

MANUEL

Robot RSE sondeur d'environnement

Projet intégrateur
Collège de Bois-de-Boulogne
Session Hiver 2018

Katherine Zamudio-Turcotte,
Arthur Van Betsbrugge
et Charles-Éric Langlois
Sciences informatiques et mathématiques

FRANÇAIS

Table des matières

I.	Matériel requis.....	2
II.	Branchements et construction.....	3
III.	Procédure pour téléverser le code.....	4
	A. Arduino UNO	
	B. Arduino MEGA	
IV.	Ouverture du projet Eclipse (application).....	5
V.	Utilisation.....	6
VI.	Annexe.....	8

I. Matériel requis

Pour construire le robot RSE, il faut s'assurer d'avoir les pièces ci-dessous.

- Structure du robot

1x Kit de robotique de base compatible avec Arduino DFRobotshop Rover V2

1x Plaque d'expansion du DFRobotshop Rover V2

- Composantes électroniques

1x Microcontrôleur Arduino MEGA 2560¹

1x Breakout Board capteur d'humidité et de température Si7021 de Adafruit²

1x Capteur de gaz analogue MQ-2 de DFRobot³

1x Capteur de monoxyde de carbone analogue MQ-7 de DFRobot⁴

1x Capteur de lumière ambiante analogue de DFRobot⁵

1x Module Bluetooth HC-06⁶

1x Capteur à ultrasons HC-SR04⁷

1x Mega sensor shield V2.4 de DFRobot⁸

1x Lumière DEL, préférablement de 10mm

1x Résistance de 400-560 ohms

- Autre matériel

1x Câble USB A mâle à Mini-B (5 pin) mâle

1x Câble USB A mâle à USB B mâle

1x Batterie LiPo de 7.4V et 5A ou batterie 9V, tant qu'elle possède ou soit connectée à une prise à baril de 2,1 mm

4x batteries AA

9x fils de jonction mâle à femelle

¹ *Les articles ci-dessous sont souvent aussi disponibles chez ABRA, Addison, Maddison, Robotshop, Amazon, etc.

<https://store.arduino.cc/arduino-mega-2560-rev3>

² <https://www.adafruit.com/product/3251>

³ <https://www.robotshop.com/ca/en/analog-gas-sensor-mq2.html>

⁴ <https://www.robotshop.com/ca/en/gravity-analog-carbon-monoxide-sensor-mq7.html>

⁵ <https://www.robotshop.com/ca/en/gravity-ambient-light-sensor.html>

⁶ <https://www.robotshop.com/ca/fr/module-bluetooth-hc-06.html>

⁷ <https://www.robotshop.com/ca/fr/module-sonar-hc-sr04.html>

⁸ <https://www.amazon.com/DFRobot-Sensor-Shield-Compatible-Arduino/dp/B00CK5B06Q>

II. Branchements et construction

Première étape : le kit de robotique et le Arduino UNO

1. S'assurer d'avoir en main toutes les composantes du kit de robotique. Assembler le kit et les moteurs Tamiya en position B (voir le manuel d'instruction joint au kit).
2. Connecter les 4 fils provenant des moteurs au Arduino du robot (**image #1**, voir Annexe).
3. Effectuer le circuit sur l'**image #2** en utilisant un breadboard ou en soudant sur une plaque d'essai (voir Annexe). Placer le capteur à ultrasons à l'avant du robot.
4. Repérer l'interrupteur on/off du Arduino. Juste à côté, il y a 2 pins métalliques. Connecter celle se situant plus près de l'extérieur du robot au port digital 13 à l'aide d'un fil de jonction.
5. Mettre les batteries AA dans le battery pack sous le robot. Brancher le fil du pack au Arduino.

Deuxième étape : le Arduino MEGA

6. Visser la plaque d'expansion sur le robot. Replacer et fixer le capteur ultrason pour qu'il soit entre les deux tiges avant de la plaque d'expansion (**image #3**, voir Annexe).
7. Empiler le Mega sensor shield V2.4 sur le Arduino MEGA. Fixer sur l'arrière de la plaque.
8. Fixer les capteurs et le module Bluetooth un peu partout autour sur la plaque.
9. Brancher les 3 capteurs DFRobot dans les ports Analog du sensor shield. Les couleurs (bleu, rouge et noir) sur les fils du câble de chaque capteur indiquent le sens dans lequel les brancher sur les ports Analog.
 - Le capteur MQ-2 se branche sur le port A6.
 - Le capteur MQ-7 se branche sur le port A8.
 - Le capteur de lumière ambiante se branche sur le port A10.
10. Faire le branchement sur l'**image #4** pour connecter le capteur de température et d'humidité au shield (voir Annexe).
11. Repérer le port de communication en série COM1 du sensor shield, juste à gauche de ses prises Analog. Il contient 4 pins : 5V, GND, RX et TX. Brancher chacune des 4 sorties du module Bluetooth à la bonne pin de COM1 :
 - VCC se rend à la pin 5V.
 - GND se rend à la pin GND.
 - TXD se rend à la pin RX.
 - RXD se rend à la pin TX.
12. Brancher à l'aide de 2 fils de jonction, du port COM2 du sensor shield, la pin RX au port TX du UNO, et la pin TX au port RX du UNO.
13. Brancher la prise de baril de 2,1mm de la batterie LiPo ou 9V au Arduino MEGA.

III. Procédure pour téléverser le code

A. Arduino UNO

1. Suivre ce lien, menant vers notre code pour le robot sur Arduino Web Editor :
<https://create.arduino.cc/editor/TheSkyMan/90c00419-ad5b-463b-a7e7-577d1db5655b/preview>
2. Cliquer sur "Add to my sketchbook", puis "Log in".
3. Se connecter sur un compte Arduino.cc existant ou en créer un.
****Attention! Ne pas copier-coller le code sur le Arduino IDE pour téléverser. Cette interface ne reconnaît pas le modèle d'Arduino du robot.**
4. Utiliser le câble USB A à Mini-B (5 pin) pour brancher le robot à l'ordinateur.
5. Mettre l'interrupteur du robot à "OFF".
6. Attendre qu'un message indique qu'un Arduino UNO est connecté.
****Si une bande jaune apparaît et dit qu'il y a un problème avec le "plugin", cliquer dessus et suivre la procédure pour installer le plugin.**
****Si le Arduino n'est toujours pas connecté, consulter le "Gestionnaire de périphériques" de l'ordinateur et aller dans "Ports". Repérer le Arduino UNO, aller dans "Propriétés" et mettre à jour les drivers.**
****Si tout cela ne fonctionne pas (l'ordinateur ne reconnaît pas ce Arduino non conventionnel), changer d'ordinateur et réessayer.**
7. Repérer le COM2 du sensor shield. Débrancher les deux fils de jonction.
8. À gauche du nom du Arduino utilisé, il y a le bouton "Upload". Cliquer dessus pour téléverser le code dans le UNO.
9. Débrancher le robot de l'ordinateur et mettre l'interrupteur à "ON".
Rebrancher les fils au COM2. Il attend maintenant son signal de l'application pour bouger.

B. Arduino MEGA

10. Suivre ce lien et descendre dans la page web pour cliquer sur "Download Adafruit Si7021 Library" :
<https://learn.adafruit.com/adafruit-si7021-temperature-plus-humidity-sensor/arduino-code>
11. Sauver le fichier .zip dans vos dossiers.
12. Retrouver le fichier et le renommer "**Adafruit_Si7021**".
13. Dans le Arduino Web Editor, cliquer dans le menu de gauche sur "Libraries".
Cliquer en haut sur le bouton "Import", à droite du bouton "Library Manager".
14. Retrouver le fichier .zip téléchargé et ouvrir.
15. Suivre ce lien pour ouvrir le code du MEGA :
<https://create.arduino.cc/editor/TheSkyMan/1f4b4a1c-fbc2-4f6a-8f74-9f88833285a0/preview>
16. Répéter l'étape 7.
17. Une fois connecté au même compte Arduino, utiliser le câble USB A à USB B pour téléverser le code dans le MEGA en cliquant sur le bouton "Upload".
Rebrancher les fils au COM2.

IV. Ouverture du projet Eclipse (application)

1. S'assurer d'avoir une copie .zip de l'application dans vos dossiers. Ouvrir ce fichier et sortir le dossier à l'intérieur du .zip.
2. ~~Suivre ce lien pour télécharger la librairie Bluecove 2.1.1, en format .jar, dans vos dossiers :~~ https://drive.google.com/open?id=1pjltxdEsU9ZIZhq4u_6mkXcObHvrzgHv
Si l'application n'est pas déjà dans vos dossiers, télécharger ici : https://github.com/KatherineZT/ProjetRSEApplication_PL_-q1_gr3
3. Si l'ordinateur utilisé ne possède pas Eclipse Java Oxygen (Windows 64-bit), le télécharger en suivant ce lien et en cliquant ensuite sur "Download" :
<https://www.eclipse.org/downloads/download.php?file=/oomph/epo/oxygen/R2/eclipse-inst-win64.exe> <https://www.eclipse.org/downloads/>
****Dans l'installateur pour Eclipse, sélectionner "Eclipse IDE for Java developers" et poursuivre les instructions.**
4. Ouvrir Eclipse Java Oxygen.
5. En haut à gauche, cliquer sur "File", puis "Open Projects from File System...".
6. À droite de "Import Source:", cliquer sur "Directory...". Retrouver le dossier sorti du .zip (à l'étape 1) et cliquer sur "OK".
7. Appuyer sur "Finish".
8. ~~Ouvrir au moins une classe de l'application pour en voir le code (regarder dans le "Package Explorer" à gauche de l'écran pour naviguer dans le projet Eclipse).~~
Le projet sera caché par le Welcome menu, il faut le fermer une fois que le projet est importé, s'il n'apparaît pas de lui-même à l'écran.
9. Faire un right-click sur le nom du projet dans le "Package Explorer" et passer la souris sur "Build Path". Cliquer sur "Configure Build Path..." dans le menu déroulant à droite. Cliquer sur l'onglet "Libraries".
10. Cliquer sur le fichier .jar "**bluecove-2.1.1.jar** - (etc...)" et cliquer à droite sur "Edit...".
Si aucun fichier .jar du nom de "bluecove-2.1.1.jar - ..." n'est dans la liste, cliquer seulement sur "Add External JARs" et non sur "Edit...".
-> qui est dans le dossier dézippé de l'application
Sélectionner le .jar de la librairie Bluecove de vos dossiers et cliquer sur "Open".
Cliquer ensuite en bas sur "Apply and Close".
11. S'assurer que le module Bluetooth sur le robot clignote rouge. Si la lumière du module ne clignote pas ou si elle n'est tout simplement pas allumée, changer la batterie LiPo ou 9V.
12. S'assurer que des petites lumières DEL vertes ou jaunes s'illuminent sur le Arduino UNO. Sinon, changer les piles AA.
13. Sur Eclipse, appuyer sur "Run", l'icône constituée d'un cercle vert et d'une flèche blanche sur la bande d'icônes en haut de l'écran, ou appuyer sur Ctrl+F11.
14. L'application ouvrira. Le module Bluetooth devrait arrêter de clignoter et les valeurs devraient s'afficher sur les cadrans.

***À noter que l'application ne fonctionne pas très bien lorsque le robot n'est pas allumé ou trop loin.**

V. Utilisation

L'application ouvre sur 5 **cadrons**. Chacun représente une information sur l'environnement captée par le Arduino MEGA : l'humidité, la température, la luminosité ambiante, la concentration en gaz combustibles (LPG, butane, propane, méthane, alcool, hydrogène, fumée) et celle en monoxyde de carbone.

Au centre de chacun, la valeur captée à l'instant par le capteur est affichée. À droite, 3 petits cadrons indiquent les valeurs maximale, minimale et moyenne des données recueillies jusqu'à présent. Les arcs colorés indiquent visuellement la taille de la valeur au centre de leur cadran. Sa couleur indique le niveau de danger associé à cette valeur :

Vert → peu ou pas de danger

Jaune → danger moyen

Rouge → danger important

Par exemple, une luminosité trop basse ou une température un peu trop chaude sera catégorisée dans le jaune, mais une concentration mortelle de CO sera catégorisée dans le rouge.

Au bas de chaque cadran se trouve l'indication "**Détails**", qui mène, si l'on clique dessus, au **graphique** de l'évolution des valeurs du cadran selon le temps. Le graphique prendra la même couleur que le cadran à l'instant où il est affiché. Le graphique n'est pas gradué sur ses axes, mais la valeur (de température, humidité ou autre) associée à la position de la souris sur l'axe des x du graphique est toujours affichée à droite du curseur. Le graphique ne s'ajuste pas aux nouvelles valeurs reçues une fois généré. Pour le rafraîchir, il faut fermer la fenêtre et recliquer sur "Détails".

Pour recommencer la prise de données, il suffit de sélectionner "Fichier" et "**Relancer**" dans la barre de menus en haut de l'écran. Pour quitter l'application et cesser par le fait même le flux de communication via bluetooth, le sous-menu "**Quitter**" est juste en-dessous.

Dans le menu "Informations", le sous-menu "**État du robot**" permet de savoir si le robot est en mouvement ou non et s'il est sur son chemin de l'allée ou du retour.

Dans le menu "Aide", "**Utilisation**" rappelle brièvement les commandes expliquées ici et "**À propos**" permet d'en savoir plus sur le robot et le module bluetooth utilisé avec l'application ainsi que le contexte de la création de celle-ci.

Si l'on revient sur le panneau principal, en-dessous du logo du robot, il y a 3 **boutons**. Au moment où on ouvre l'application, aucun n'est enclenché et le robot est immobile au sol. Le Arduino MEGA prend les données de ses capteurs et les envoie via bluetooth à l'application via sa première communication en série (le COM1), tandis que sa deuxième communication (celle qui le relie physiquement au Arduino UNO, le COM2) est inutilisée.

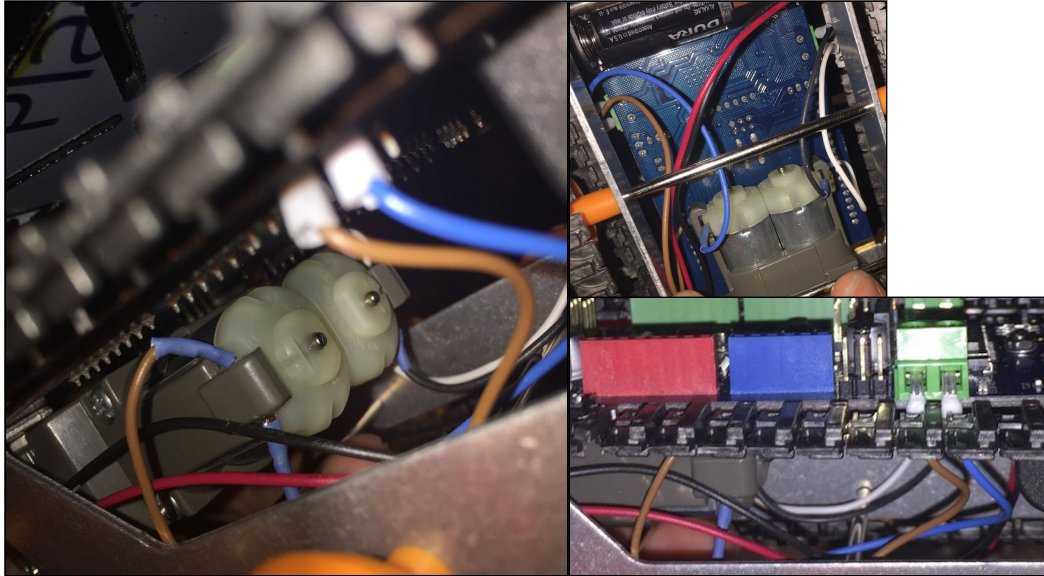
“Partir” permet à l’application d’envoyer via bluetooth une commande au MEGA via le COM1, qu’il envoie ensuite au UNO par le COM2. Cette commande est la permission au robot pour avancer et commencer à éviter les obstacles qu’il rencontre.

“Revenir” permet à l’application d’envoyer une autre commande au MEGA, qui dit cette fois au robot de revenir sur ses pas jusqu’à son point de départ. Les boutons doivent être appuyés dans leur ordre logique et ne doivent pas être réappuyés avant que le robot ne soit revenu à son point de départ. Sinon, un message d’erreur est affiché.

Le bouton “Délai” permet à l’utilisateur de rebrancher le module bluetooth sur le robot sans bousculer la prise de données. Il n’a qu’à l’appuyer avant de débrancher le module pour suspendre la communication bluetooth. Une fois le module rebranché, il appuie sur “OK” et l’application se met à attendre que le flux de données soit à nouveau stable.

VI. Annexe

Image #1 :



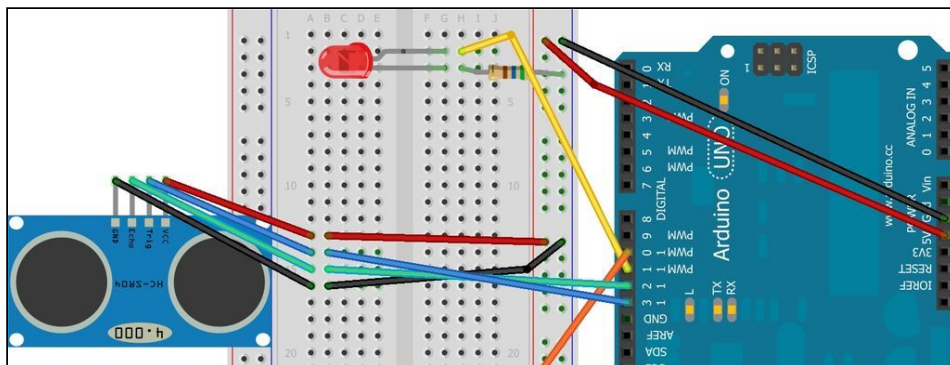
Notes :

*On voit ici que le fil **brun** provient de la connection **supérieure** du moteur **droit** ("droit" ici étant utilisé comme l'Est si le devant du robot est au Nord). Il faut le brancher dans le trou le plus proche de lui, soit le trou de **gauche** de la borne **verte** qui se situe sur le côté **droit** du robot.

*Le fil bleu, provenant de la connection inférieure du moteur droit, est branché dans le trou le plus loin de lui, dans la même borne verte.

*De l'autre côté, on procède de la même façon : fil supérieur → trou le plus proche, fil inférieur → trou le plus loin.

Image #2 :



Notes (voir page suivante aussi) :

*Ignorer le fil orange.

*Sur une plaquette d'essai pour la soudure, souder les fils rouge, noir, vert et jaune, le capteur, la lumière DEL et la résistance. OU reproduire les branchements sur un breadboard.

*Le fil bleu doit être composé de **deux** fils de jonction (doit être long).

*Connecter aux bons ports du Arduino du robot. À noter que le port digital 11 devient le port digital **2**, 12 devient **3** et 13 devient **4**.

*Le fil rouge peut être composé d'un seul fil (pas divisé en deux comme sur le schéma).

Image #3 :

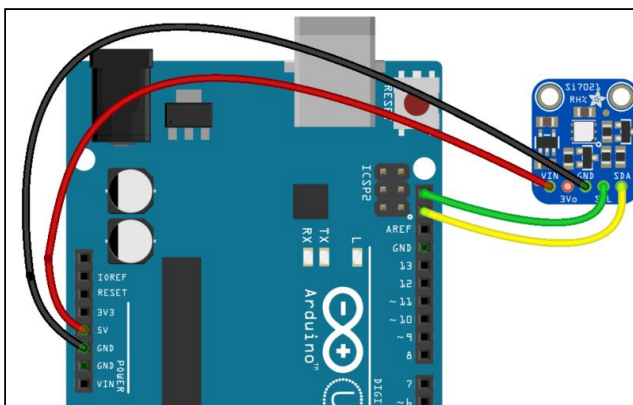


Notes :

*L'important est que rien ne se trouve devant les deux "yeux" du capteur (rien ne bloque les ultrasons envoyés et reçus) et que la lumière DEL soit visible.

*Si vous avez utilisé un breadboard pour le circuit, il se peut que le capteur ne rentre pas à cet endroit.

Image #4 :



Notes :

*Le fil vert se branche dans le port SCL du shield et le fil jaune, dans son port SDA.