Pilotage d'un véhicule par Bluetooth

Ce dossier pédagogique comprend 3 parties :

1 – Triporteur Arduino B4R - Construction du véhicule que nous nommerons désormais « triporteur » et développement du programme Arduino B4R qui permettra de faire rouler ce triporteur. (Pages 1 à 11)

2 – Pilote Bluetooth B4A - Développement de l'application B4A pour Smartphone Android qui permettra de piloter le triporteur à distance via le système Bluetooth. (Pages 12 à 14)

3- Joystick Shield Arduino B4R - Construction d'une télécommande ARDUINO permettant de piloter le triporteur à distance toujours par Bluetooth, étude de ce deuxième programme Arduino B4R. (Pages 14 à 21)

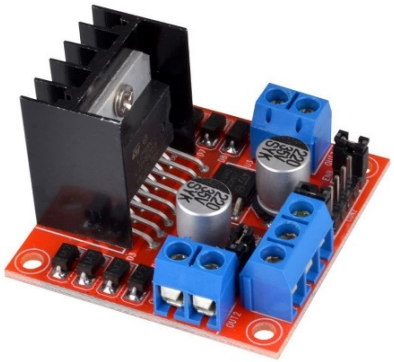
---

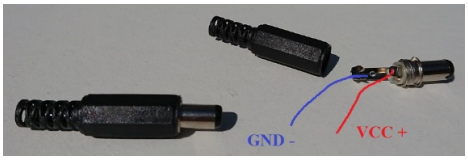
1 – Construction du triporteur Arduino

Matériel nécessaire :

- Une carte Arduino Uno
- Un quart de plaque de connexions rapides (« Breadboard »)
- Un module Bluetooth HC-05
- Deux batteries rechargeables Li-ION et leur boîte de connexion avec couvercle
- Un module L298N Shield permettant la connexion et la gestion des deux moteurs
- Un support ou châssis en PVC avec 2 moteurs DC et 2 roues adaptables, une roulette pivotante arrière, un petit interrupteur central
- Une fiche mâle 5,5 X 2,5 destinée à l'alimentation de la carte Arduino Uno
- Un Smartphone Android disposant de la technologie Bluetooth

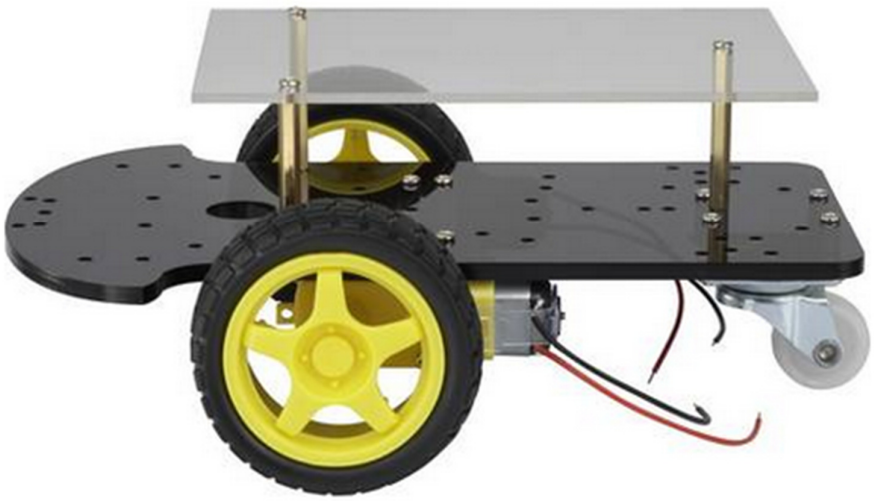






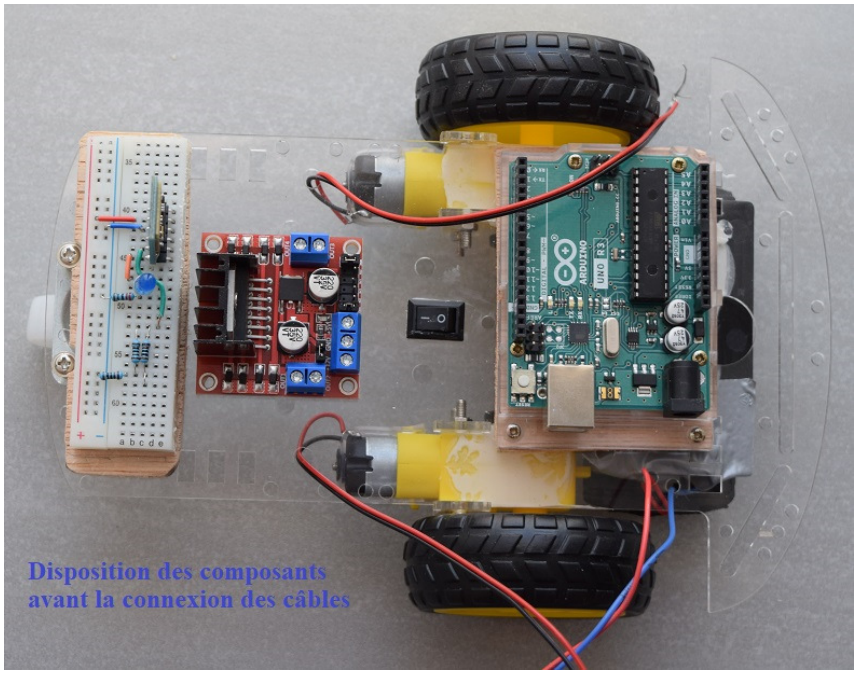
Support-socle choisi

Support-socle une fois assemblé

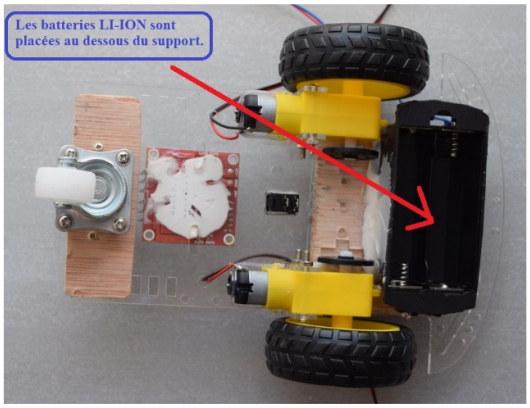


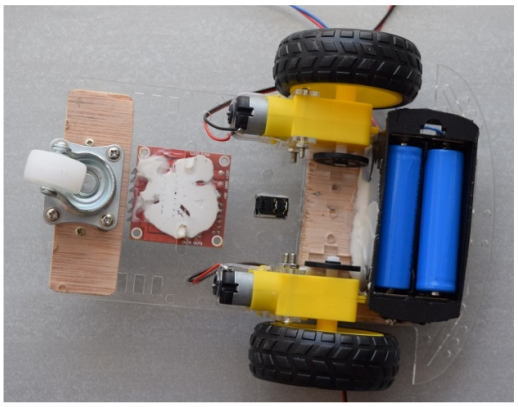
Support alternatif disponible dans le commerce en ligne

L'ensemble est plus résistant et les moteurs sont précâblés à l'achat…

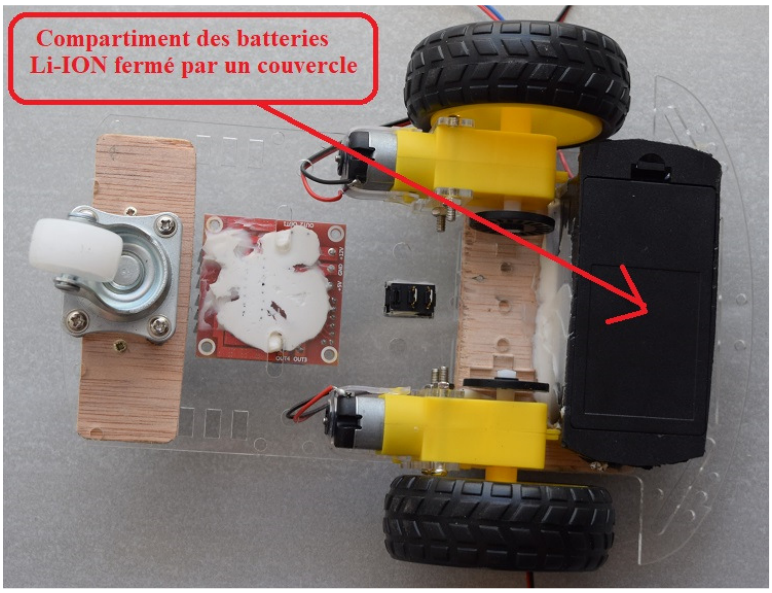


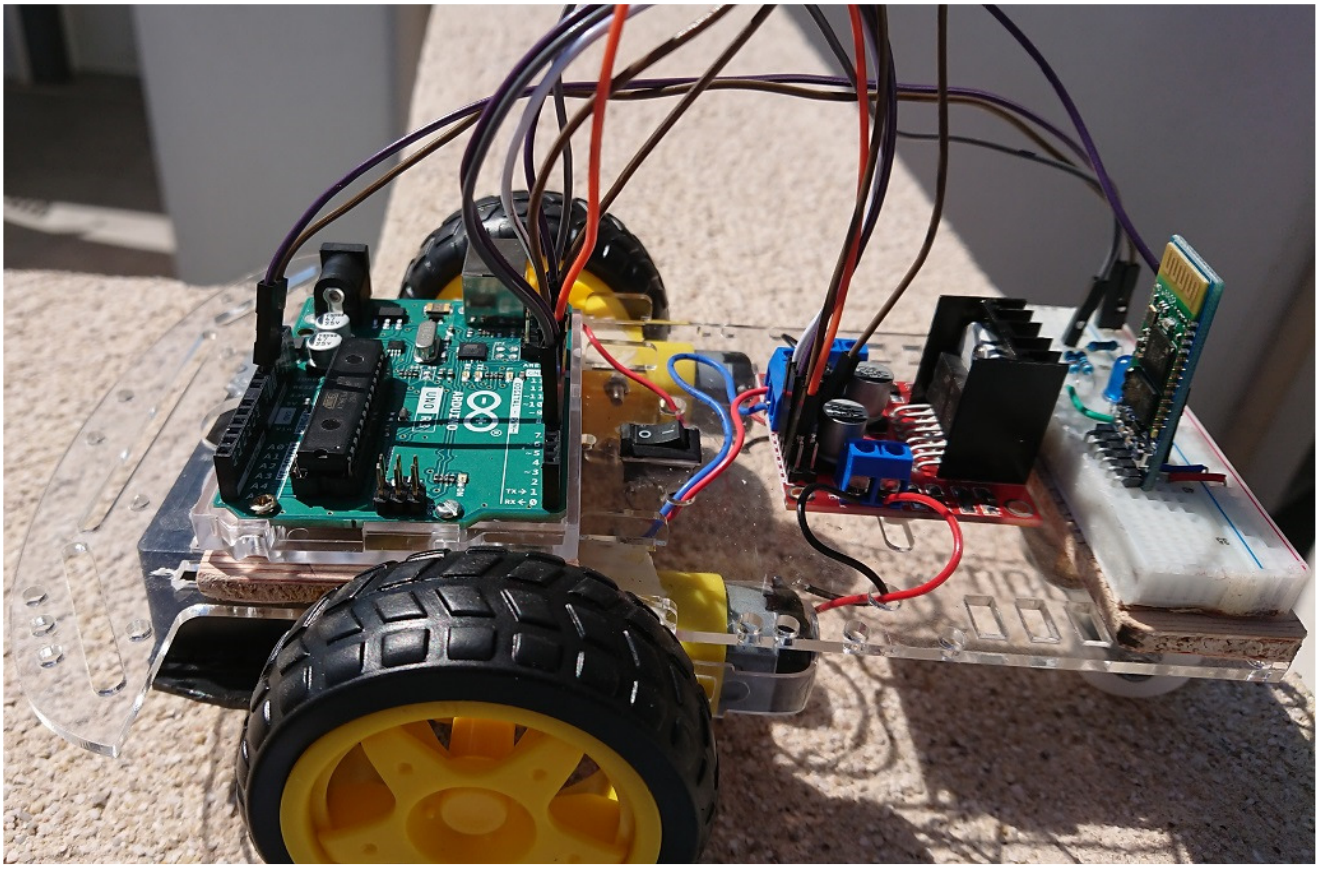
Nous plaçons la carte Arduino Uno à l'avant du triporteur, vissée sur une petite plaque de contre-plaqué elle-même vissée par en dessous du châssis, l'interrupteur central a son emplacement prêt, nous plaçons le module L298N au centre et le quart de plaque de connexions rapides à l'arrière avec le module Bluetooth HC-05.





Nous plaçons la boîte de connexion des 2 batteries rechargeables Li-ION de 3,7 volts chacune en dessous du châssis du triporteur. Ces batteries sont des modèles 18650. La tension de sortie est donc de 7,4 volts. Les batteries sont directement connectées d'une part au module L298N et d'autre part à la carte Arduino Uno, via le petit interrupteur central du triporteur.





Connexion des différents éléments entre eux

Le triporteur est pratiquement opérationnel, il reste quelques détails à respecter au niveau des connexions des différents modules en respectant le schéma général des connexions page suivante.

Schéma des connexions du TRIPORTEUR ARDUINO

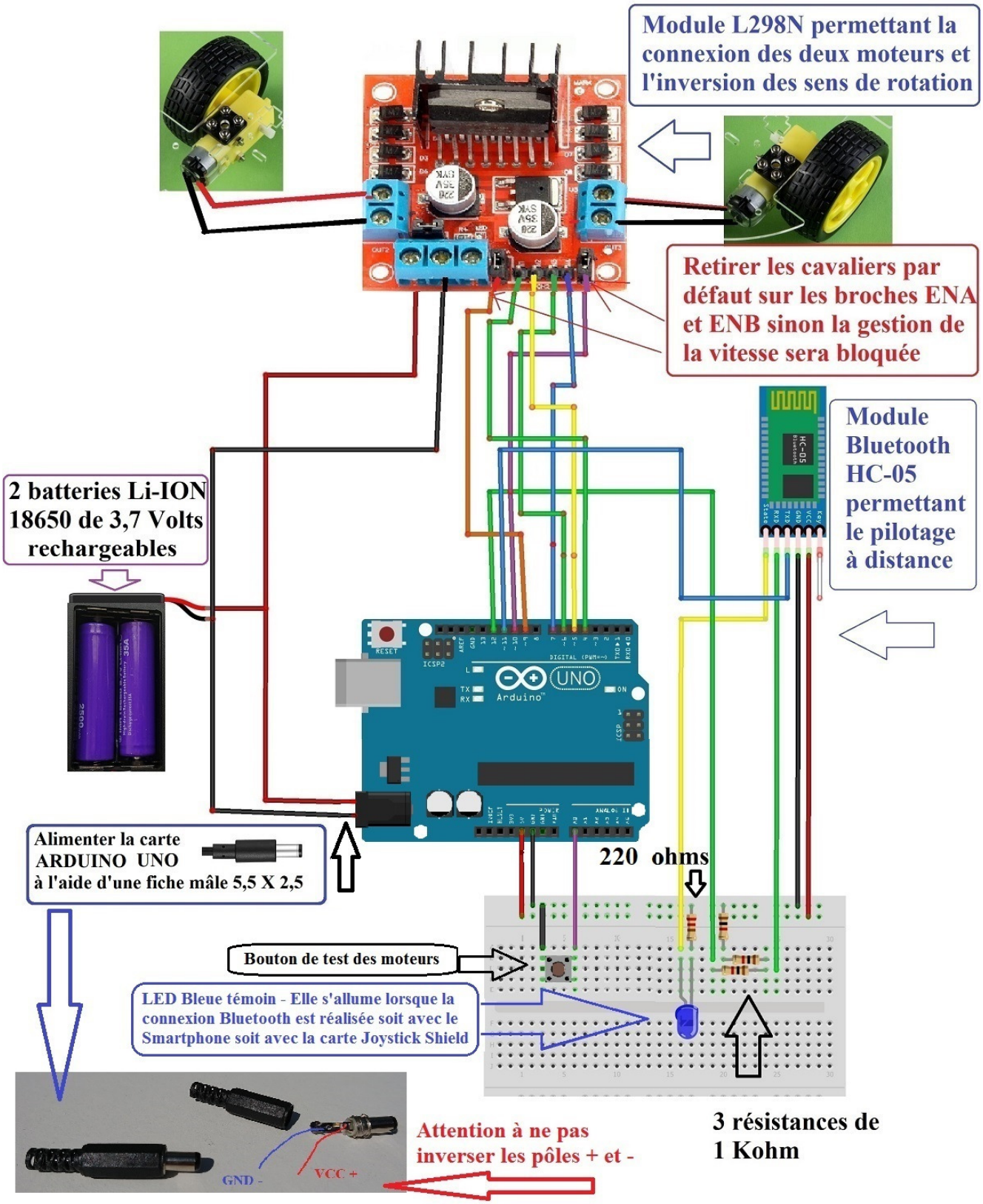
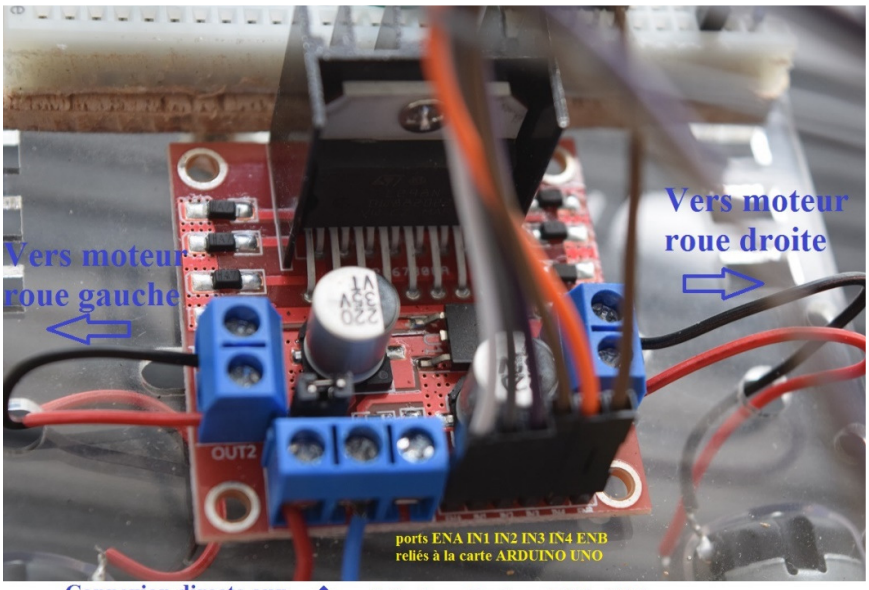
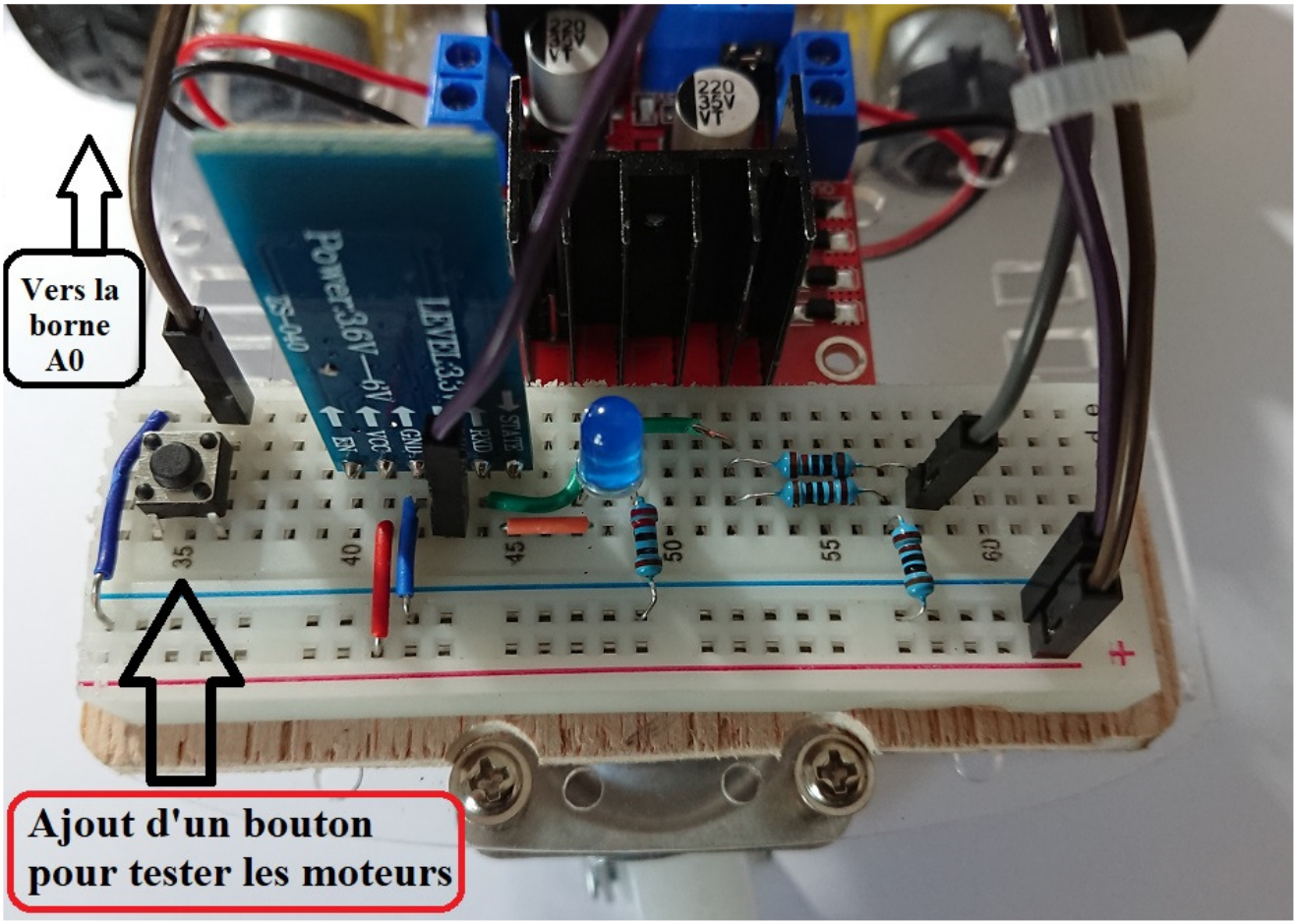


Tableau général des connexions du triporteur ARDUINO

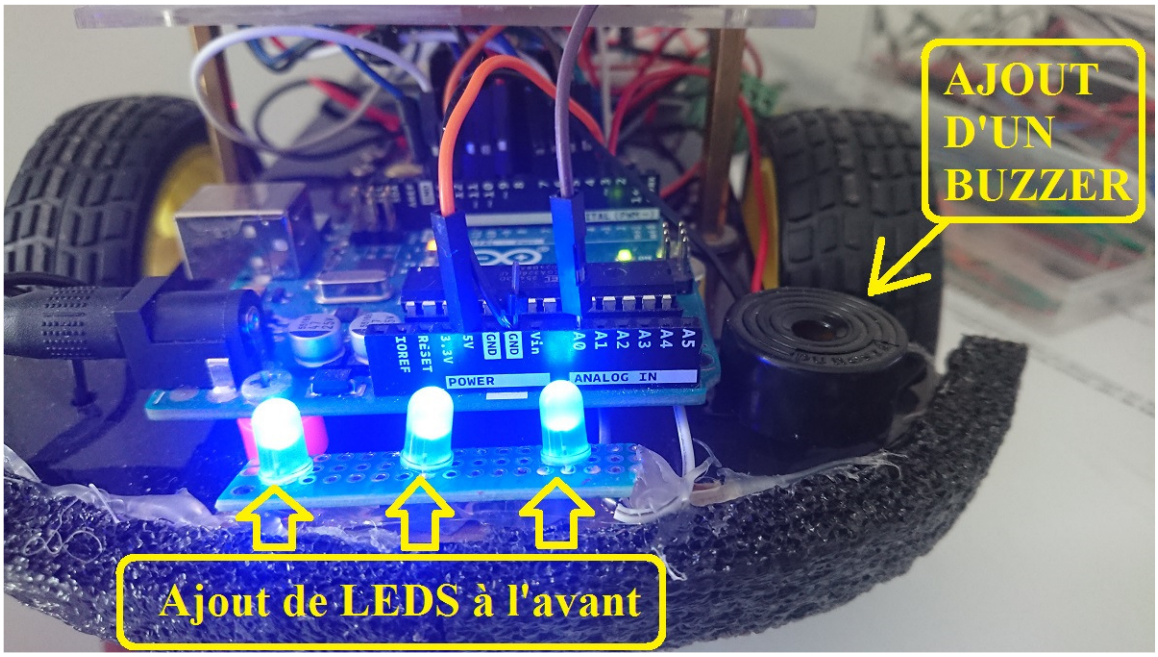


Connexions du module L298N

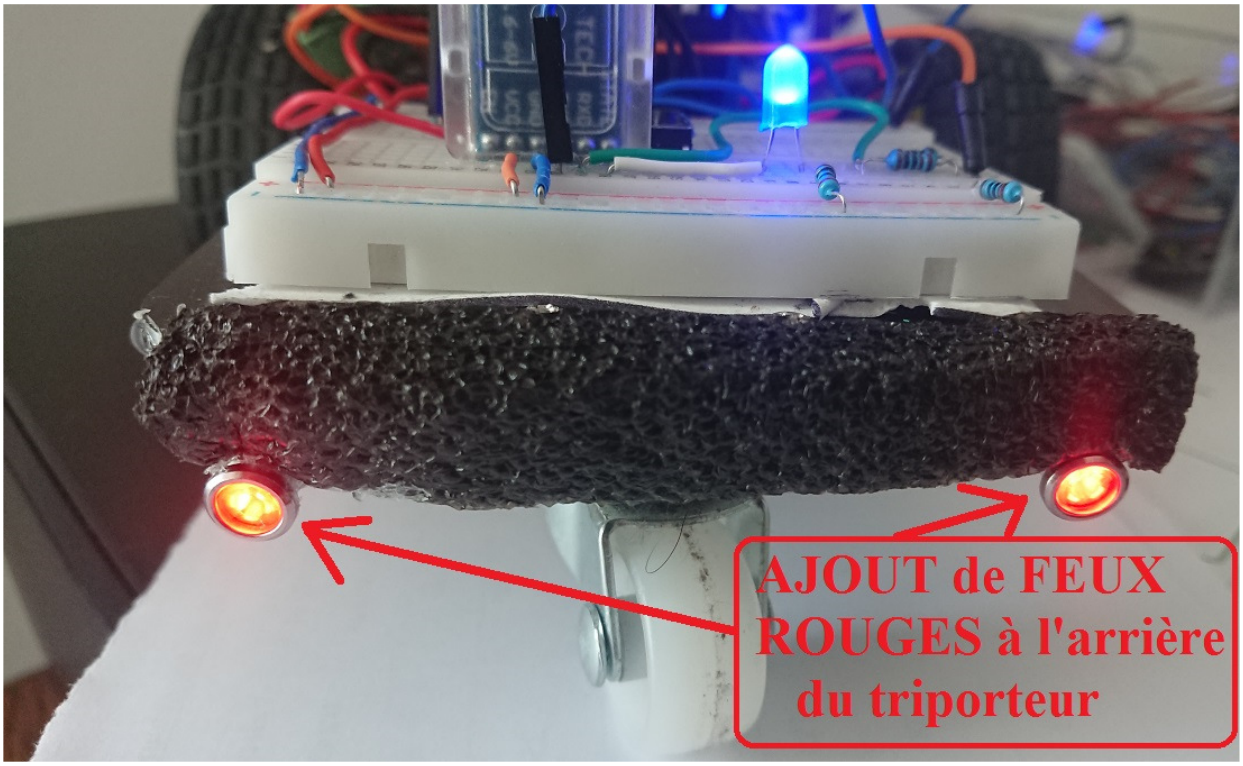
Avant de connecter les ports ENA et ENB aux broches 9 et 10 de la carte ARDUINO UNO, vous devez impérativement ôter les cavaliers installés par défaut sur ces ports car ils bloqueraient la gestion de la vitesse des moteurs.



Nous rajoutons sur la mini-plaque arrière un petit bouton-poussoir qui servira à tester le fonctionnement des moteurs, même hors connexion Bluetooth. Ce bouton est relié à la broche anlogique A0.



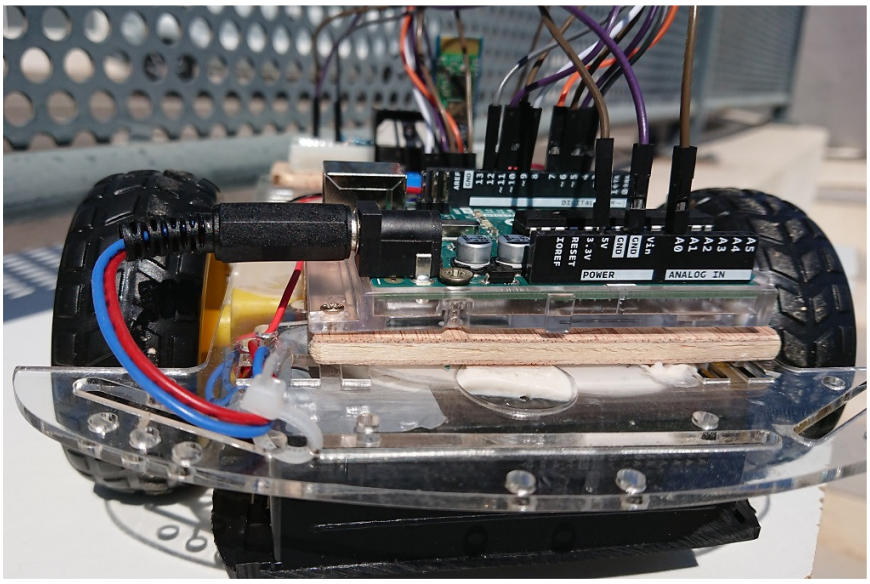
Nous rajoutons des LEDs (bleues ou blanches) à l'avant du triporteur ainsi qu'un buzzer. Nous commanderons à distance ces équipements à partir du Smartphone ou de la télécommande Arduino.



Nous ajoutons également deux diodes rouges à l'arrière du triporteur. Les LEDS avant et les feux rouges devront être connectés (pôle positif +) à la broche digitale n° 13. Protégez éventuellement les LEDs par une résistance de 220 ohms placée sur le pôle négatif (– GND).

Le buzzer devra être connecté (fil rouge +) à la broche digitale n° 3 et à un port négatif GND disponible (fil noir) de la carte Arduino Uno.

Ces équipements ne sont, bien sûr, pas indispensables et leur absence ne perturbera pas les programmes B4R et B4A. (Voir plus loin le poste de pilotage Bluetooth du Smartphone Android ou les boutons de la télécommande Arduino « Joystick Shield ».)



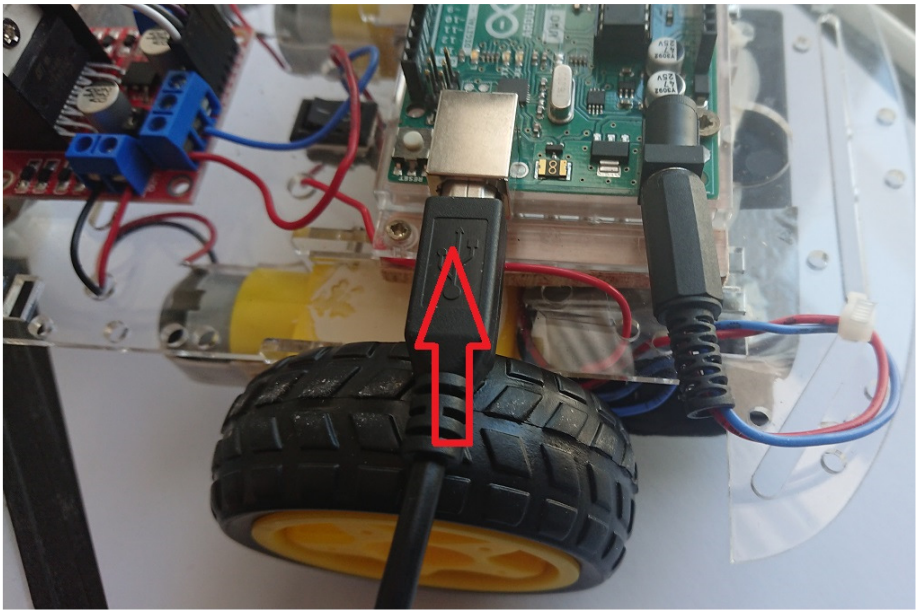
Connexions de la carte ARDUINO UNO placée à l'avant du triporteur

Téléchargez le programme B4R : «TriporteurARDUINO» - Téléchargements page 31

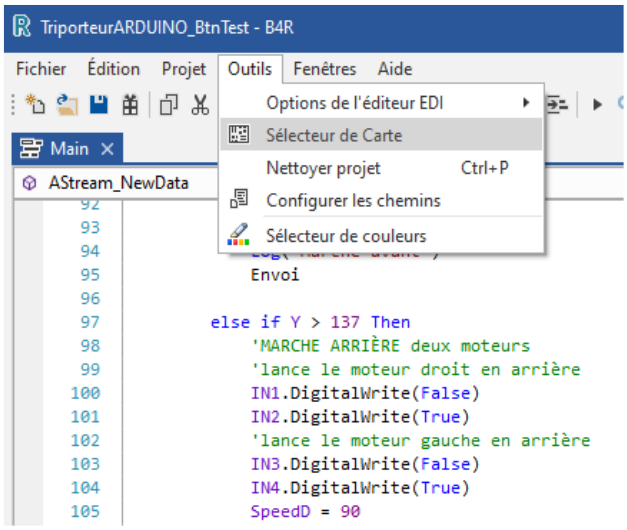
Si vous l'avez déjà installé, démarrez B4R sinon vous le trouverez ici >>> https://www.b4x.com/b4r.html

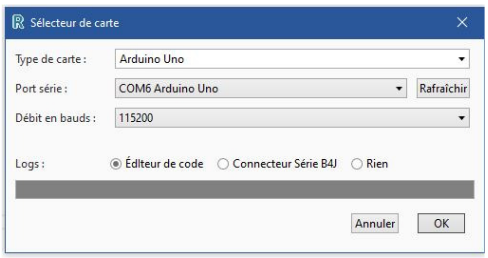
Installation des logiciels nécessaires ici >>> https://www.b4x.com/b4r.html#installation

- Lorsque B4R est installé sur votre ordinateur, ouvrez le programme TriporteurARDUINO.B4R ».
- Connectez le triporteur à votre PC à l'aide d'un câble USB approprié :



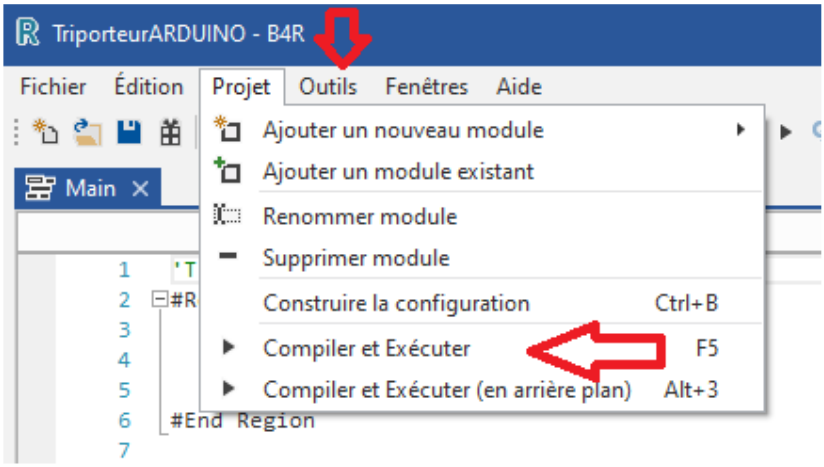
- Sélectionnez la carte ARDUINO UNO et son port COM adéquat > Outils > Sélecteur de carte





Notez que selon les cartes Arduino, le n° du port peut varier (COM3, COM5, COM6 …)

- Procédez au « téléversement » du programme sur le triporteur : Projet > Compiler et Exécuter



Une fois le programme chargé ou plus exactement « téléversé » sur la carte ARDUINO UNO du triporteur, celui-ci est opérationnel et vous pouvez allumer l'interrupteur central et tester les moteurs avec le petit bouton-poussoir.

Si vous constatez des dysfonctionnements du système, notamment lors du test de fonctionnement des moteurs, pas de panique. Si une roue ou les 2 roues recule(nt) au lieu d'avancer, inversez les connexions + et - aux sorties OUT1 (et/ou OUT2) de l'alimentation des moteurs. Par la suite, lors du pilotage Bluetooth, si le triporteur tourne à gauche au lieu de tourner à droite et inversement, permutez simplement la connexion des câbles en provenance du module L298N (broches ENA et ENB) au niveau des bornes 9 et 10 de la carte ARDUINO UNO. Tout devrait rentrer dans l'ordre...

Le programme « Triporteur Bluetooth B4R » détaillé est disponible à la fin de cette brochure pages 22 à 24.

Ce programme développé sous Visual BASIC pour cartes ARDUINO (B4R) peut être adapté au matériel utilisé, certains **conseils ou explications en caractères verts** vous sont donnés en ce sens…

Si le triporteur fonctionne correctement, nous allons pouvoir passer à l'étape suivante : le développement de l'application Android qui va gérer le pilotage par Bluetooth de notre triporteur.

2 – Pilote Bluetooth B4A – Application pour appareil Android

Nous allons à présent nous consacrer au développement de l'application Android permettant de transformer votre Smartphone ou autre dispositif Android en poste de pilotage du triporteur précédemment réalisé.

Cette application sera développée en Visual BASIC pour Android (B4A)

Captures d'écran successives de l'écran du Smartphone utilisé

Avant d'utiliser cette application destinée à détecter le module Bluetooth du triporteur, puis à se connecter à lui, vous devez bien entendu allumer le triporteur afin que le signal Bluetooth émis par le module HC-05 soit détectable par votre Smartphone.

Ceci se traduit par le clignotement rapide et continu de la petite LED rouge du module HC-05. Vous devez également vérifier dans les paramètres de connexion de votre Smartphone que le système Bluetooth est activé.

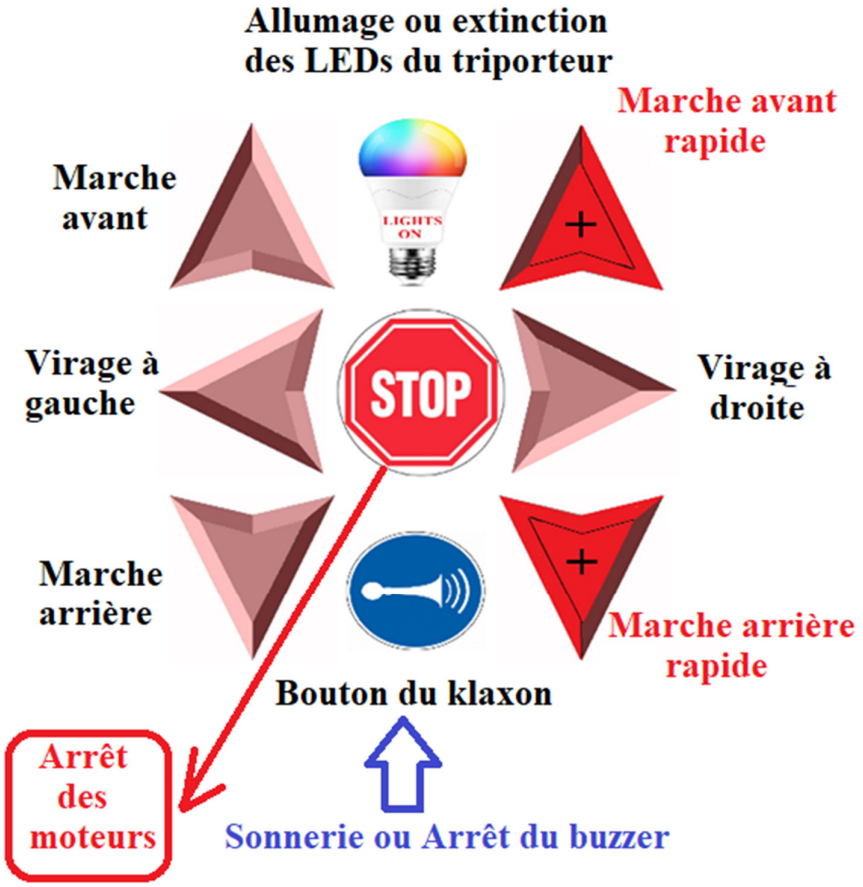
L'écran d'accueil du Smartphone vous invite à cliquer sur la touche verte « Connexion » puis la mention « Recherche dispositif en cours » s'affiche.

Si, au bout d'un certain temps, le Smartphone ne détecte pas un module Bluetooth HC-05 allumé, un message vous invite à vérifier l'alimentation du triporteur et donc du module HC-05 dont la petite LED rouge doit clignoter rapidement d'une façon continue.

Lorsque le module HC-05 est détecté par le système Bluetooth de votre Smartphone, la mention « HC-05 détecté, veuillez patienter » s'affiche puis, quelques secondes plus tard, la connexion est effective et se vérifie de deux façons :

1- La petite LED rouge du module HC-05 installé sur le triporteur se met à clignoter 2 fois rapidement avec une pause de quelques secondes avant le double clignotement suivant et la grosse LED bleue témoin s'allume de façon fixe.
2- Sur l'écran du Smartphone, la mention « Module Bluetooth HC-05 connecté » s'affiche et le logo bleu Bluetooth apparaît ainsi que les touches de commande du poste de pilotage du triporteur.

A partir de ce moment, vous pouvez piloter le triporteur à l'aide de votre Smartphone en utilisant les touches appropriées :



Touches disponibles pour le pilotage du triporteur

Touches utilisées		Résultat attendu	Valeurs de X Transmises par le Smartphone	Valeurs de Y Transmises par le Smartphone	Vitesses du triporteur
	↑ AVANT ↑	Marche avant	127	80	90
	↑ AVANT ↑	Marche avant plus rapide	250	250	120
	DROITE →	Rotation à droite	180	63	55 (roue D) 90 (roue G)
	↓ ARRIÈRE ↓	Marche arrière	127	180	90
	↓ ARRIÈRE ↓	Marche arrière plus rapide	200	200	120
	← GAUCHE	Rotation à gauche	63	63	90 (roue D) 55 (roue G)
	ou aucune touche	**Arrêt des moteurs**	127	127	0
	LIGHTS ON	Allumage (ou extinction) des LEDs si installées	249	249	0
	Klaxon	Sonnerie ou arrêt du buzzer si installé	199	199	0

Valeurs de X et Y transmises par le Smartphone et vitesses correspondantes sur le triporteur

Téléchargez l'application Android «Pilote Bluetooth» page 31

Pour ouvrir le programme « Pilote Bluetooth » développé en Visual BASIC pour Android (B4A), il faut disposer de l'environnement B4A disponible ici >>> https://www.b4x.com/b4a.html

Si vous avez déjà installé B4A sur votre PC, vous pouvez ouvrir le programme « Pilote Bluetooth » et éventuellement lui apporter des modifications avant de le recompiler sur votre Smartphone.

Vous avez également la possibilité d'installer directement le fichier APK (Android Pack Kit) sur votre Smartphone sans modification : **Téléchargez directement le fichier APK page 31**

L'application « Pilote Triporteur » développée en Visual BASIC pour appareils Android (B4A) est composée de trois modules : Module principal (Main), Module BluetoothAsynchStream, Module Starter et aussi la fenêtre graphique (« Designer ») permettant la mise en page des éléments graphiques sur l'écran du Smartphone.

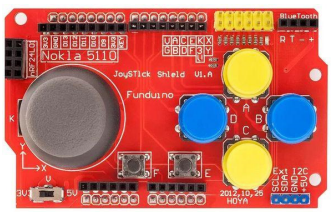
Vous trouverez le programme Pilote Triporteur en 3 modules détaillés à la fin de cette brochure aux pages 25 à 29.

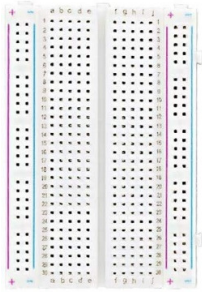
3- Joystick Shield Arduino B4R

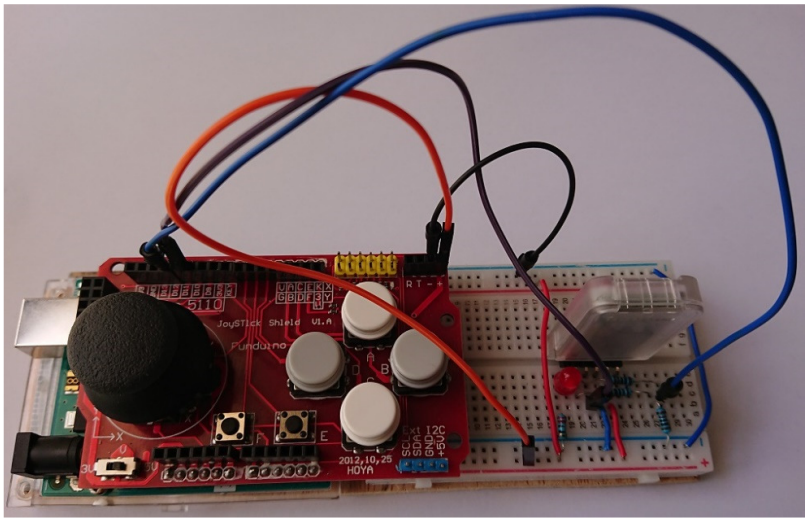
Construction d'une télécommande alternative pour piloter le triporteur

Notez-bien que cette option est inutile si vous utilisez un Smartphone Android disposant de la technologie Bluetooth…

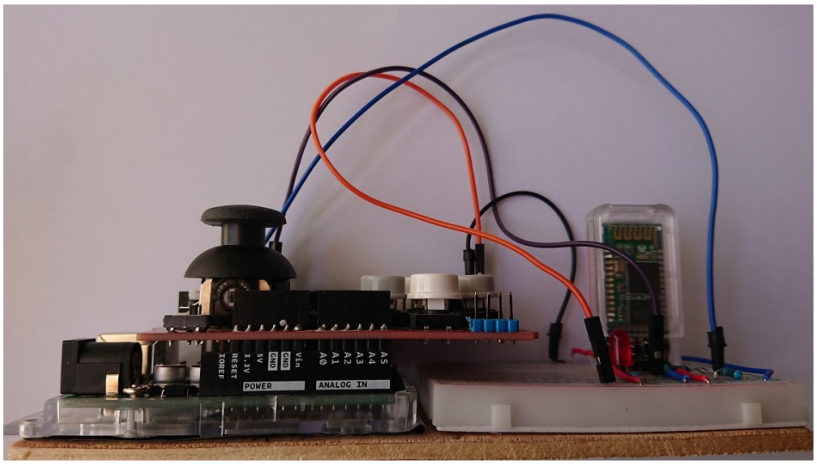
Matériel nécessaire: Une seconde carte ARDUINO UNO, une carte Joystick Shield clipée sur la carte Arduino Uno, un second module Bluetooth HC-05 que l'on va rendre "maître" et qui transmettra les valeurs à l'autre module Bluetooth installé sur le triporteur qui est déjà programmé comme "esclave", une demi plaque de connexions, câbles et jumpers pour les connexions, 3 résistances de 1 Kohm, une résistance de 220 ohms, une LED pour visualiser la connexion Bluetooth avec le triporteur.







« Télécommande » Arduino avec module Bluetooth HC-05



Nous allons utiliser les boutons A, B, C, D, E, F et G pour piloter à distance notre triporteur grâce à un deuxième module Bluetooth HC-05.
En plus des 4 boutons A, B, C et D utilisés pour la marche avant, le virage à droite, la marche arrière et le virage à gauche, le gros bouton G du joystick permettra désormais d'obtenir la marche avant rapide. Nous utiliserons les deux petits boutons E et F pour commander les feux (LEDs, diodes lumineuses, etc …) et le klaxon (buzzer).

Bouton A > BtnA > broche D2 (marche avant)
Bouton B > BtnB > broche D3 (rotation à droite)
Bouton C > BtnC > broche D4 (marche arrière)
Bouton D > BtnD > broche D5 (rotation à gauche)

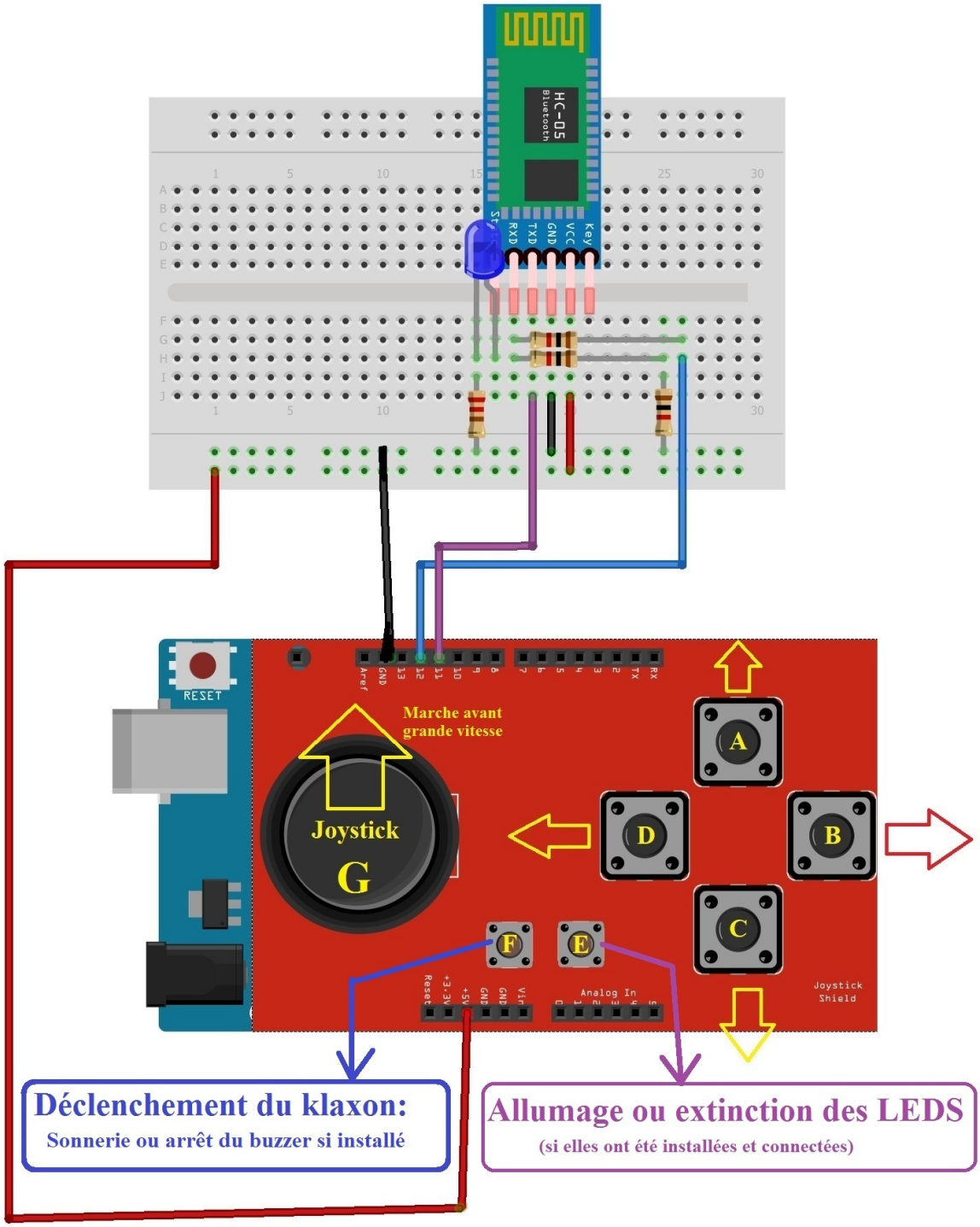
Petit Bouton E > BtnE > broche D6 (Allumage ou extinction des diodes lumineuses)
Petit Bouton F > BtnF > broche D7 (Coups de klaxon)

Gros bouton G du joystick > BTJ > broche D8 (marche avant rapide)

Notez bien que la grosse différence de fonctionnement de cette télécommande à bouton-poussoir par rapport aux touches tactiles du Smartphone est que l'envoi des valeurs X et Y ne

se fait que si le ou les boutons restent pressés. Dès que vous relâchez un bouton, le triporteur s'arrête, cela équivaut à la touche STOP du Smartphone. Au contraire, lorsque vous touchez une touche du Smartphone, les valeurs X et Y sont envoyées en continu jusqu'à ce que vous changiez de touche ou que vous touchiez la touche STOP…

Boutons utilisés	Résultat attendu	Valeur de X	Valeur de Y	**Vitesses**
↑ A ↑	Marche avant	127	80	**90**
↑ G ↑ Gros bouton du joystick	Marche avant rapide	250	250	**120**
→ B →	Rotation à droite	180	63	**90 (roue D) 55 (roue G)**
↓ C ↓	Marche arrière	127	180	**90**
← D ←	Rotation à gauche	63	63	**55 (roue D) 90 (roue G)**
Aucun bouton pressé	**Arrêt des moteurs**	127	127	**0**
E	Allumage ou extinction des LEDS	249	**249**	**0**
F	Sonnerie ou arrêt du buzzer si installé	199	**199**	**0**



NB - La carte "Joystick Shield" étant "clipée" sur la carte Arduino Uno, les connexions des boutons que l'on utilise sont automatiquement réalisées avec les broches de la carte Arduino Uno.

Schéma des connexions de la télécommande Arduino

NB - Il y a un port Bluetooth en haut et à droite de la carte Joystick Shield mais il ne permet pas de transmettre correctement les données . La broche RX du module Bluetooth HC-05, protégée par 3 résistances de 1 Kohms est reliée à la broche D12 de la carte Shield (et donc de la carte Arduino Uno) et la broche TX est reliée à la broche D11.

Modifications à apporter au second module Bluetooth HC-05 pour qu'il devienne maître: "Master"

Avant de pouvoir utiliser la télécommande Arduino, il faut apporter des modifications à ce second module Bluetooth HC-05 et pour cela, il faut le connecter avant tout montage, directement à la carte Arduino Uno en respectant le tableau de connexions ci-dessous. Vous pouvez effectuer cette opération dans l'environnement ARDUINO >>> https://www.b4x.com/b4r.html#installation

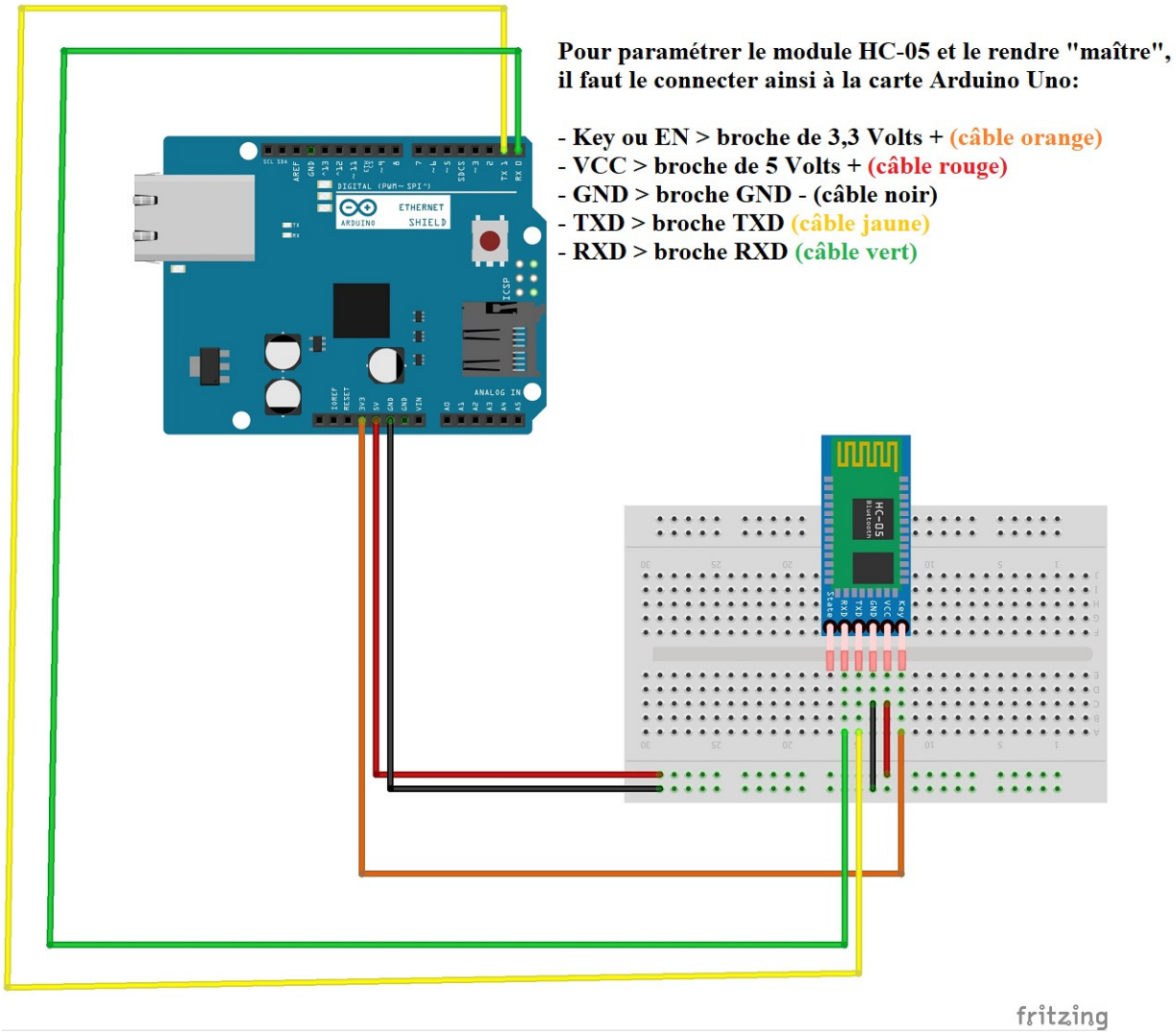
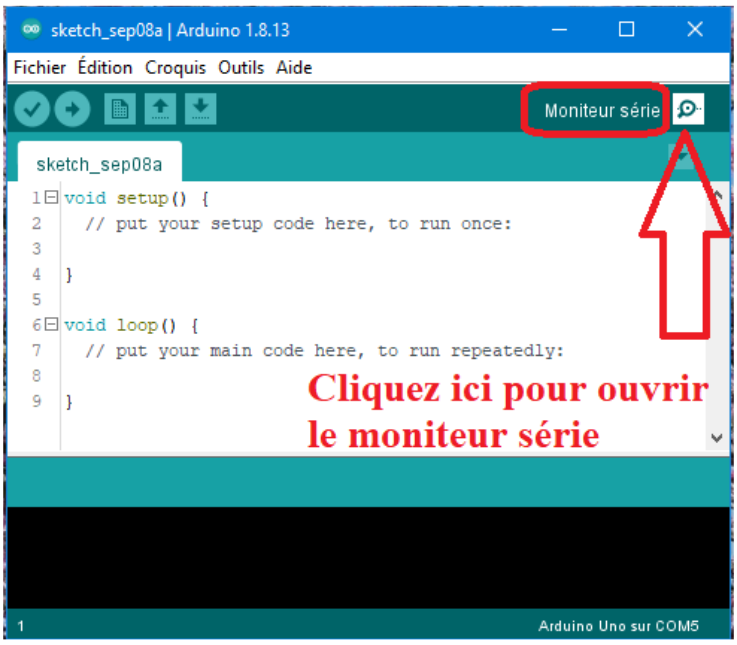


Schéma des connexions du module HC-05 pour paramétrage

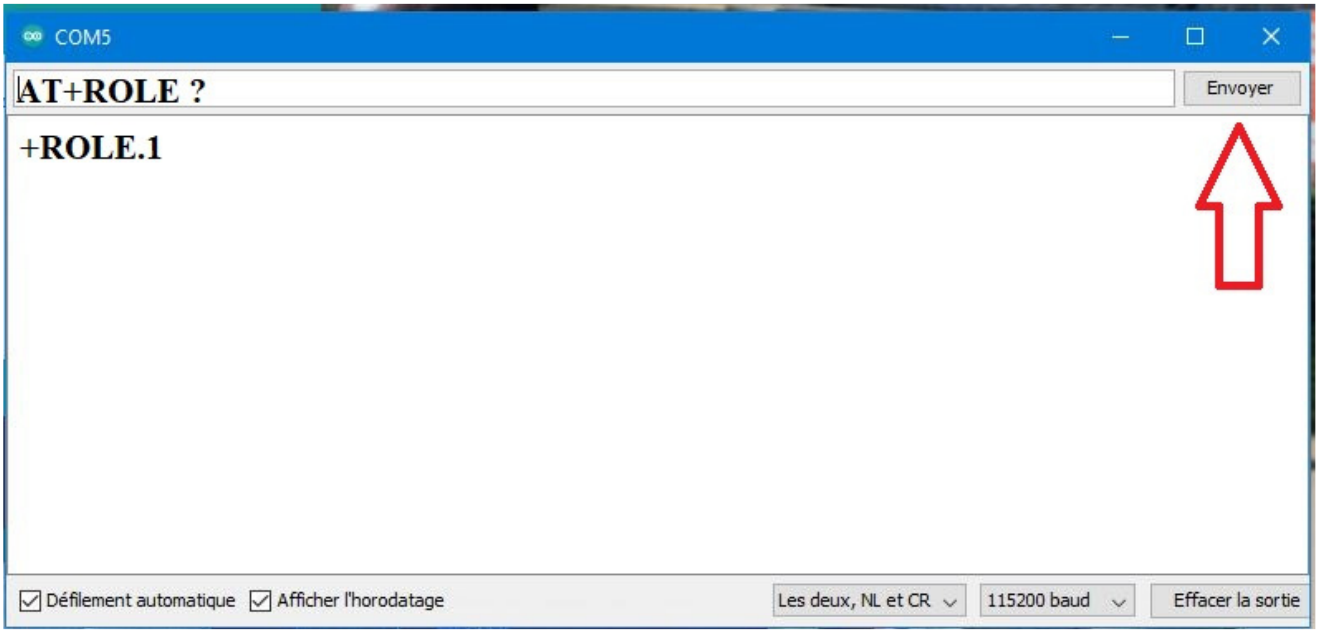
Le module HC-05 sera « maître » lorsqu'il enverra comme réponse « **+ROLE.1** » à la question posée via ARDUINO : « **AT+ROLE ?** ». Voir plus bas le détail des actions possibles.



Fenêtre de travail dans l'environnement ARDUINO

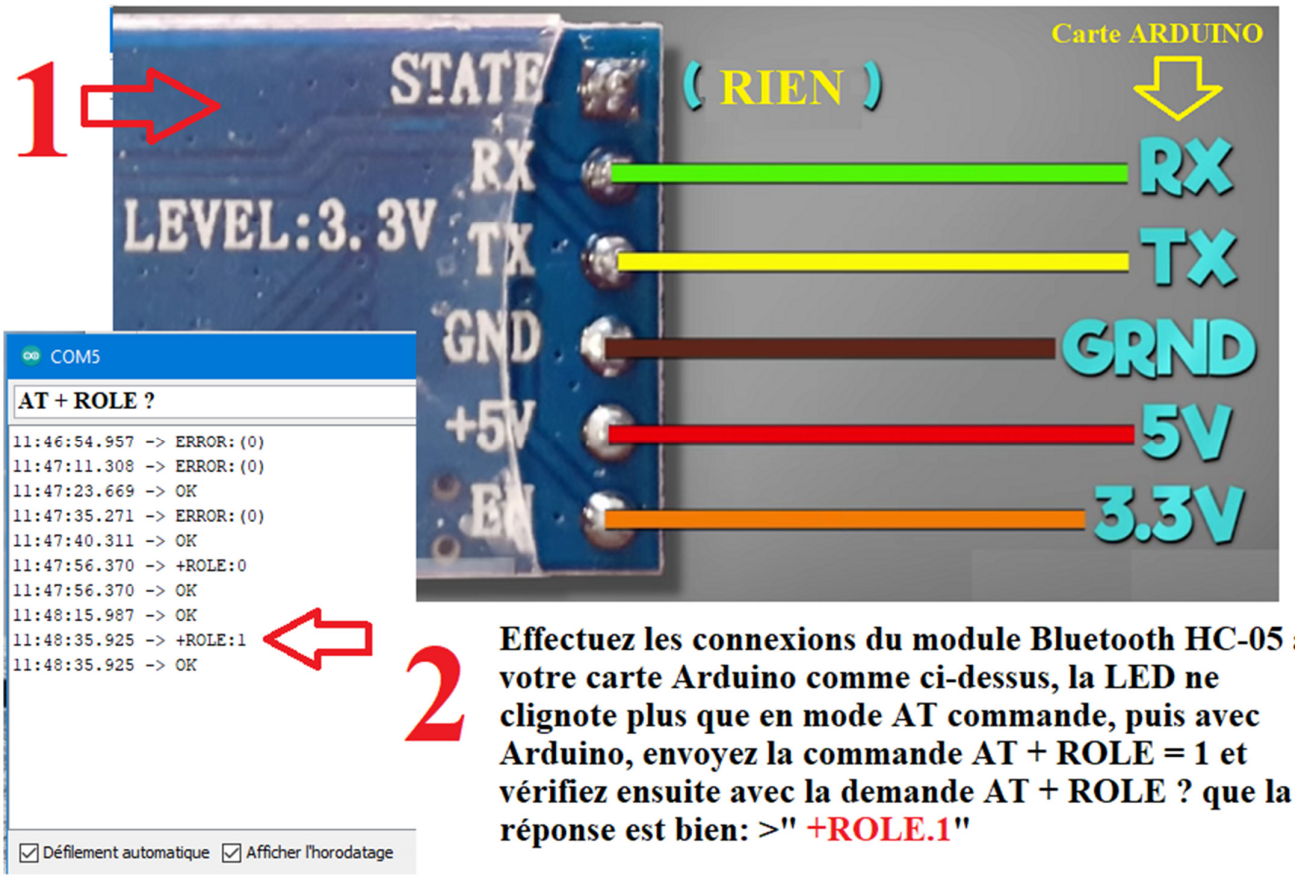
Démarrez l'environnement ARDUINO, vous allez obtenir l'ouverture de cette fenêtre de travail. Connectez votre carte ARDUINO UNO à votre ordinateur avant toute opération de téléversement. Cliquez sur Outils pour sélectionner la bonne carte Arduino Uno et son port de connexion (COM 5 ou autre).

Lorsque vous êtes connecté, cliquez sur « Moniteur Série » pour pouvoir communiquer avec le module Bluetooth HC-05 :



Vous tapez vos messages sur la ligne du haut, par exemple la question : AT+ROLE ? puis vous cliquez sur la touche « Envoyer ». La réponse transmise par le module HC-05 s'affiche alors en dessous ligne après ligne, par exemple : +ROLE.1 (Ce qui signifie : Mode Maître ou Master).

Ensuite, vous pouvez poursuivre votre travail en connectant ce module come indiqué plus haut page

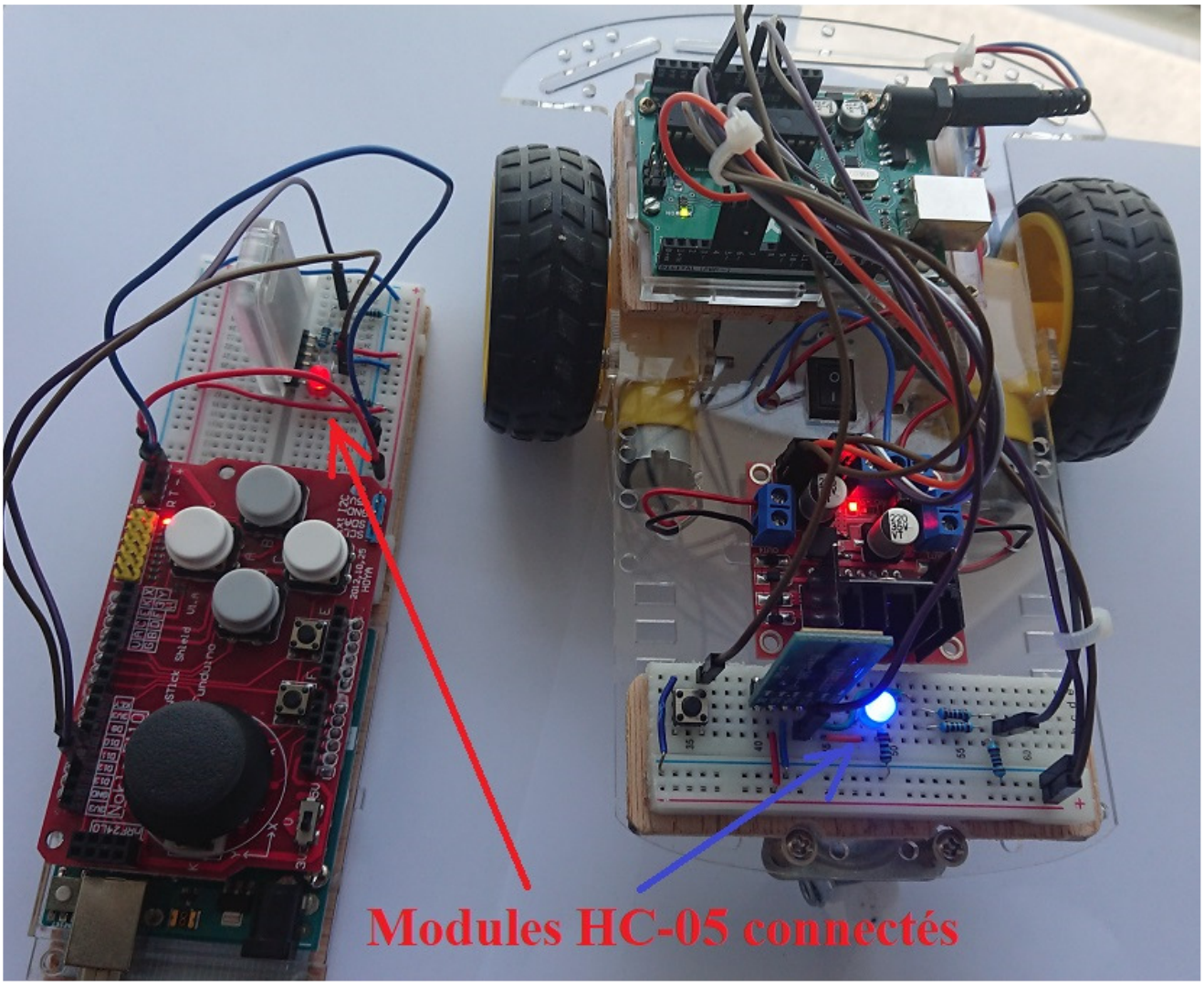


Ensuite, profitez de ce mode de connexion vous permettant d'enregistrer vos préférences d'utilisation de ce module en utilisant les commandes "AT"

Commandes AT utiles:

AT + ROLE = 0 >>> rend le module "esclave" (C'est le cas du module installé sur le triporteur)
AT + ROLE = 1 >>> rend le module "maître" ***(C'est ce que nous voulons faire avec le module connecté à la carte Joystick Shield)***
AT + ROLE ? >>> Vous posez une question au module qui va vous répondre par exemple: **+ROLE:1** si vous avez déjà envoyé la commande précédente
AT + CMODE = 0 >>> connexion avec un seul dispositif (toujours le même)
AT + CMODE = 1 >>> connexion avec n'importe quel dispositif disponible
AT + CMODE ? >>> Vous posez une question au module qui va vous répondre par exemple: **+CMOD:1** si vous avez envoyé la commande précédente
AT + PSWD ? >>> Question pour demander le mot de passe (PassWord), la réponse sera certainement : **+PSWD:1234**

Vous constaterez ensuite que les deux modules HC-05, le module « esclave » du triporteur et le module « maître » de la télécommande se connectent automatiquement en quelques secondes, comme ci-dessous, chaque fois que les deux systèmes ARDUINO sont "allumés".



Vous trouverez le programme Arduino de la télécommande (B4R) détaillé plus bas aux pages 30 et 31 de cette brochure.

Téléchargez le programme B4R: "4BoutonsShield" page 31

Programme Triporteur Arduino B4R détaillé

```
'Triporteur ARDUINO B4R - par Marc DANIEL - Juillet/Septembre 2021
#Region Project Attributes
#AutoFlushLogs: True
#CheckArrayBounds: True
#StackBufferSize: 300
#End Region
Sub Process_Globals
Public Serial1 As Serial
Private SoftwareSerial1 As SoftwareSerial
Private astream As AsyncStreams
Private IN1, IN2, IN3, IN4, ENA, ENB, Lights, Klaxon As Pin
Private BtnTEST As Pin
Private SpeedD, SpeedG As UInt
Private X=0, Y=0 As UInt
Private FEUX As Boolean
Private SON As Boolean
End Sub
Private Sub AppStart
Serial1.Initialize(115200)
Log("Démarrage du Triporteur")
IN1.Initialize(4, IN1.MODE_OUTPUT)
IN2.Initialize(5, IN2.MODE_OUTPUT)
IN3.Initialize(6, IN3.MODE_OUTPUT)
IN4.Initialize(7, IN4.MODE_OUTPUT)
ENA.Initialize(9, ENA.MODE_OUTPUT)
ENB.Initialize(10, ENB.MODE_OUTPUT)
Lights.Initialize(13, Lights.MODE_OUTPUT)
Klaxon.Initialize(3, Klaxon.MODE_OUTPUT)
BtnTEST.Initialize(BtnTEST.A0,BtnTEST.MODE_INPUT_PULLUP) 'Bouton poussoir qui lance le test des moteurs
BtnTEST.AddListener("BtnTEST_StateChanged")
SoftwareSerial1.Initialize(9600, 11, 12) 'Software Serial port sur les broches 11 et 12
'Broche TXD de HC-05 directement reliée à Arduino 11 - Broche RXD de HC-05 protégée par 3 résistances de 1 Ko reliée à Arduino 12
astream.Initialize(SoftwareSerial1.Stream, "astream_NewData", Null)
SpeedD=0
SpeedG=0
FEUX = False
SON = False
End Sub
Private Sub BtnTEST_StateChanged(State As Boolean) 'Appui sur le bouton de test
If State = False Then 'BoutonTEST=0
Log("Bouton test: ", State)
TestMoteurs
Else 'BoutonTEST=1
Log("Bouton test: ", State)
End If
End Sub
Private Sub TestMoteurs
' Test de fonctionnement des moteurs après l'appui sur le bouton TEST
Log("Démarrage du test des moteurs")
'lance le moteur droit en avant
IN1.DigitalWrite(True)
IN2.DigitalWrite(False)
'lance le moteur gauche en avant
IN3.DigitalWrite(True)
IN4.DigitalWrite(False)
SpeedD = 90
SpeedG = 90
Log("Marche avant")
Envoi
Delay(3000) 'environ 3 secondes de marche avant
'lance le moteur droit en arrière
IN1.DigitalWrite(False)
IN2.DigitalWrite(True)
'lance le moteur gauche en arrière
IN3.DigitalWrite(False)
```

```
IN4.DigitalWrite(True)
SpeedD = 90
SpeedG = 90
Log("Marche arrière")
Envoi
Delay(3000) 'environ 3 secondes de marche arrière
SpeedD = 0 'arrêt du triporteur avec des vitesses nulles
SpeedG = 0
Envoi
'Fin du test de fonctionnement des moteurs
End Sub
Sub AStream_NewData (Buffer() As Byte)
If Buffer.Length = 2 Then
X= Buffer(0)
Log("X: ",X)
Delay(10)
Y=Buffer(1)
Log("Y: ",Y)
If Y < 117 Then
' MARCHE AVANT deux moteurs
'lance le moteur droit en avant
IN1.DigitalWrite(True)
IN2.DigitalWrite(False)
'lance le moteur gauche en avant
IN3.DigitalWrite(True)
IN4.DigitalWrite(False)
SpeedD = 90
SpeedG = 90
Log("Marche avant")
Envoi
else if Y > 137 And Y <> 199 Then '199 est réservé à l'utilisation du Klaxon
If Y <> 249 Then ' 249 est réservé à l'allumage des feux ou LEDs
'MARCHE ARRIÈRE deux moteurs
'lance le moteur droit en arrière
IN1.DigitalWrite(False)
IN2.DigitalWrite(True)
'lance le moteur gauche en arrière
IN3.DigitalWrite(False)
IN4.DigitalWrite(True)
SpeedD = 90
SpeedG = 90
Log("Marche arrière")
Envoi
End If
Else
SpeedD=0
SpeedG=0
Log("arrêt des moteurs")
Envoi
End If
If X < 117 Then
'Rotation à gauche
Log("Rotation à gauche")
SpeedD = 55 ' Ces valeurs peuvent être légèrement modifiées en fonction des moteurs utilisés
SpeedG = 90 ' ou de la tension des batteries installées
' Sur certains moteurs, la vitesse 55 n'agira pas, seule la roue gauche avancera, tentez alors les valeurs de 60 ou 65 pour la roue droite
Envoi
else if X > 137 And X <> 199 Then ' 199 est réservé à l'utilisation du Klaxon
If X <> 249 Then ' 249 est réservé à l'allumage des feux ou LEDs
'Rotation à droite
Log("Rotation à droite")
SpeedD= 90 ' Ces valeurs peuvent être légèrement modifiées en fonction des moteurs utilisés
SpeedG= 55 ' ou de la tension des batteries installées
' Sur certains moteurs, la vitesse 55 n'agira pas, seule la roue droite avancera, tentez alors les valeurs de 60 ou 65 pour la roue gauche
Envoi
End If
End If
```

```
End If
If Y = 250 And X = 250 Then
' MARCHE AVANT rapide deux moteurs
'lance le moteur droit en avant
IN1.DigitalWrite(True)
IN2.DigitalWrite(False)
'lance le moteur gauche en avant
IN3.DigitalWrite(True)
IN4.DigitalWrite(False)
SpeedD = 120
SpeedG = 120
Log("Marche avant rapide")
Envoi
else if Y=200 And X=200 Then
'MARCHE ARRIÈRE rapide deux moteurs
'lance le moteur droit en arrière
IN1.DigitalWrite(False)
IN2.DigitalWrite(True)
'lance le moteur gauche en arrière
IN3.DigitalWrite(False)
IN4.DigitalWrite(True)
SpeedD = 120
SpeedG = 120
Log("Marche arrière rapide")
Envoi
else if Y=249 And X=249 Then ' GESTION des FEUX, LEDS, DIODES et lumières diverses si connectés broche 13
If FEUX=False Then
Lights.DigitalWrite(True)
FEUX = True
SpeedD = 0
SpeedG = 0
Log("Allumage des LEDs") 'Allumage des LEDS si elles existent et ont été connectées sur la broche 13
Envoi
Else
Lights.DigitalWrite(False)
FEUX = False
SpeedD = 0
SpeedG = 0
Log("Extinction des LEDs") 'Extinction des LEDS si elles existent et ont été connectées sur la broche 13
Envoi
End If
else if Y = 199 And X = 199 Then ' GESTION du «Klaxon» à savoir du buzzer ou dispositif similaire si connecté à la broche 3
If SON = False Then
Klaxon.DigitalWrite(True)
SON = True
SpeedD = 0
SpeedG = 0
Log("Déclenchement du Klaxon si un buzzer a été installé et connecté")
Envoi
Else
Klaxon.DigitalWrite(False)
SON = False
SpeedD = 0
SpeedG = 0
Log("Arrêt du Klaxon")
Envoi
End If
End If
End Sub
Private Sub Envoi ' Transmission des vitesses en fonction des données reçues par Bluetooth ou exécution du test des moteurs
'Gestion vitesse roue avant droite
ENA.analogWrite(SpeedD)
Log("Vitesse roue droite:", SpeedD)
'Gestion vitesse roue avant gauche
ENB.analogWrite(SpeedG)
Log("Vitesse roue gauche:",SpeedG)
End Sub
```

Application PiloteBluetooth.B4A détaillée

(Cette application comprend trois modules et la fenêtre graphique appelée « Layout » ou « Virtual Designer ».)

Module principal « Main »

```
#Region Project Attributes
' Application B4A développée par Marc DANIEL - Juillet Août 2021
#ApplicationLabel: Pilote_TriporteurBT
#VersionCode: 2
#VersionName:
'SupportedOrientations possible values: unspecified, landscape or portrait.
#SupportedOrientations: portrait
#CanInstallToExternalStorage: False
#End Region
#Region Activity Attributes
#FullScreen: False
#IncludeTitle: False
#End Region
Sub Process_Globals
Public xui As XUI
Public rp As RuntimePermissions
End Sub
Sub Globals
Private BTA As BluetoothAsynchStream
Private BtnConnect, Sortie As Button
Private lblStatus, Pilotage As Label
Private pnlMain As B4XView
Private Avant, Arriere, ARPlus, AVPlus, Droite, Gauche, STOP, Lights, Klaxon As
ImageView
Private ProgressBar1 As ProgressBar
Private DeviceName = "HC-05" As String
Private Connected As Boolean
Private x=127, y=127 As Int 'Valeurs de x et y pour l'arrêt du véhicule (= Bouton STOP)
Private BLT As ImageView
End Sub
Sub Activity_Create(FirstTime As Boolean)
Activity.LoadLayout("Main")
BTA.Initialize(Me, "BTA", lblStatus, ProgressBar1)
Pilotage.Text=" Bienvenue à bord du " & CRLF & " poste de pilotage Bluetooth " & CRLF & " de votre triporteur !"
Pilotage.Text=Pilotage.Text & CRLF & " Les commandes apparaîtront " & CRLF & " dès que la connexion sera " & CRLF & " établie avec le
véhicule."
Pilotage.Text=Pilotage.Text & CRLF & " Appuyez sur la touche verte " & CRLF & "CONNEXION !"
End Sub
Sub Activity_Resume
End Sub
Sub Activity_Pause (UserClosed As Boolean)
If UserClosed = True And Connected = True Then
BTA.SendBytes(Array As Byte(127, 127))
BTA.Disconnect
End If
End Sub
Private Sub Activity_PermissionResult (Permission As String, Result As Boolean)
End Sub
Sub BtnConnect_Click
rp.CheckAndRequest(rp.PERMISSION_ACCESS_FINE_LOCATION)
Wait For Activity_PermissionResult (Permission As String, Result As Boolean)
If Result = False Then
lblStatus.Text = "Statut: Permission refusée..."
Else
BTA.Connect(DeviceName)
End If
End Sub
Sub Klaxon_Click
x=199
y=199
BTA.SendBytes(Array As Byte(x,y))
End Sub
Sub Lights_Click
x=249
y=249
```

```
BTA.SendBytes(Array As Byte(x,y))
End Sub
Sub STOP_Click
x=127
y=127
BTA.SendBytes(Array As Byte(x,y))
End Sub
Sub Avant_Click
x=127
y=80
BTA.SendBytes(Array As Byte(x, y))
End Sub
Sub AVPlus_Click
x=250
y=250
BTA.SendBytes(Array As Byte(x, y))
End Sub
Sub Arriere_Click
x=127
y=180
BTA.SendBytes(Array As Byte(x, y))
End Sub
Sub ARPlus_Click
x=200
y=200
BTA.SendBytes(Array As Byte(x, y))
End Sub
Sub Droite_Click
x=180
y=63
BTA.SendBytes(Array As Byte(x, y))
End Sub
Sub Gauche_Click
x=63
y=63
BTA.SendBytes(Array As Byte(x, y))
End Sub
Private Sub BTA_Connected (Success As Boolean)
If Success = True Then
Connected = Success
BtnConnect.Visible=False
BLT.Visible=True
Sortie.Visible=True
Pilotage.Visible=False
Avant.Visible=True
Arriere.Visible=True
ARPlus.Visible=True
AVPlus.Visible=True
Droite.Visible=True
Gauche.Visible=True
STOP.Visible=True
Lights.Visible=True
Klaxon.Visible=True
End If
End Sub
Sub Sortie_Click
BTA.SendBytes(Array As Byte(127,127)) 'Arrêt des moteurs
Activity.Finish
End Sub
Sub Activity_KeyPress (KeyCode As Int) As Boolean
If KeyCode = KeyCodes.KEYCODE_BACK Then openMsgBox
Return True
End Sub
Sub openMsgBox
Msgbox2Async("Voulez-vous vraiment quitter le pilotage Bluetooth ?","Quitter ?","Oui", "", "Non", Null, True)
Wait For Msgbox_Result (Result As Int)
If Result=DialogResponse.POSITIVE Then
BTA.SendBytes(Array As Byte(127,127)) 'Arrêt des moteurs
Activity.Finish
End If
End Sub
```

Module « BluetoothAsynchStream »

```
#Event: Connected (Success As Boolean)
#Event: StateChanged (State As String)
Sub Class_Globals
Private Admin As BluetoothAdmin
Public Astreams As AsyncStreams
Private Serial As Serial
Private PH As Phone
Private mParent As Object
Private mEventName As String
Private lblStatus As Label
Private ProgressBar1 As ProgressBar
Private GenericDeviceName, DeviceName As String
Public BluetoothState, ConnectionState, DeviceFound As Boolean
Private DeviceName, DeviceMacAdress As String
Public CharSet = "UTF-8" As String
Private sb As StringBuilder
Public State As String
End Sub
Public Sub Initialize(Parent As Object, EventName As String, StatusLabel As Label, Pgb As ProgressBar)
mParent = Parent
mEventName = EventName
lblStatus = StatusLabel
Admin.Initialize("Admin")
Serial.Initialize("Serial")
ProgressBar1 = Pgb
sb.Initialize
If Admin.IsEnabled = False Then
If Admin.Enable = False Then
ToastMessageShow("Module Bluetooth non disponible", True)
Else
ToastMessageShow("Module Bluetooth disponible...", False)
End If
Else
BluetoothState = True
End If
End Sub
Public Sub Connect(Name As String)
GenericDeviceName = Name
Dim success As Boolean = Admin.StartDiscovery
If success = False Then
lblStatus.Text = "Erreur du processus de recherche"
Else
lblStatus.Text = "Recherche dispositif en cours"
ProgressBar1.Visible = True
End If
End Sub
Public Sub Disconnect
If Astreams.IsInitialized Then Astreams.Close
If Serial.IsInitialized Then Serial.Disconnect
End Sub
Private Sub Admin_DeviceFound (Name As String, MacAddress As String)
Log(Name & ":" & MacAddress)
If Name.Contains(GenericDeviceName) Then
Log(GenericDeviceName & " trouvé")
DeviceName = Name
DeviceMacAdress = MacAddress
If PH.SdkVersion <= 24 Then
Admin.CancelDiscovery
lblStatus.Text = GenericDeviceName & " trouvé"
Else
lblStatus.Text = GenericDeviceName & " trouvé, veuillez patienter."
End If
End If
End Sub
Private Sub Admin_DiscoveryFinished
If DeviceName = "" Then
lblStatus.Text = "Module " & GenericDeviceName & " non détecté"
Msgbox2Async("Le Module Bluetooth HC-05 ne semble pas être alimenté, allumez le triporteur, la mini-led rouge du module doit clignoter
rapidement puis appuyer de nouveau sur le bouton «Connexion».","Vérifiez l'alimentation du module", "OK", "", "", Null, True)
```

```
Wait For Msgbox_Result (Result As Int)
Else
lblStatus.Text = "Connexion à " & GenericDeviceName
Serial.Connect(DeviceMacAdress)
End If
End Sub
Private Sub Admin_StateChanged (NewState As Int, OldState As Int)
Log("Changement de statut: " & NewState)
lblStatus.Text = "Statut actuel: " & NewState
BluetoothState = NewState = Admin.STATE_ON
End Sub
Sub Serial_Connected (Success As Boolean)
Private msg As String
If Success = True Then
If Astreams.IsInitialized Then Astreams.Close
Astreams.Initialize(Serial.InputStream, Serial.OutputStream, "Astreams")
msg = "Module Bluetooth HC05 Connecté"
ProgressBar1.Visible = False
Else
Log(LastException.Message)
msg = LastException.Message
End If
lblStatus.Text = msg
CallSubDelayed2(mParent, mEventName & "_Connected", Success)
End Sub
Public Sub SendBytes(Buffer() As Byte)
Astreams.Write(Buffer)
End Sub
Public Sub SendText(Text As String)
Astreams.Write(Text.GetBytes(CharSet))
End Sub
Public Sub Ast_NewText(Text As String)
CallSubDelayed2(mParent, mEventName & "_NewText", Text)
Log(Text)
End Sub
Public Sub WriteText(Text As String)
Astreams.Write(Text.GetBytes(CharSet))
End Sub
Public Sub WriteBytes(Buffer() As Byte)
Astreams.Write(Buffer)
End Sub
Private Sub Astreams_NewData (Buffer() As Byte)
Dim newDataStart As Int = sb.Length
sb.Append(BytesToString(Buffer, 0, Buffer.Length, CharSet))
Dim s As String = sb.ToString
Dim start As Int = 0
For i = newDataStart To s.Length - 1
Dim c As Char = s.CharAt(i)
If i = 0 And c = Chr(10) Then '\n...
start = 1
Continue
End If
If c = Chr(10) Then
CallSubDelayed2(mParent, mEventName & "_NewText", s.SubString2(start, i))
start = i + 1
Else If c = Chr(13) Then
CallSubDelayed2(mParent, mEventName & "_NewText", s.SubString2(start, i))
If i < s.Length - 1 And s.CharAt(i + 1) = Chr(10) Then
i = i + 1
End If
start = i + 1
End If
Next
If start > 0 Then sb.Remove(0, start)
End Sub
Private Sub Astreams_Terminated
CallSubDelayed(mParent, mEventName & "_Terminated")
End Sub
Private Sub Astreams_Error
Log("error: " & LastException)
Astreams.Close
```

```
CallSubDelayed(mParent, mEventName & "_Terminated")
End Sub
Public Sub Close
Astreams.Close
End Sub
```

Module « Starter »

```
#Region Service Attributes
#StartAtBoot: False
#ExcludeFromLibrary: True
#End Region
Sub Process_Globals
End Sub
Sub Service_Create
End Sub
Sub Service_Start (StartingIntent As Intent)
End Sub
Sub Service_TaskRemoved
'Cet événement sera déclenché lorsque l'utilisateur supprimera l'application de la liste des applications récentes.
End Sub
'Retourne «true » pour permettre au gestionnaire d'exceptions par défaut du système d'exploitation de gérer l'exception non interceptée.
Sub Application_Error (Error As Exception, StackTrace As String) As Boolean
Return True
End Sub
Sub Service_Destroy
End Sub
```

Programme B4R – Télécommande Carte Joystick Shield à boutons

```
'Triporteur Bluetooth - Pilotage sans fil - Dispositif Maître avec Module Bluetooth HC-05
' Carte Joystick Shield ==> utilisation simple des boutons A, B, C, D, E, F, G
' Marc DANIEL – Mai-Septembre 2021
Sub Process_Globals
Public Serial1 As Serial
Private SoftWareSerial1 As SoftwareSerial
Private astream As AsyncStreams
Private BtnA, BtnB, BtnC, BtnD, BtnE, BtnF, BTJ As Pin
Private X=127, Y=127 As UInt ' Valeurs des moteurs à l'arrêt par défaut
Private Timer1 As Timer
End Sub
Private Sub AppStart
Serial1.Initialize(115200)
Log("Démarrage du triporteur !")
SoftWareSerial1.Initialize(9600, 11, 12)
astream.Initialize(SoftWareSerial1.Stream, "astream_NewData", Null)
BtnA.Initialize(2, BtnA.MODE_INPUT_PULLUP)
BtnA.AddListener("BtnA_StateChanged")
BtnB.Initialize(3, BtnB.MODE_INPUT_PULLUP)
BtnB.AddListener("BtnB_StateChanged")
BtnC.Initialize(4, BtnC.MODE_INPUT_PULLUP)
BtnC.AddListener("BtnC_StateChanged")
BtnD.Initialize(5, BtnD.MODE_INPUT_PULLUP)
BtnD.AddListener("BtnD_StateChanged")
BtnE.Initialize(6, BtnE.MODE_INPUT_PULLUP)
BtnE.AddListener("BtnE_StateChanged")
BtnF.Initialize(7, BtnF.MODE_INPUT_PULLUP)
BtnF.AddListener("BtnF_StateChanged")
' Gros bouton du Joystick
BTJ.Initialize(8, BtnD.MODE_INPUT_PULLUP)
BTJ.AddListener("BtnD_StateChanged")
Timer1.Initialize("timer1_Tick", 50)
Timer1.Enabled = True
End Sub
Private Sub BtnA_StateChanged(StateA As Boolean) 'Marche avant >>> OK
Log("État: ", StateA)
If StateA Then
X=127 'False (Moteurs à l'arrêt)
Y=127
Else
X= 127 'True (Les deux roues tournent dans le sens normal)
Y= 80
End If
End Sub
Private Sub BtJ_StateChanged(StateJ As Boolean) 'Marche avant rapide >>> OK
Log("État: ", StateJ)
If StateJ Then
X=127 'False (Moteurs à l'arrêt)
Y=127
Else
X= 250 'True (Les deux roues tournent rapidement en avant)
Y= 250
End If
End Sub
Private Sub BtnB_StateChanged(StateB As Boolean) ' Virage à droite >>> OK
Log("État: ", StateB)
If StateB Then
X=127 'False (Moteurs à l'arrêt)
Y=127
Else
X=180 'True (La roue gauche avance seule)
Y=63
End If
End Sub
Private Sub BtnC_StateChanged(StateC As Boolean) 'Marche arrière >>> OK
Log("État: ", StateC)
If StateC Then
X=127 'False (Moteurs à l'arrêt)
Y=127
```

```
Else
X=127 'True (Les deux roues tournent en sens inverse)
Y=180
End If
End Sub
Private Sub BtnD_StateChanged(StateD As Boolean) ' Virage à gauche >>> OK
Log("État: ", StateD)
If StateD Then
X=127 ' False (Moteurs à l'arrêt)
Y=127
Else
X=63 ' True (La roue droite avance seule)
Y=63
End If
End Sub
Private Sub BtnE_StateChanged(StateE As Boolean) ' Allumage ou Extinction des feux si des
LEDS ont été installées
Log("État: ", StateE)
If StateE Then
X=127 ' False (Moteurs à l'arrêt)
Y=127
Else
X=249 ' True (Les feux s'allument ou s'éteignent)
Y=249
End If
End Sub
Private Sub BtnF_StateChanged(StateF As Boolean) ' Coup de klaxon si buzzer installé
Log("État: ", StateF)
If StateF Then
X=127 ' False (Moteurs à l'arrêt)
Y=127
Else
X=199 ' True (Klaxon: 2 secondes de buzzer si installé et connecté)
Y=199
End If
End Sub
Sub Timer1_Tick
Private b(2) As Byte
Log ("X: ", X)
Log("Y: ", Y)
b=Array As Byte(X, Y)
astream.Write(b)
End Sub
Sub AStream_NewData (Buffer() As Byte)
End Sub
```

Téléchargement des logiciels

- Triporteur Arduino B4R >>>
  https://www.marcalaindaniel.fr/ARDUINO/TriporteurBluetooth/TriporteurARDUINO.zip

- Pilote Bluetooth B4A >>>
  https://www.marcalaindaniel.fr/ARDUINO/TriporteurBluetooth/PilotePlus_Bluetooth.zip

- Fichier Bluetooth APK directement installable >>>
  https://www.marcalaindaniel.fr/ARDUINO/TriporteurBluetooth/NewTriporteur/PiloteAPK.zip

- Programme JoystickShieldBoutons B4R >>>
  https://www.marcalaindaniel.fr/ARDUINO/TriporteurBluetooth/4BoutonsShield.zip

Table des matières