ici, vous serez un expert en programmation !

Défi 16 - Enregistrement des données

Les possibilités sont nombreuses : Utilisez les commandes API de la carte SD pour enregistrer les données du capteur lors d'un voyage ou le niveau de luminosité lorsque le soleil se couche le soir. Vous pouvez également faire tourner l'appareil tout en enregistrant le niveau de lumière pour produire un graphique à 360° de la lumière dans la pièce.

Défi 17 - Utilisation des servos

Attachez des servomoteurs au port d'extension et transformez le robot en pelleuse ou donnez-lui un "pied" pour taper dans un ballon.

Il y a 4 servomoteurs à l'avant du robot et les commandes pour les contrôler sont listées dans la référence API.

Défi 18 - Maquillez ma voiture

Concevez une coque pour le robot Formula AllCode qui rendra votre robot unique. Vous pouvez peut-être utiliser une imprimante 3D pour créer un corps vraiment unique pour le robot.

Fixez-le au châssis du robot en utilisant les 4 trous situés près des roues.

Défi 19 - L'électronique sur mesure

Augmentez l'individualité de votre robot en ajoutant des LED et d'autres composants électroniques personnalisés à ses ports d'extension.

N'oubliez pas de câbler correctement vos composants électroniques afin de ne pas endommager le microcontrôleur !

Défi 20 - Accélération et orientation

Saviez-vous qu'il y a un accéléromètre intégré ? Vous pouvez l'utiliser pour mesurer l'accélération du robot AllCode, les forces qui s'exercent sur lui lorsqu'il prend un virage, et même dans quelle direction il se trouve (Einstein nous a appris que la gravité et l'accélération sont essentiellement la même chose !)

La commande API pour lire l'accéléromètre est [ReadAxis](#).

Cette section répertorie les appels de l'API par groupes fonctionnels, de sorte qu'il est facile de trouver la commande particulière que vous recherchez. Il existe différents types d'appels - certains renvoient une valeur, d'autres non. Certains requièrent un ou plusieurs paramètres, d'autres non.

Par exemple, l'appel API **SetMotors** nécessite deux paramètres pour définir la vitesse des deux moteurs, alors que l'appel **ReadLight** renvoie simplement une valeur (c'est-à-dire la valeur du capteur de lumière). D'autres appels nécessitent un paramètre et renvoient également une valeur.

Un bon exemple est l'appel API **ReadIR**. Cet appel nécessite un paramètre pour définir le capteur IR que vous souhaitez vérifier. La valeur de retour est la mesure de la distance pour ce capteur particulier.

Connexion			
Retour	Commandement	Paramètre(s)	Description
statut	ComOpen	port	Ouvrir le port COM Port= 1 à 255 Statut= 0(OK) ou 255(erreur)
statut	ComClose		Fermer le port Statut= 0(OK) ou 255(erreur)
version	ObtenirAPIVersion		Renvoie le numéro de version de l'API Version+ 1 à 65535

Capteurs			
Retour	Commandement	Paramètre(s)	Description
valeur	ReadSwitch	index	Lecture de la valeur du commutateur Index= 0(gauche) ou 1(droite) Valeur= 0(faux) ou 1(vrai)
valeur	ReadIR	index	Lecture d'un capteur IR Index= 0 à 7 Valeur = 0 à 4095
valeur	Ligne de lecture	index	Lecture d'un capteur de ligne Index= 0(gauche) ou 1(droite)
valeur	Lire la lumière		Lecture d'un capteur de lumière Valeur = 0 à 4095
valeur	ReadMic		Lecture du capteur de microphone Valeur = 1 à 4095
valeur	ReadAxis	index	Lecture d'un axe de l'accéléromètre Index= 0 (x), 1 (y) ou 2 (z) Valeur= -32768 à 32768

Annexe 1

Tableau de référence de l'API

Moteurs			
Retour	Commandement	Paramètre(s)	Description
	SetMotors	Gauche droite	Régler la vitesse des moteurs Gauche = -100 à 100 Droite = -100 à 100
	Les attaquants	distance	Déplacement vers l'avant distance (en mm) Distance = 0 à 1000
	Vers l'arrière	distance	Distance de recul (en mm) Distance = 0 à 1000
	Gauche	angle	Angle du virage à gauche (en degrés) Angle = 0 à 360
	Droit	angle	Angle de rotation droit (en degrés) Angle = 0 à 360

LED / Haut-parleur			
Retour	Commandement	Paramètre(s)	Description
	LEDWrite	valeur	Ecriture de la valeur dans les LEDs Valeur = 0 à 7
	LEDOn	index	Allumer une LED Index = 0 à 7
	LEDOff	index	Éteindre et LED Éteindre Index = 0 à 7
	PlayNote	temps de note	Note audio de sortie (en Hz) pour le temps (en ms) Note = 1 à 1000 Temps = 1 à 1000

Servo			
Retour	Commandement	Paramètre(s)	Description
	ServoEnable	index	Activation d'une voie de servo Index = 0 à 3
	ServoDisable	index	Désactivation d'une voie de servo Index = 0 à 3
	ServoSetPos	Position de l'index	Définir une position du servo Index = 0 à 3 Position = 0 à 255
	ServoAutoMove	Position de l'index	Déplacement automatique vers une position servo Index = 0 à 3 Position = 0 à 255
	ServoMoveSpeed	vitesse	Réglage de la vitesse de déplacement automatique du servo Vitesse = 1 à 50

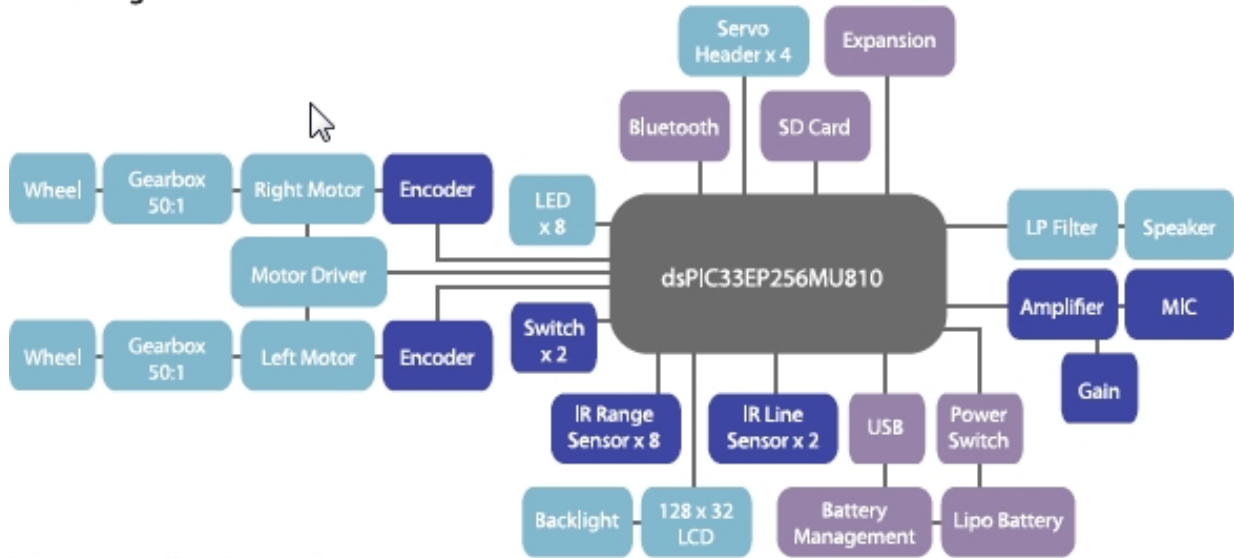
LCD			
Retour	Commandement	Paramètre(s)	Description
	LDClear		
	LCDPrint	X Y texte	Imprimer le texte sur l'écran LCD X = 0 à 127 Y = 0 à 31 Texte = <chaîne>
	Numéro de l'écran LCD	X Y valeur	Imprimer une valeur entière sur l'écran LCD X = 0 à 127 Y = 0 à 31 Valeur = -32768 à 32767
	LCDPixel	X Y État	Dessine un pixel sur l'écran LCD X = 0 à 127 Y = 0 à 31 État = 0 (éteint) ou 1 (allumé)
	LCDLine	X1 Y1 X2 y2	Tracer une ligne sur l'écran LCD X1= 0 à 127 Y1 = 0 à 31 X2 = 0 à 127 Y2 = 0 à 31
	LCDRect	X1 Y1 X2 y2	Dessinez un rectangle sur l'écran LCD X1 = 0 à 127 Y1 = 0 à 31 X2 = 0 à 127 Y2 = 0 o 31
	Rétro-éclairage LCD	Valeur	Régler la luminosité du rétroéclairage de l'écran LCD Valeur = 0 à 100
	Options LCD	Avant-plan Arrière-plan transparent	Définit l'option de dessin sur l'écran LCD Premier plan = 0 (blanc) ou 1 (noir) Arrière-plan = 0 (blanc) ou 1 (noir) Transparent = 0 (faux) ou 1 (vrai)

Carte SD			
Retour	Commandement	Paramètre(s)	Description
statut	CardInit		Initialiser la carte SD État = 0 (OK), 254 (erreur) ou 255 (pas de carte)
statut	Créer une carte	nom de fichier	Créer un nouveau fichier Nom de fichier = <chaîne> Statut = 0 (OK), 1 (le fichier existe) ou 255 (erreur)
statut	CardOpen	nom de fichier	Ouvrir un fichier existant Nom de fichier = <chaîne> Statut = 0 (OK), 239 (fichier non trouvé) ou 255 (erreur)
statut	Suppression de la carte	nom de fichier	Supprimer un fichier Nom de fichier = <chaîne> Statut = 0 (OK) ou 255 (erreur)
statut	CardWriteByte	données	Écriture d'un octet de données dans le fichier ouvert Données = 0 à 255 Statut = 0 (OK) ou 255 (erreur)
données	CardReadByte		Lire un octet de données dans le fichier ouvert Données = 0 à 255
statut	CardRecordMic	Bitdepth Samplerate Time nom de fichier	Enregistrer le microphone Bitdepth = 0 (8bit) ou 1 (16bit) Samplerate = 0 (8K) ou 1 (16K) Filename = <string> (nom de fichier) Statut = 0 (OK), 239 (le fichier existe) ou 255 (erreur)
statut	CarteLecture	nom de fichier	Lire un fichier audio Nom de fichier = <chaîne> Statut = 0 (OK), 239 (fichier non trouvé) ou 255 (erreur)
statut	CardBitmap	X Y nom de fichier	Affichage d'une image sur l'écran LCD X = 0 à 127 Y = 0 à 31 Nom de fichier = <chaîne> Statut = 0 (OK), 239 (fichier non trouvé) ou 255 (erreur)

Annexe 2

Connexions des broches du microcontrôleur

Block Diagram



Microcontroller Connections

Motors		IR Range		Expansion J15		Bluetooth		SD Card	
Left Speed Control	D1	IR1 Enable	D8	1	VBATT	TX	F12	MOSI	D4
Left Direction Control	D9	IR1 Signal	B4	2	SW3	RX	F8	MISO	D5
Left Encoder Feedback	B11	IR2 Enable	D11	3	G12	Pair Status	F0	SCK	D6
Right Speed Control	D2	IR2 Signal	B5	4	G13	Reset	F13	CS	D7
Right Direction Control	D3	IR3 Enable	D12	5	G14	Servo Headers		USB	
Right Encoder Feedback	B12	IR3 Signal	B14	6	G15	Servo1	A7	USB Detect and LED	B13
LCD		IR4 Enable	D13	7	D9	Servo2	A9	D+	G2
Data	G8	IR4 Signal	B15	8	D10	Servo3	A10	D-	G3
Clock	G6	IR5 Enable	D16	9	D15	Servo4	A14	Switches	
A0	A9	IR5 Signal	B8	10	GND	E-Header Expansion		SW1	A8
CS	A2	IR6 Enable	A5	Expansion J16		1	E0	SW2	A1
Reset	A4	IR6 Signal	B9	1	A15	2	E1		
LEDs		IR7 Enable	A6	2	C1	3	E2		
D0	E0	IR7 Signal	B10	3	C2	4	E3		
D1	E1	IR8 Enable	C4	4	C3	5	E4		
D2	E2	IR8 Signal	B8	5	C13	6	E5		
D3	E3	Outputs		6	C14	7	E6		
D4	E4	Microphone	B3	7	G1	8	E7		
D5	E5	Speaker	F3	8	F1	9	GND		
D6	E6	Light Sensor	B2	9	F2	10	GND		
D7	E7	Battery Voltage	E9	10	GND				

Annexe 3

Fréquences des notes de musique

Note	Freq (Hz)
C ₁	33
C [#] ₁ /D ^b ₁	35
D ₁	37
D [#] ₁ /E ^b ₁	39
E ₁	41
F ₁	44
F [#] ₁ /G ^b ₁	46
G ₁	49
G [#] ₁ /A ^b ₁	52
A ₁	55
A [#] ₁ /B ^b ₁	58
B ₁	62

Note	Freq (Hz)
C ₂	65
C [#] ₂ /D ^b ₂	69
D ₂	73
D [#] ₂ /E ^b ₂	78
E ₂	82
F ₂	87
F [#] ₂ /G ^b ₂	93
G ₂	98
G [#] ₂ /A ^b ₂	104
A ₂	110
A [#] ₂ /B ^b ₂	117
B ₂	123

Note	Freq (Hz)
C ₃	131
C [#] ₃ /D ^b ₃	139
D ₃	147
D [#] ₃ /E ^b ₃	156
E ₃	165
F ₃	175
F [#] ₃ /G ^b ₃	185
G ₃	196
G [#] ₃ /A ^b ₃	208
A ₃	220
A [#] ₃ /B ^b ₃	233
B ₃	247

Note	Freq (Hz)
C ₄	262
C [#] ₄ /D ^b ₄	277
D ₄	294
D [#] ₄ /E ^b ₄	311
E ₄	330
F ₄	349
F [#] ₄ /G ^b ₄	370
G ₄	392
G [#] ₄ /A ^b ₄	415
A ₄	440
A [#] ₄ /B ^b ₄	466
B ₄	494

Note	Freq (Hz)
C ₅	523
C [#] ₅ /D ^b ₅	554
D ₅	587
D [#] ₅ /E ^b ₅	622
E ₅	659
F ₅	698
F [#] ₅ /G ^b ₅	740
G ₅	784
G [#] ₅ /A ^b ₅	831
A ₅	880
A [#] ₅ /B ^b ₅	932
B ₅	988

Note	Freq (Hz)
C ₆	1047
C [#] ₆ /D ^b ₆	1109
D ₆	1175
D [#] ₆ /E ^b ₆	1245
E ₆	1319
F ₆	1397
F [#] ₆ /G ^b ₆	1480
G ₆	1568
G [#] ₆ /A ^b ₆	1661
A ₆	1760
A [#] ₆ /B ^b ₆	1865
B ₆	1976

Note	Freq (Hz)
C ₇	2093
C [#] ₇ /D ^b ₇	2217
D ₇	2349
D [#] ₇ /E ^b ₇	2489
E ₇	2637
F ₇	2794
F [#] ₇ /G ^b ₇	2960
G ₇	3136
G [#] ₇ /A ^b ₇	3322
A ₇	3520
A [#] ₇ /B ^b ₇	3729
B ₇	3951

Note	Freq (Hz)
C ₈	4186
C [#] ₈ /D ^b ₈	4435
D ₈	4699
D [#] ₈ /E ^b ₈	4978
E ₈	5274
F ₈	5588
F [#] ₈ /G ^b ₈	5920
G ₈	6272
G [#] ₈ /A ^b ₈	6645
A ₈	7040
A [#] ₈ /B ^b ₈	7459
B ₈	7902

Note	Freq (Hz)
C ₉	8372
C [#] ₉ /D ^b ₉	8870
D ₉	9398
D [#] ₉ /E ^b ₉	9956
E ₉	10548
F ₉	11176
F [#] ₉ /G ^b ₉	11840
G ₉	12544
G [#] ₉ /A ^b ₉	13290
A ₉	14080
A [#] ₉ /B ^b ₉	14918
B ₉	15804

Annexe 4

Définition du nom du robot

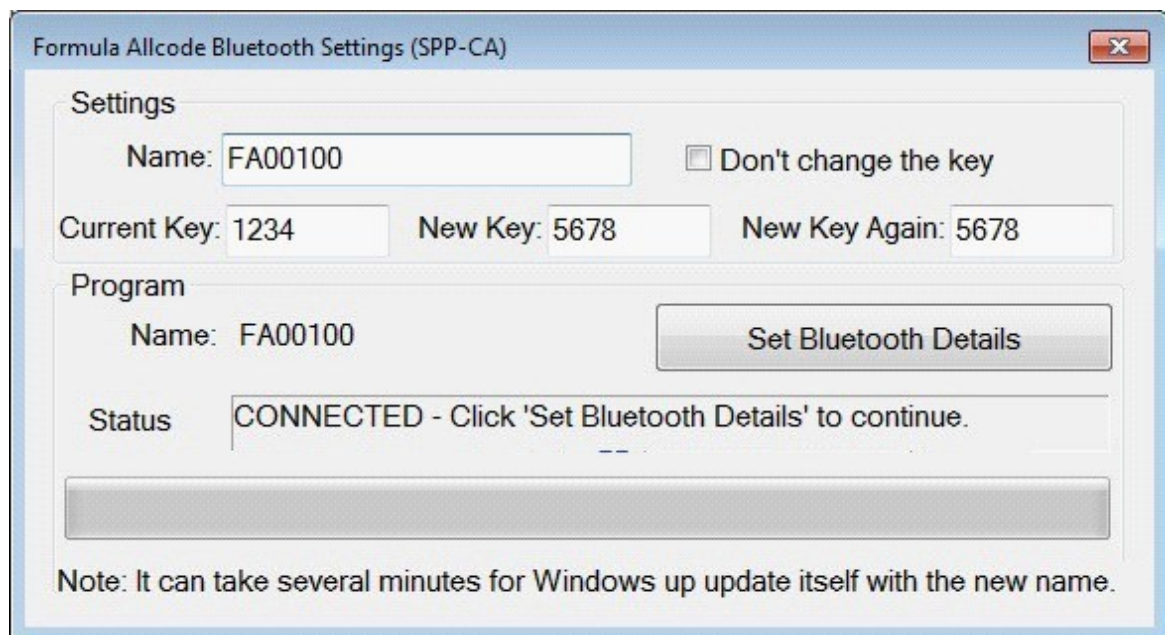
Le nom Bluetooth de la Formula AllCode peut être modifié en utilisant un PC sur lequel est installé le logiciel mLoader. Le logiciel mLoader peut être téléchargé à partir du site web de MatrixTSL.

<http://www.matrixtsl.com/formula-allcode/>

Une fois que vous avez téléchargé le logiciel mLoader, vous trouverez dans le dossier du logiciel un fichier intitulé Formula AllCode Rename. Double-cliquez sur ce fichier pour lancer le logiciel mLoader en mode de renommage Bluetooth.

Connectez le robot Formula AllCode au PC à l'aide du câble USB fourni et appuyez sur le bouton de réinitialisation du robot pour permettre au logiciel mLoader de voir le robot.

Vous pouvez ensuite utiliser le logiciel mLoader pour définir le nom Bluetooth et la clé d'appariement (parfois appelée code d'accès) en saisissant les informations requises dans le champ de saisie.



Nous vous recommandons de conserver la clé par défaut "1234", sauf si vous travaillez au sein d'un groupe d'utilisateurs et que vous souhaitez empêcher d'autres personnes de se connecter à votre robot. Si vous changez la clé de passe, vous devrez saisir la clé existante, alors essayez de vous en souvenir !

Si la clé a été changée et que vous ne vous en souvenez plus, veuillez contacter Matrix pour savoir comment la réinitialiser.

Annexe 5

Recharger le micrologiciel de l'API AllCode

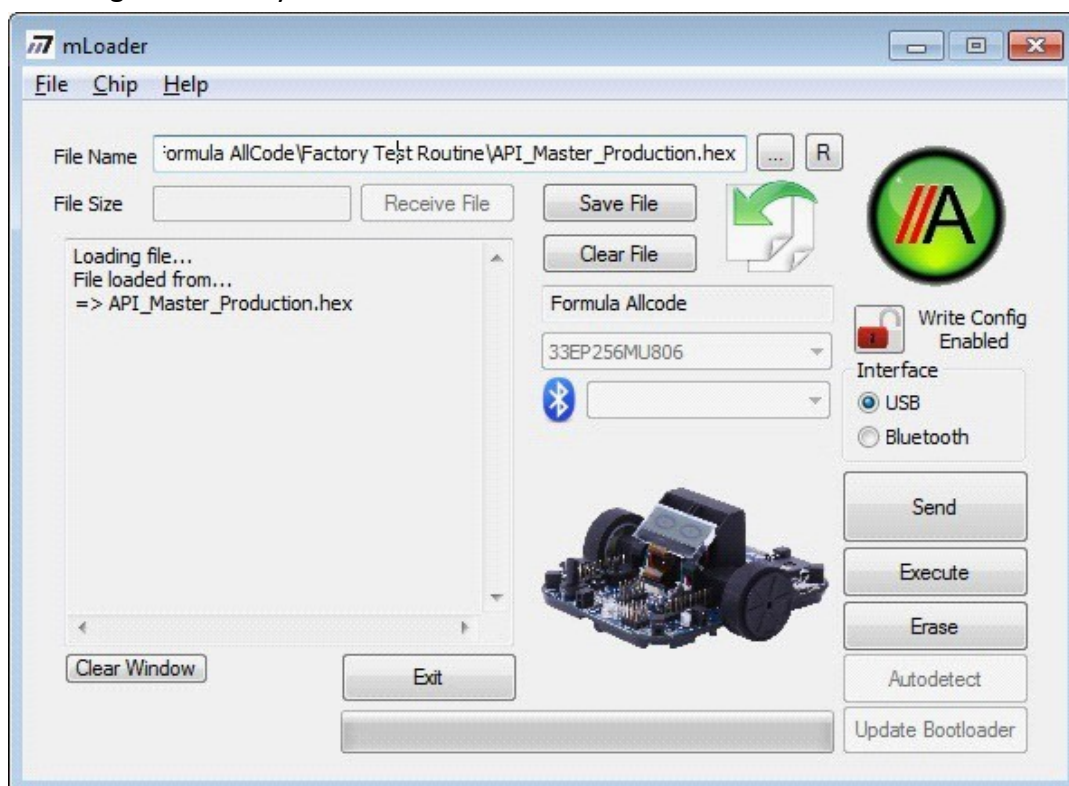
Par défaut, le robot Formula AllCode est livré avec le micrologiciel API déjà installé. Comme le robot est reprogrammable à l'aide de langages tels que Flowcode ou C, il est possible d'écraser la fonctionnalité de l'API AllCode installée en usine. Il est également possible que des versions plus récentes de l'API sont publiées et vous voudrez peut-être passer à la version la plus récente.

Le micrologiciel API peut être téléchargé sous la forme d'un fichier hexadécimal à partir de la page Formula AllCode du site Web de MatrixTSL. Pour charger le fichier hex sur la Formula AllCode, vous avez besoin d'une copie du dernier logiciel mLoader qui est également disponible sur le site web :

<https://www.matrixtsl.com/allcode/resources/>

Vous pouvez connecter la Formula AllCode au PC à l'aide du câble USB fourni ou à l'aide de la fonction la connexion de données Bluetooth. Ouvrez le logiciel mLoader et appuyez sur le bouton de réinitialisation du robot pour permettre au logiciel mLoader de voir le robot.

Cliquez sur l'icône "..." à côté du champ de texte Nom du fichier pour sélectionner le fichier hexagonal à envoyer à l'AllCode.



Cliquez sur le bouton Envoyer pour transférer le fichier hexadécimal à la Formula AllCode et restaurer le micrologiciel API. N'oubliez pas d'appuyer sur le bouton d'exécution une fois que le micrologiciel a été envoyé pour permettre au micrologiciel de Formula AllCode de s'exécuter.

Marques déposées

PIC, PICMicro et dsPIC sont des marques déposées de Microchip Technology. Raspberry Pi est une marque déposée de la Raspberry Pi Foundation.

Les autres noms de produits figurant dans ce document peuvent être des marques déposées de leurs propriétaires respectifs.

Notes de mise à jour

Version	Date de sortie	Notes
Version 1	25/02/2016	Version initiale
Version 2	04/05/2016	Converti en fichier Publisher
Version 3	04/08/2017	Liens fixes vers diverses ressources