

Information de Laboratoire de Programmation Arduino et Développement d'Application Mobile Supplémentaire

Programmation Arduino

Contrôle de Moteur Bluetooth pour des dispositifs ANDROID Plus Vieux (bluetooth régulier PAS LE)

Ce programme va utiliser les données entrantes d'un dispositif compatible BluetoothLE pour contrôler les moteurs avec la chip Bluetooth BLUESMiRF. Le capteur va agir comme une vérification qui va limiter quand les moteurs peuvent être utilisés.

1. Ouvre un nouveau croquis et inclus la bibliothèque AFMotor. Aussi n'oublie pas d'initialiser les moteurs DC (fait référence à la section C). Inclus et initialise aussi le capteur ultrason (fait référence à la section B). Continue par inclure la bibliothèque série en logiciel (Software serial), cette bibliothèque va nous permettre de nommer des pins pour agir comme RX (pin receveur) et TX (pin transmetteur). La bibliothèque logiciel peut être utilisée seulement après avoir tapé “#include <SoftwareSerial.h>” avant le “Void setup()”.
2. Attribue le pin 16 et 17 pour être RX et TX respectivement en tapant le code suivant “SoftwareSerial Serial1(16,17);” avant “void setup()”. Voir l'exemple ci-dessous comme référence.

```
#include <AFMotor.h>
#include <NewPing.h>

#define TRIGGER_PIN 6 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 5 //Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Max distance to ping in cm
NewPing sonar (TRIGGER_PIN,ECHO_PIN,MAX_DISTANCE); // passing the pins to newping to creat sonar.
boolean prox;
int distance;
#include <SoftwareSerial.h>

AF_DCMotor rightMotor(1, MOTOR12_64KHZ);
AF_DCMotor leftMotor(2, MOTOR12_64KHZ);
SoftwareSerial Serial1(16,17);

void setup()
{
```

3. Maintenant on a besoin d'initialiser un taux de communication (rate) pour le module bluetooth à la plaque à l'intérieur de “void setup()”. On va le faire en tapant “Serial1.begin(115200);” on a choisi 115200 car c'est le taux de défaut pour les deux modules. Tu vas aussi besoin de fixer la vitesse (speed) des moteurs à 200 en tapant “NomDeTonMoteur.setSpeed(200);”. Voir l'exemple ci-dessous comme référence.

```

void setup()
{
  Serial1.begin(115200);
  rightMotor.setSpeed(200);
  leftMotor.setSpeed(200);
}

void loop()
{

```

- À l'intérieur de la boucle (loop) on a besoin de continuellement vérifier la proximité d'autres objets (fait référence à la section B) ainsi que de lire le lien de série pour vérifier si on a reçu des commandes additionnelles de direction. Tu as besoin d'assigner une variable char pour entreposer les lettres reçues du téléphone intelligent. Voir l'exemple ci-dessous comme référence.

```

void loop()
{
  sensor_read(); // calling the function to determin if the robot is too close to an object.
  if( prox == false){
  if(Serial1.available()) // if theres data in the serial connection.
  {
    char c=Serial1.read(); // read and store data to c.

```

- Ajoute des déclarations pour contrôler la direction de l'auto en utilisant le .run(FORWARD) ou .run(BACKWARD) pour chaque moteur. Voir l'exemple ci-dessous comme référence.

```

    {
        char c=Serial1.read(); // read and store data to c.
    }
    if(c=='f')// if c is f then Move front
    {

rightMotor.run(FORWARD);
leftMotor.run(FORWARD);
    }
    if(c=='b')//if c is b then Move back
    {
        rightMotor.run(BACKWARD);
        leftMotor.run(BACKWARD);
    }
    if(c=='l')//if c is l then Move left
    {
        rightMotor.run(FORWARD);
        leftMotor.run(BACKWARD);
    }
    if(c=='r')//if c is r then Move Right
    {
        leftMotor.run(FORWARD);
        rightMotor.run(BACKWARD);
    }
    if(c=='s')//if c is s then stop moving
    {
        rightMotor.run(RELEASE);
        leftMotor.run(RELEASE);
    }
    }
    }
    }
void sensor_read(){

```

6. Maintenant ajoute des déclarations pour contrôler le capteur. Crée une nouvelle fonction et appelle-le “sensor_read()”. Fait en sorte que la fonction ping le capteur, imprime la distance au port de série et fixe la variable boolean qui a été initialisée plus tôt à “true” si la distance lue est plus grande que 10. Souviens-toi qu’une valeur lue de “0” veut dire que la distance est hors de portée. S’il y a quelque chose à moins de 10 cm du capteur, fixe la variable boolean à “false”. Voir l’exemple ci-dessous comme référence.

```

void sensor_read(){
    delay(250);
    distance= sonar.ping_cm();
    Serial1.println(distance);
    if(distance >10 or distance==0){
        Serial1.println("No Obstruction");
        prox= true;
    }else{Serial1.println(" Obstruction");
        prox= false;
    }
}
}

```

7. Finalement télécharge le programme à l’Arduino et test en utilisant l’application mobile qui va être développée dans un lab plus tard. Montre ton travail au TA.

Développement d'Application Mobile

- Iphone (**option 1**)
- **OU** pour des dispositifs Android avec n'importe quelle autre version de Bluetooth voir **option 2** plus loin dans le lab.

Activer les options développeurs (Android seulement)

1. Clique le bouton Maison pour arriver à l'écran d'accueil de ton téléphone.
2. Clique le bouton Menu, ensuite Paramètres (Settings), ensuite Applications.
3. Si ton téléphone à un paramètre Sources inconnues (Unknown sources), assure-toi qu'il est sélectionné.
4. Clique Développement.
5. Assure-toi que Débogage USB (USB Debugging) et Reste Réveillé (Stay Awake) sont sélectionnés.

Les étapes sont similaires pour des dispositifs plus nouveaux mais les options peuvent ptre trouvés à des places différentes sur le dispositif : Sur Android 4.0* et plus récent, ces paramètres sont dans Paramètres > Options développeur (Settings > Developer options).

Note: SureAndroid 4.2* et plus récent, les Options développeur (Developer options) sont cachés par défaut. Pour les rendre visible va à Paramètres > À propos (Settings > About phone) et clique Numéro de construction (Build number) sept fois. Retourne à l'écran précédent pour trouver les Option développeur.

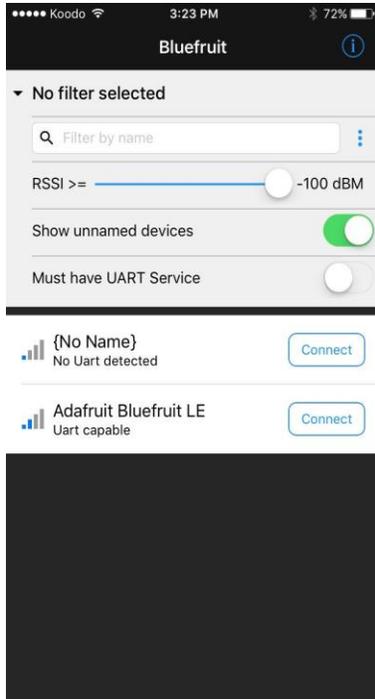
* Débogage USB (USB Debugging) et Reste Réveillé (Stay Awake) peuvent être trouvés dans un annuaire différent dans Paramètres dépendant de la version d'Android. Utilise le bouton recherche (search) dans Paramètres si tu ne peux pas les trouver.

Option 1:

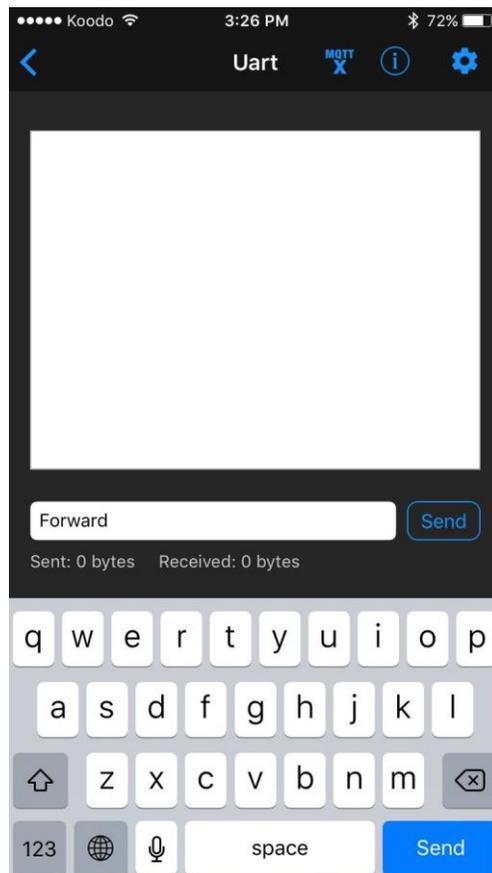
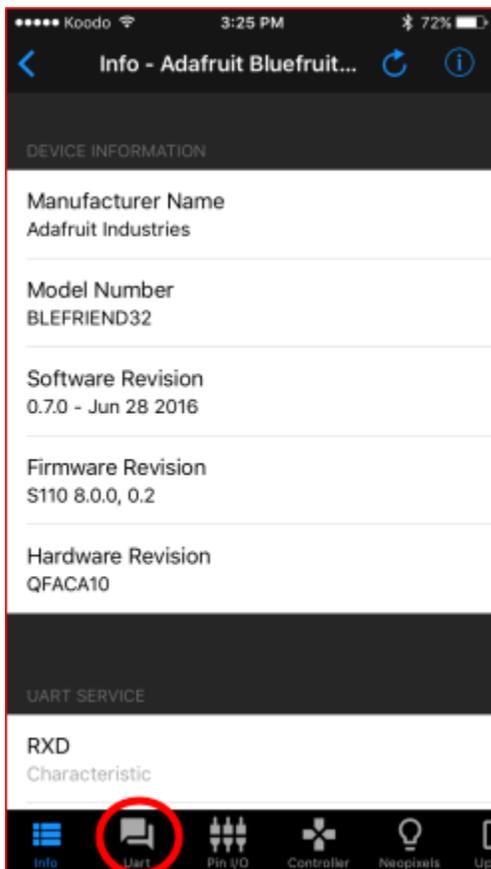
Pour iOS:

Assure-toi que BLUETOOTH est allumé.

1. Va au magasin d'application et installe l'application Bluefruit.
2. Ouvre-le et sélectionne le dispositif Adafruit Bluefruit LE et clique Connect.



3. Ensuite choisi l'option UART au bas de la page.



4. Tape tes commandes (“forward”, “stop”, ...etc) dans la boîte et clique Send pour contrôler ton auto.
5. Montre ton travail au TA.

Option 2:

Pour n’importe quelle autre version de Bluetooth qui n’est pas low energy.

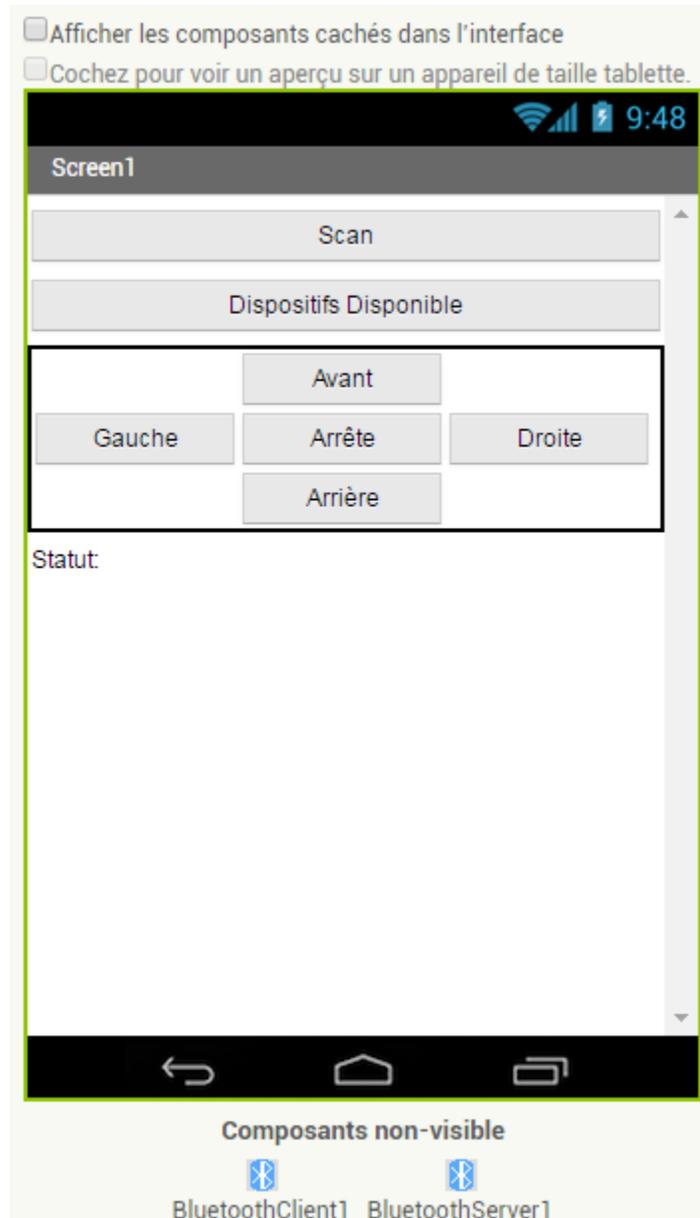
1. Ouvre MIT App Inventor en visitant <http://appinventor.mit.edu/explore/> et clique sur le bouton “Create Apps!” (créer des applications). Sélectionne un compte Gmail pour utiliser (comptes @uottawa.ca fonctionnent). Une fois enregistré, crée et nomme un nouveau projet.
2. Dans le coin en haut à droite il y a une option de langue où tu peux sélectionner Français.
3. Maintenant commence à créer l’interface usager. Pour ajouter un objet à l’application, tire l’icône correspondante du menu à gauche dans l’écran de construction prototype, les objets suivants sont requis pour cette application introductive.

Pour la connectivité Bluetooth:

- Un bouton, appelle-le “Déconnexion”
- Un objet sélectionneur de liste
- Un client Bluetooth régulier et un serveur Bluetooth

Pour contrôle:

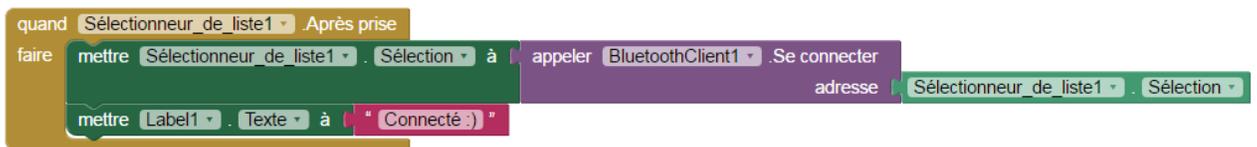
- 4 boutons pour direction
 - Une étiquette (label)
 - Un bouton d’arrêt
4. Utilise la fonction disposition pour arranger les objets de données. Utilise un arrangement tableau pour le contrôle de direction (indice : utilise une table 3x3) et l’arrangement horizontal pour les boutons de connectivité.
 - Note : change la taille des tables (remplir parent) avant d’ajouter les boutons
 5. Change le texte de défaut sur les objets pour refléter leur fonction intentionnelle. Ceci peut être fait dans la fenêtre de droite, une fois qu’un objet (comme un bouton ou étiquette) est sélectionné. Ci-dessous est un exemple d’une interface usager utilisé dans cette application. Montre ton travail au TA.



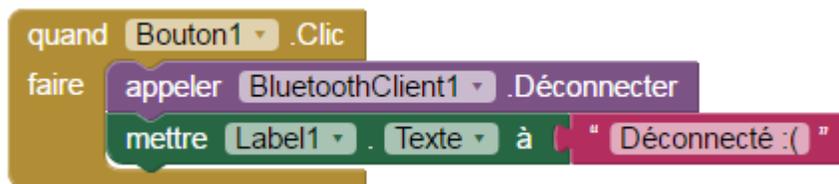
6. Programme la fonctionnalité de l'application. Change à l'onglet "Blocs" en utilisant le bouton dans le coin en haut à droite. Ceci montre l'environnement de programmation utilisé par app inventor.
7. Premièrement programme l'élément sélectionneur de liste. Choisi l'option "quand Sélectionneur_de_liste1.Avant prise" sous l'objet sélectionneur de liste et met le bloc "mettre Sélectionneur_de_liste1.Éléments à " du menu du sélectionneur de liste à l'intérieur. Connecte le bloc "BluetoothClient1.Adresses et noms" (trouvé dans le menu du BluetoothClient) à la fin du bloc sélectionneur de liste, comme montré ci-dessous. Ceci va afficher tous les dispositifs dans un menu déroulant (l'objet sélectionneur de liste).



- Une fois qu'un dispositif a été sélectionné de la liste par l'utilisateur, il doit être connecté au téléphone intelligent. Du menu du sélectionneur de liste, sélectionne "quand Sélectionneur_de_liste1.Après prise". Ce bloc impose qu'est-ce qui va arriver quand un dispositif est choisi de la liste.
- Crée un bloc "mettre Sélectionneur_de_liste1.Sélection à" du menu sélectionneur de liste à la gauche. Du menu "BluetoothClient1", choisis le bloc "appeler BluetoothClient1.Se connecter adresse" et connecte-le au bloc "Sélectionneur_de_liste1.Sélection" du menu sélectionneur de liste. Sélectionne "mettre Label1.Texte à" du menu label1 et ensuite prend texte du menu à gauche, écrit "Connecté :)" et ajoute-le à label. (La photo ci-dessous peut être utilisée comme référence)



- Maintenant sélectionne un bloc "quand Bouton.clic do" d'un des menu de bouton et attache un bloc "appeler BluetoothClient1.Déconnecter". Ceci va déconnecter le téléphone intelligent. Ensuite sélectionne "mettre Label1.Texte à" du menu label et prend texte du menu à gauche, écrit "Déconnecté :(" et ajoute-le à label. (La photo ci-dessous peut être utilisée comme référence)



- En continuant avec les boutons de contrôle, sélectionne "quand Button?.clic faire" (? étant le numéro du bouton) et met "appeler BluetoothClient1.Envoyer texte" du menu bluetooth à l'intérieur. Ajoute un objet texte avec la direction dans laquelle tu veux bouger en cliquant sur le bouton. (La photo ci-dessous peut être utilisée comme référence)



- Pour tester l'application sur un dispositif android, télécharge et enregistre-le comme un fichier .apk (sélectionne "Construire" en haut de la page web). Envoie le fichier par courriel à une adresse que tu as accès sur un téléphone et ouvre le fichier. Tu aurais peut-être besoin de permettre au téléphone d'installer des applications tierces. L'application devrait ensuite s'installer et peut être exécuté comme une application normale quand

terminé. Tu peux aussi demander à MIT App Inventor de montrer un code QR. Scan le code avec le téléphone android (MIT AI2 Companion) et l'application devrait s'installer automatiquement. Montre ton travail au TA.