

Construire, équiper puis piloter à distance une voiture Arduino

Brochure pédagogique par Marc DANIEL – Novembre 2021



Table des matières

- Accueil	page 1
- Présentation	page 2
- Montage du châssis	pages 3 à 10
- Équipement électronique n° 1 de la voiture (Carte MEGA2560)	pages 11 à 15
- Programme Visual BASIC B4R pour matériel Arduino	pages 15 à 20
- Équipement électronique n° 2 de la voiture (Carte UNO)	pages 21 à 26
- Modifications du programme B4R pour la carte UNO	pages 26 et 27
- Application Visual BASIC B4A pour Android	pages 28 à 30
- Bricolage artisanal alternatif	pages 31 à 37

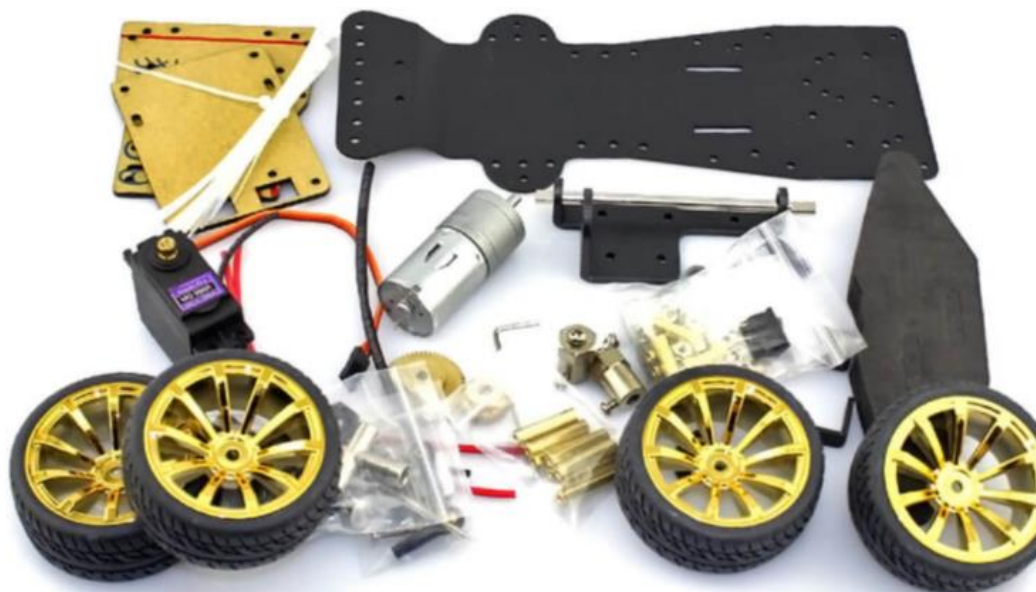
Se procurer le kit Arduino de très bonne qualité fabriqué en Chine et souvent fourni sans notice de montage :

>>> <https://www.elecrow.com/4wd-smart-car-robot-chassis-for-arduino-servo-steering.html>

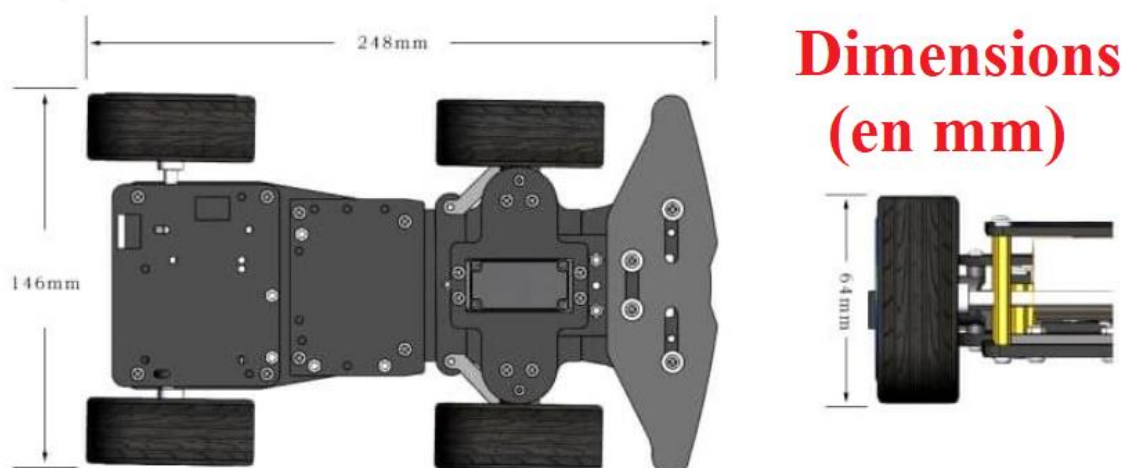
>>> https://fr.aliexpress.com/wholesale?SearchText=4wd-smart-car-robot-chassis-for-arduino-servo&origin=y&d=y&catId=0&initiative_id=SB_20211104014144

Compter entre 45 et 50 € selon les fournisseurs

Il s'agit d'une voiture classique à propulsion arrière avec deux roues motrices dont le moteur est fourni et une direction avant commandée par un servomoteur fourni également.

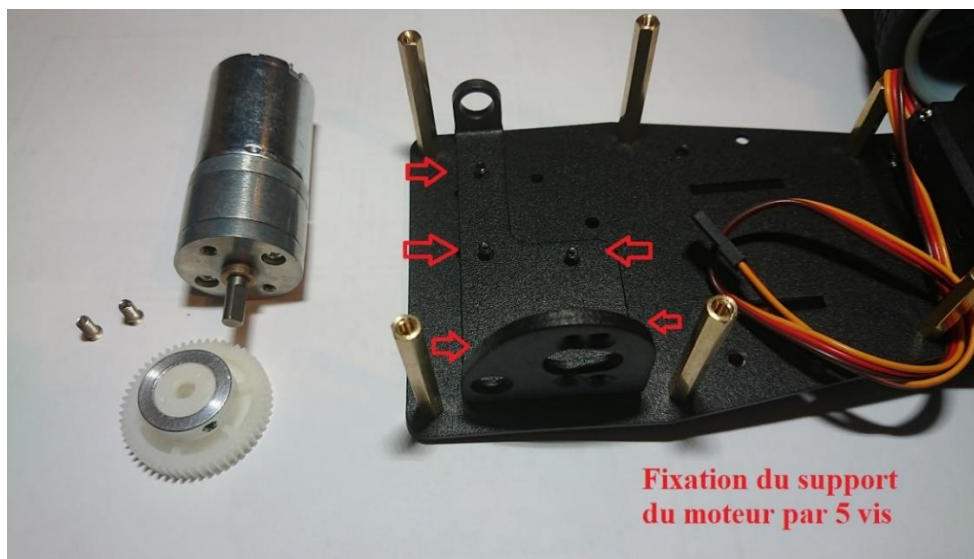


Composition du kit fourni par les commerçants

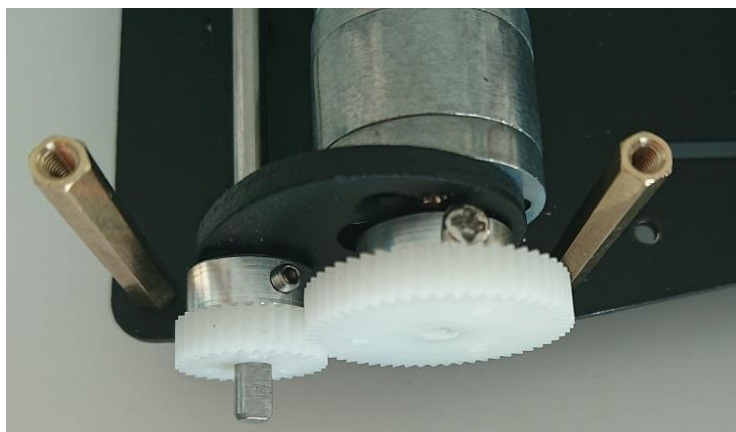
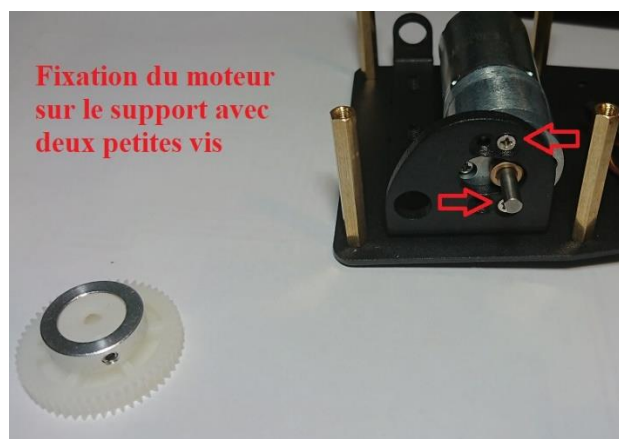


Dimensions de la voiture Arduino

Montage du châssis



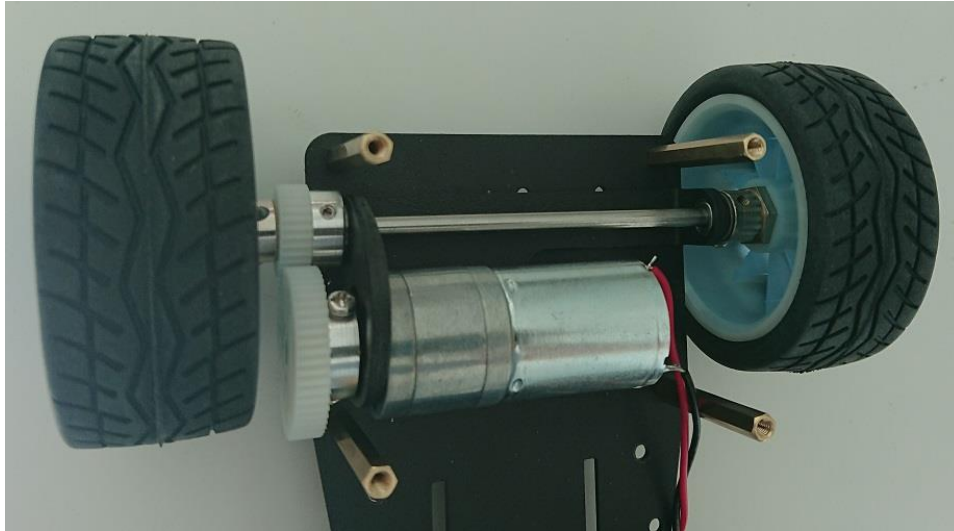
Fixation du support du moteur et du train arrière



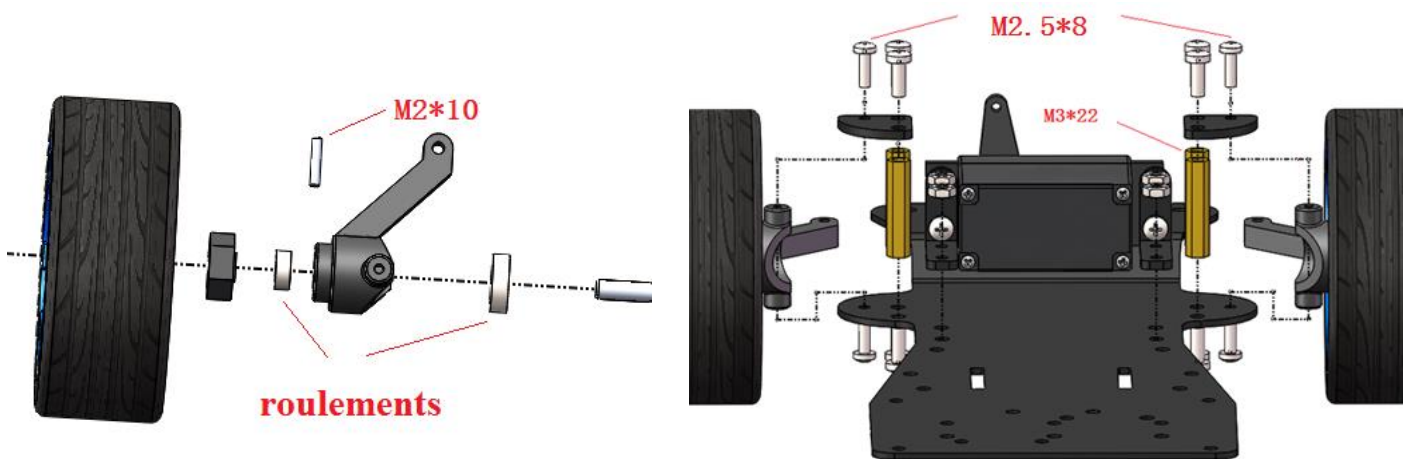
Mise en place et fixation du moteur



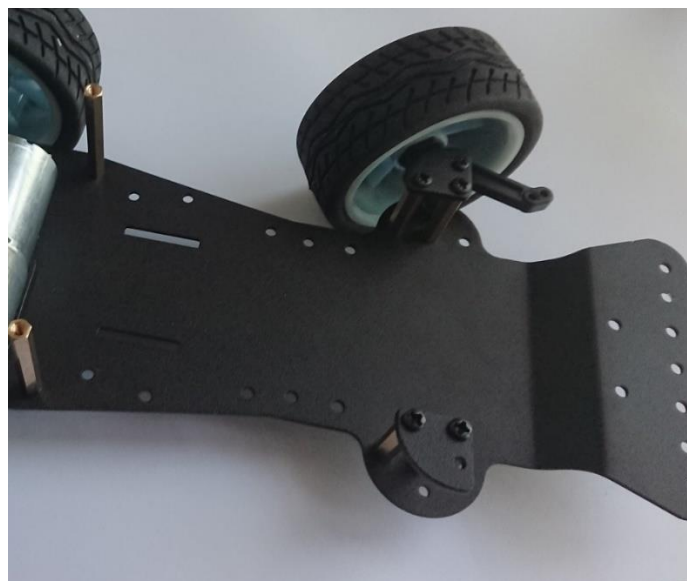
Préparer les pièces composant le train arrière de la voiture

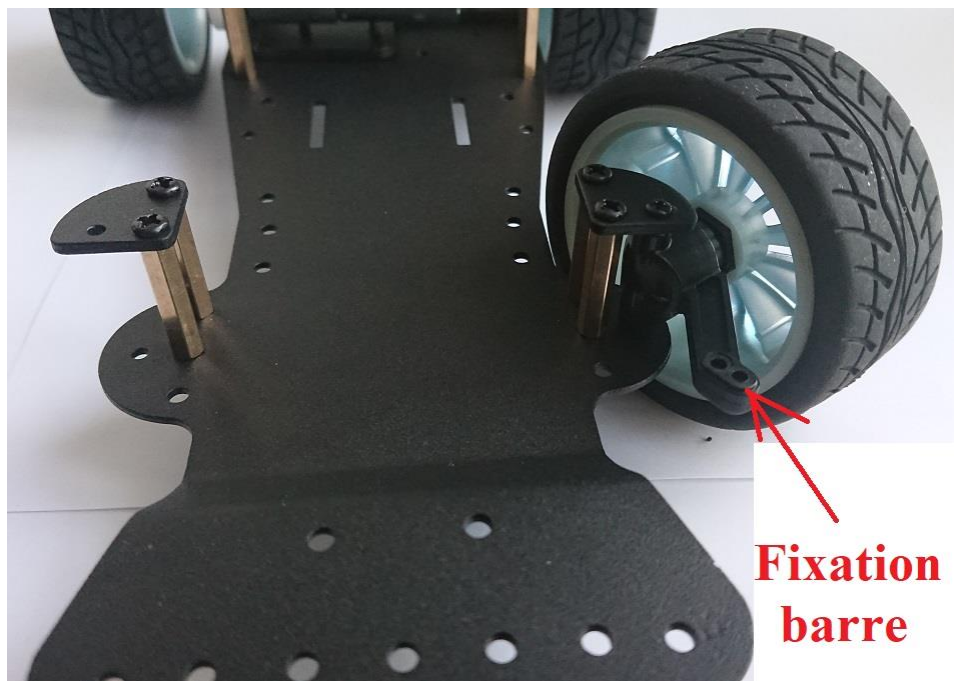


Placer l'axe arrière ainsi sans oublier les deux petits roulements qui s'emboîtent dans le support, fixer solidement le petit pignon à l'aide des vis encastrées.



Préparer les supports et accessoires pour les roues avant et le système de direction

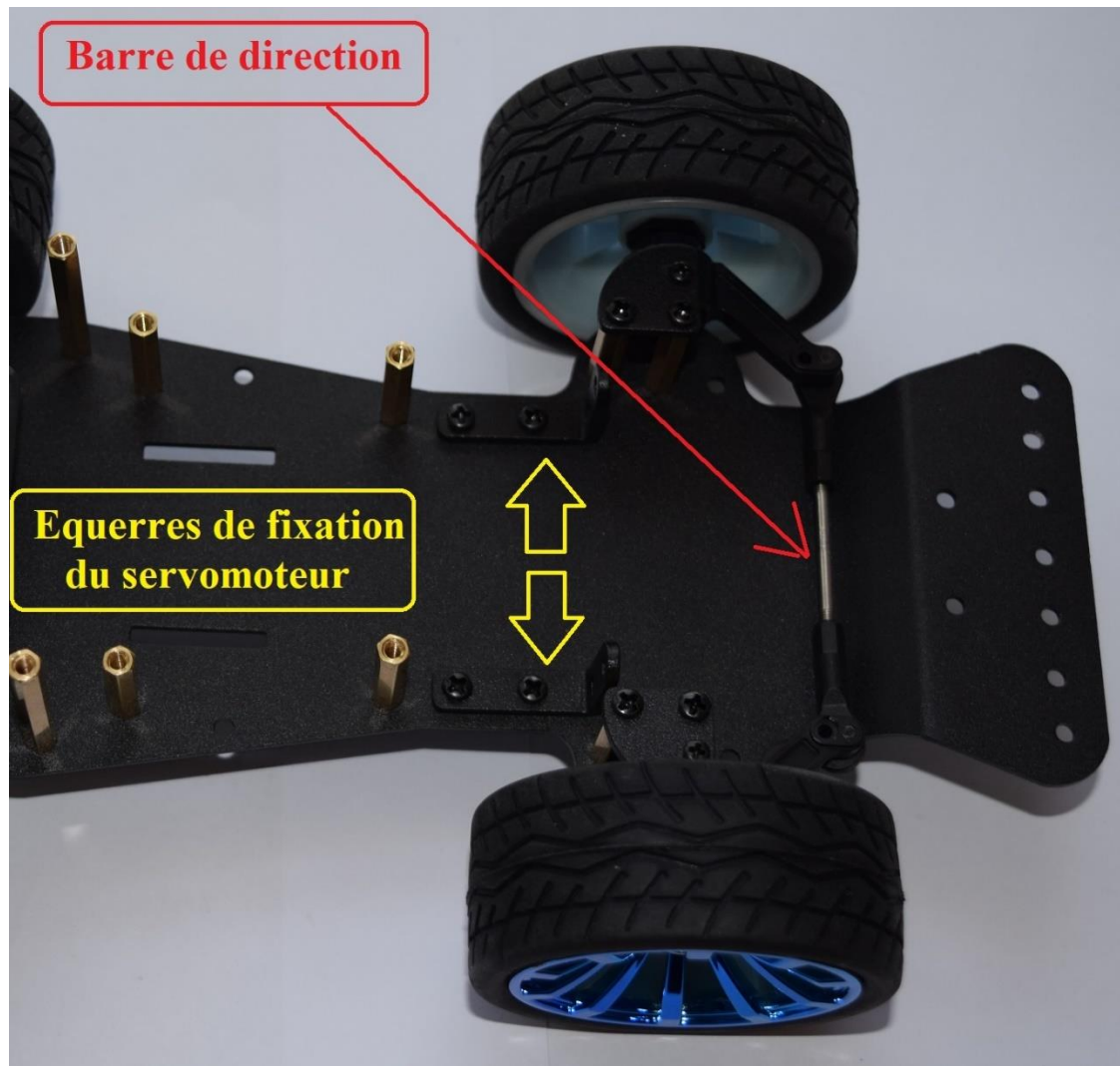




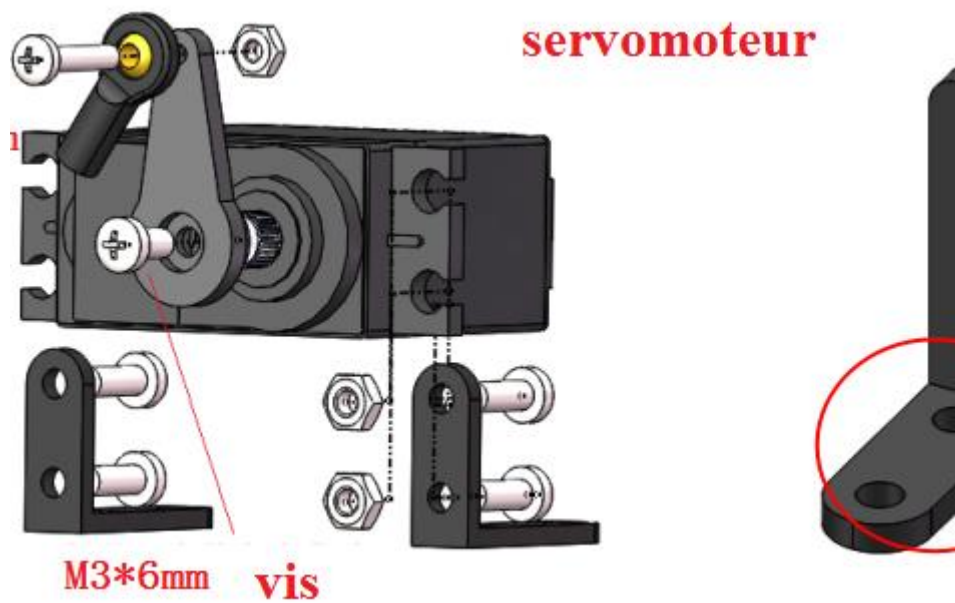
Zoom sur les éléments du train avant

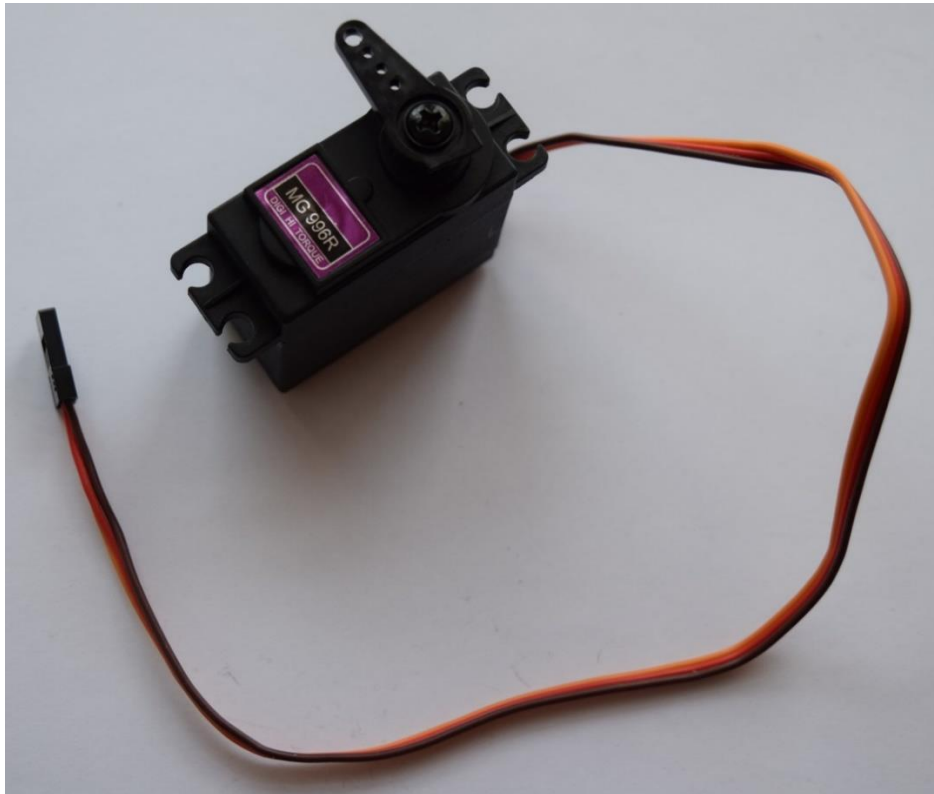


Mise en place de la barre de direction à écartement réglable

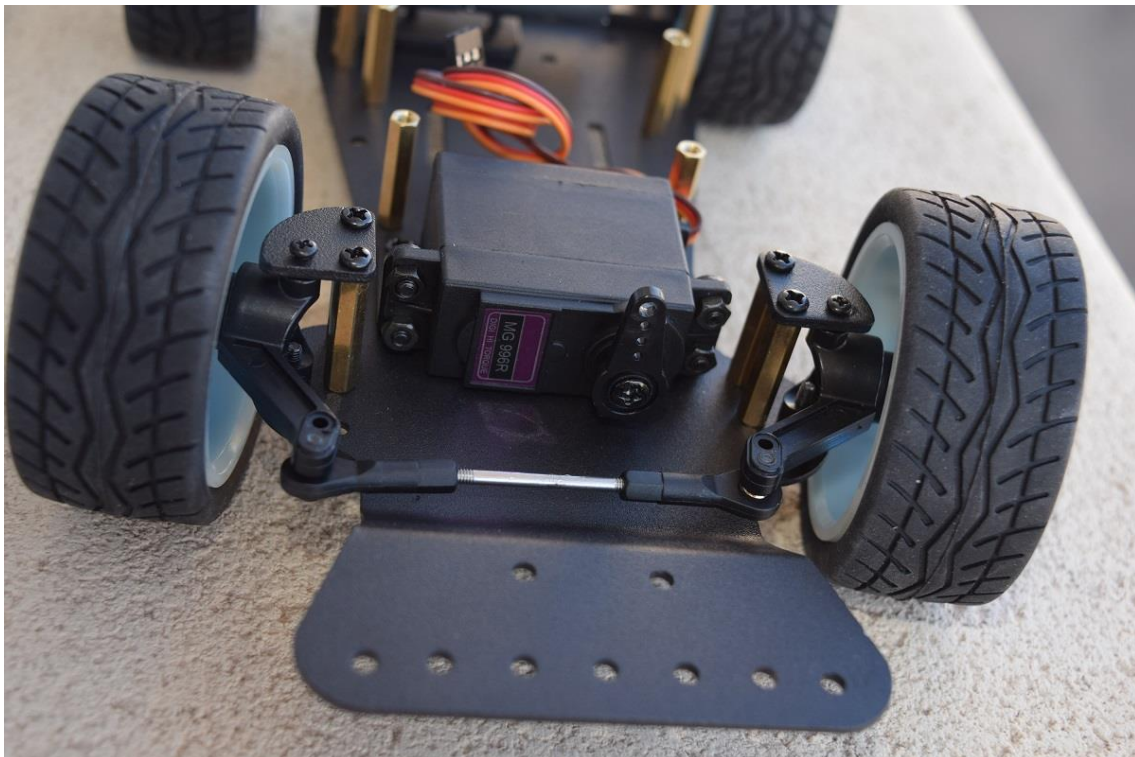


Fixer les équerres de support du servomoteur par 4 vis



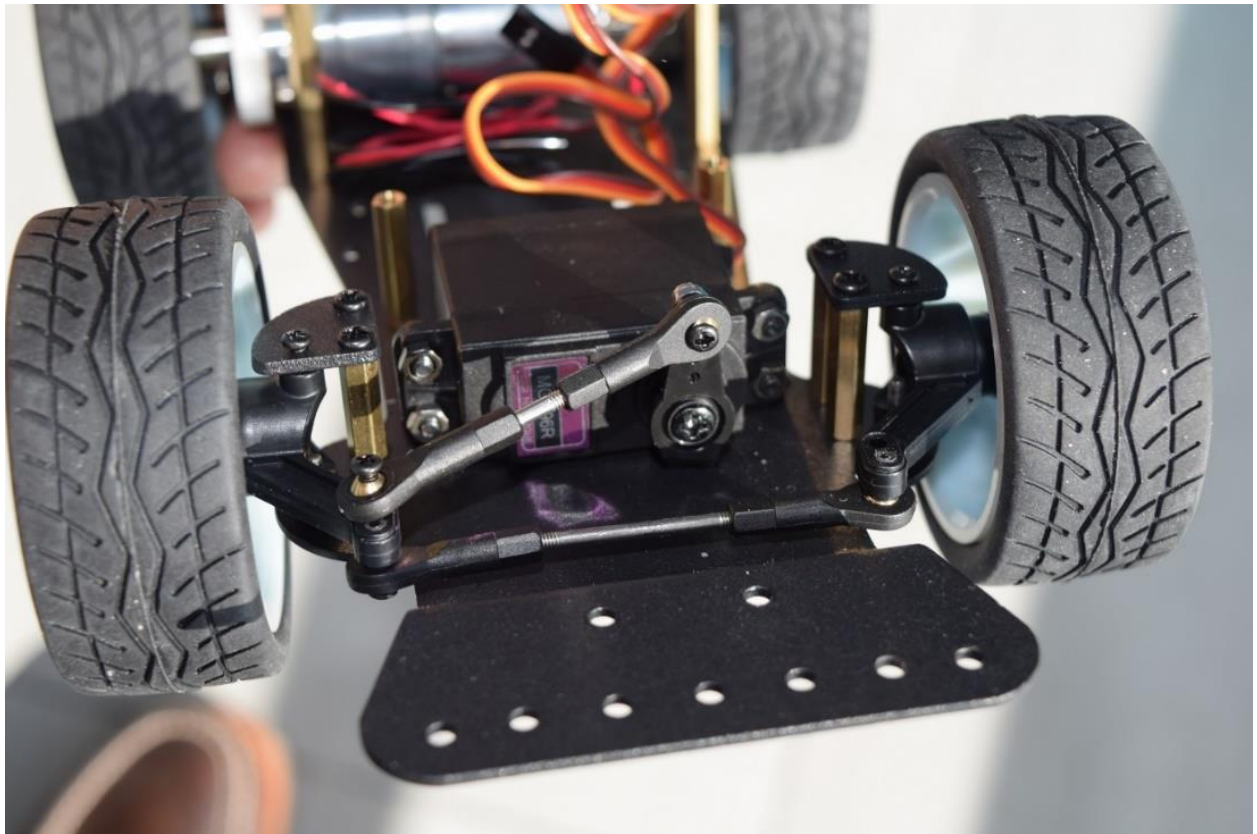


Servomoteur fourni MG 996R

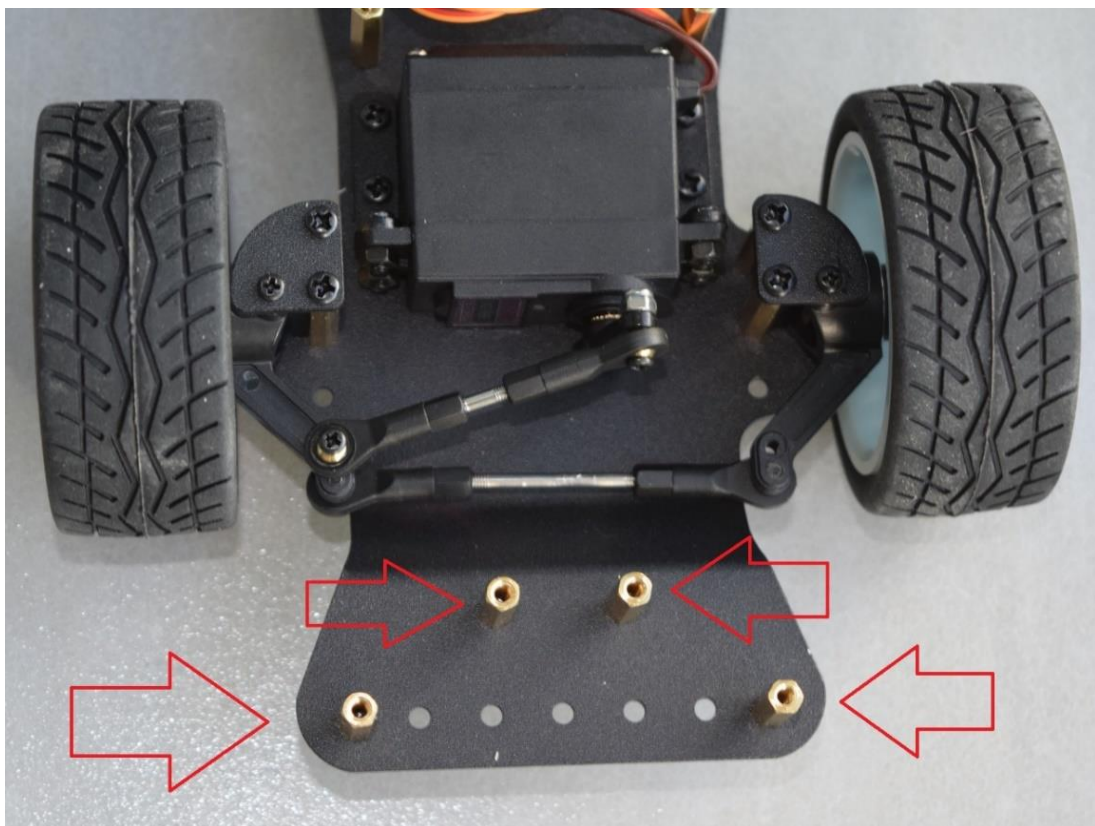


Servomoteur mis en place et solidement fixé par 4 vis + 4 écrous

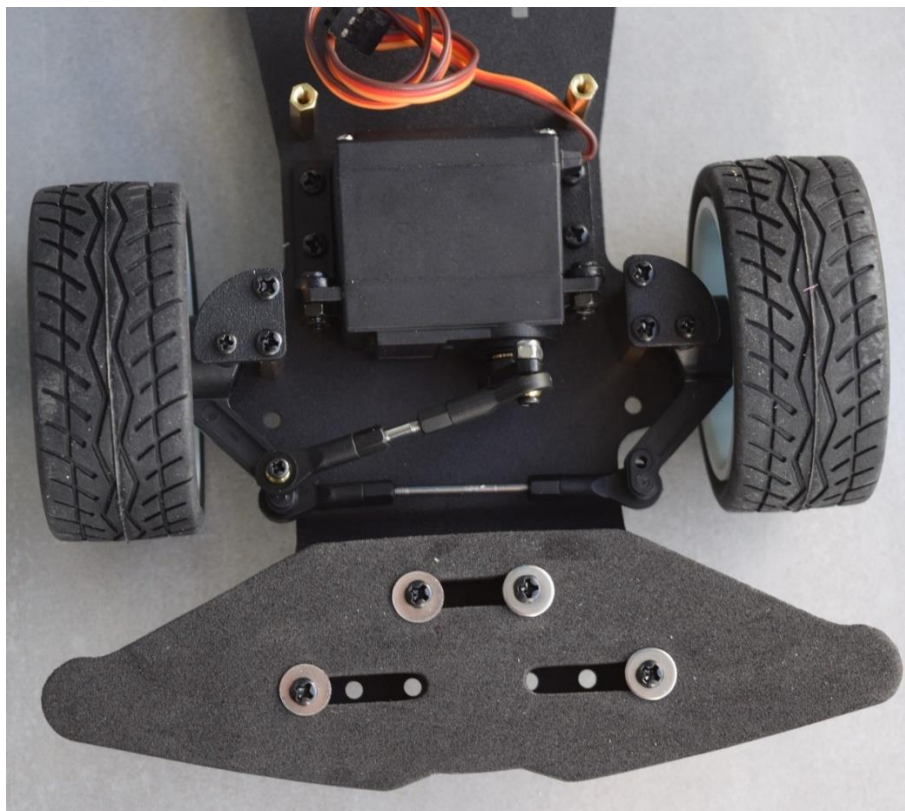
Il faut positionner le bras mobile à mi-chemin de la course du moteur avant de le visser solidement !
Il doit pouvoir tourner approximativement de 90° à gauche et de 90° à droite. Ceci est important pour le futur bon fonctionnement de la direction de la voiture.



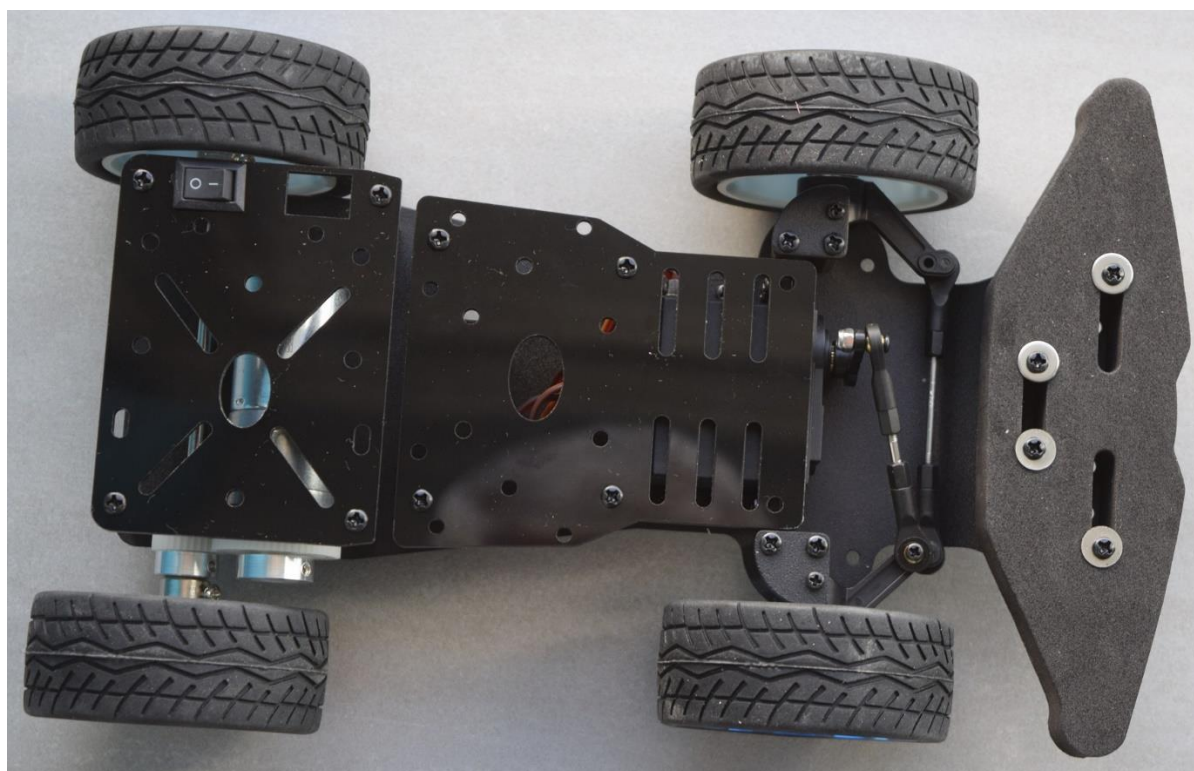
Pose de la biellette réglable reliant le bras du servomoteur à la barre de direction



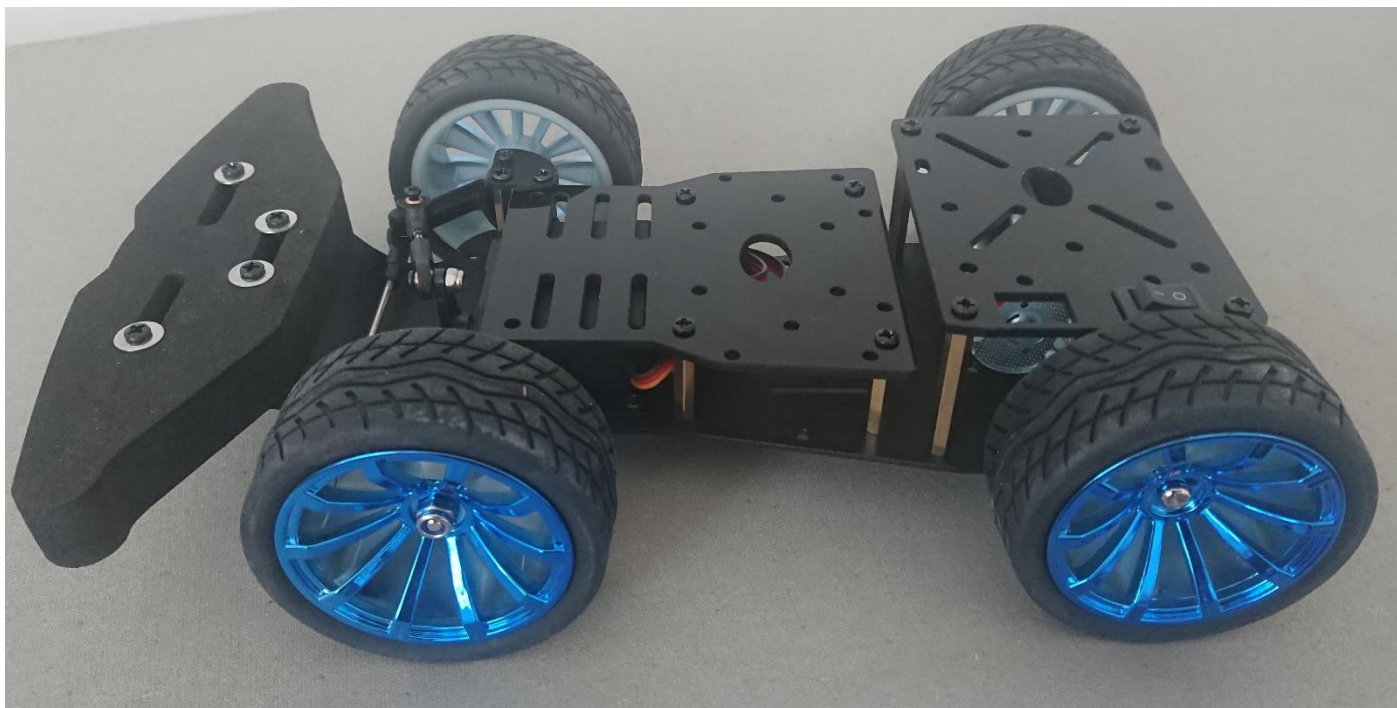
Préparation des supports du bouclier avant



Bouclier avant mis en place et fixé par 4 vis et 4 rondelles



C'est la fin du montage du châssis avec le kit fourni. Ci-dessus, nous avons fixé les deux plaques en résine et placé le petit interrupteur général qui servira à alimenter la voiture Arduino.



Il va falloir à présent équiper cette voiture des composants électroniques nécessaires à son bon fonctionnement puis les relier entre eux et concevoir le programme B4R Arduino qui permettra de la faire fonctionner correctement.

C'est ce que nous allons étudier dans la deuxième partie de cette brochure.



Équipement électronique des voitures

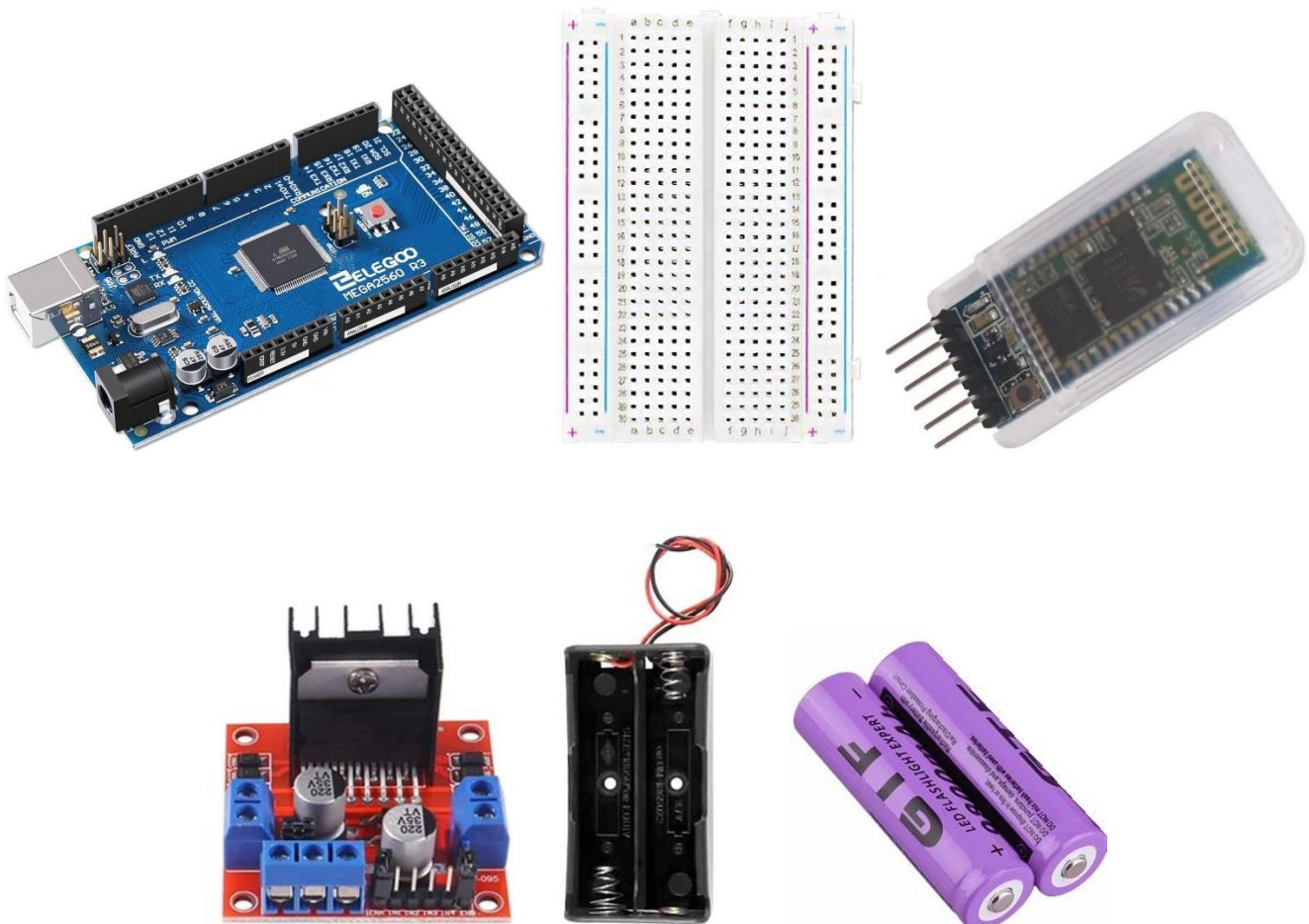
Ce matériel a été construit et conçu pour accueillir une carte ARDUINO UNO utilisée en association avec une carte « shield » fixée sur les broches Arduino Uno :

« YFROBOT PM-R3 Smart Car Drive Board R3 Expansion Board Multi-Function Motor Drive Board » fabriquée par XBee.

Malheureusement, il est actuellement difficile de trouver cette carte « shield » dans le commerce en Europe et les fournisseurs sont presque tous chinois. C'est pourquoi, je vais proposer deux façons d'équiper les voitures Arduino :

- 1- En utilisant une carte ARDUINO MEGA 2560 ou similaire associée à un module L298N >>> Voiture n° 1
- 2- En utilisant une carte ARDUINO UNO en association avec la fameuse carte YFROBOT ce qui est la méthode la plus aboutie au niveau technologie >>> Voiture n° 2

Voiture n° 1 >>> utilisation d'une carte ARDUINO MEGA 2560

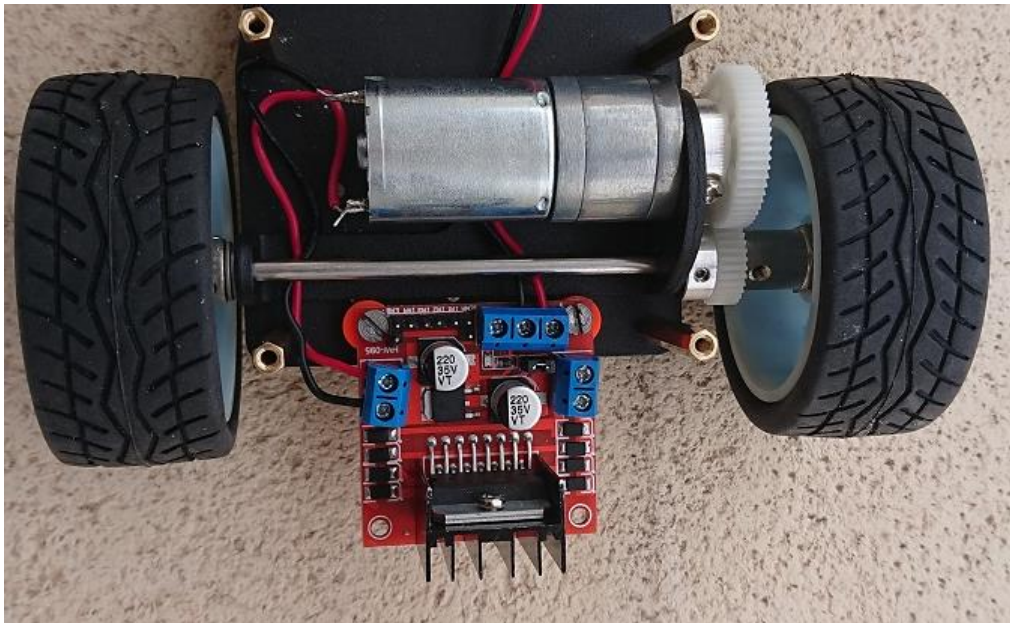




Composants nécessaires

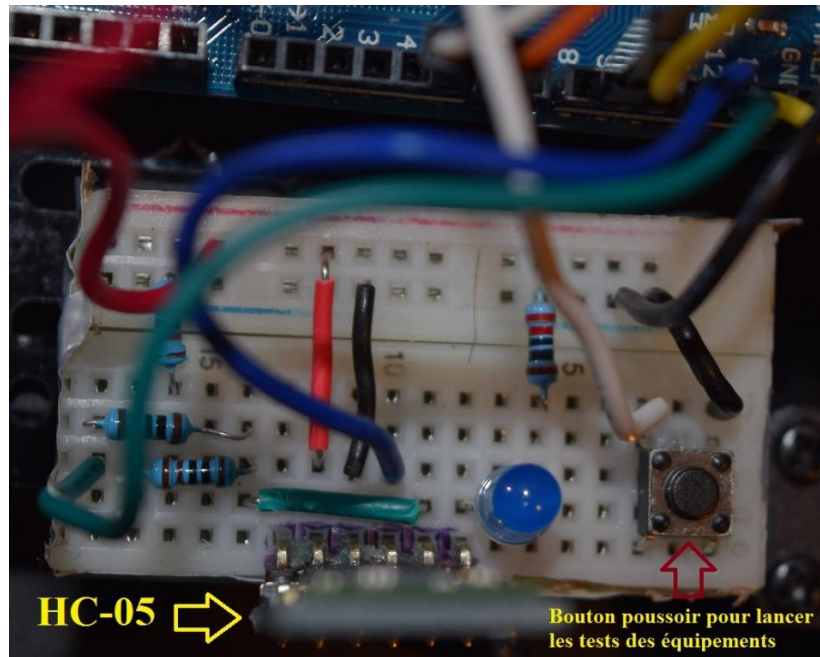
- Une carte Arduino MEGA2560 ou similaire ;
- Une demi-plaque de connexions (que l'on va réduire aux dimensions disponibles au-dessus du servomoteur) ;
- Un module Bluetooth HC-05;
- Un module L298N Dual H-Bridge Motor Driver;
- Un support pour 2 batteries au lithium 18650 Li-ion ;
- Deux batteries rechargeables Li-Ion 18650 de 3,7 volts chacune ;
- 3 résistances de 1 Kohm, 1 résistance de 220 ohms, 1 LED bleue, des LEDs blanches pour placer à l'avant, 2 diodes rouges pour placer à l'arrière ;
- 1 buzzer qui servira de Klaxon ;
- Câbles de couleur, jumpers pour les diverses connexions entre tous ces éléments.

Rappelons que le moteur (propulsion des roues arrière), le servomoteur (gestion de la direction avant) et l'interrupteur général ont été fournis dans le kit d'origine.



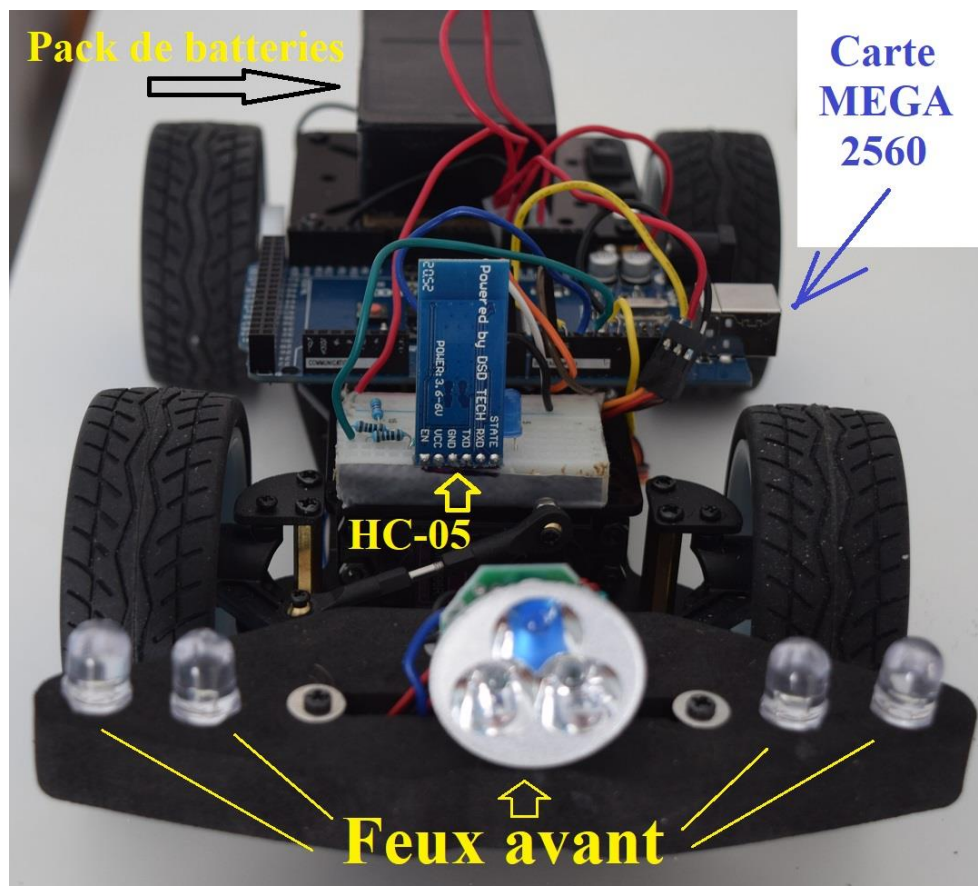
Fixation du module L298N Dual H-Bridge Motor Driver - Connexion du moteur fourni dans le kit

Notons que nous n'utiliserons que la moitié du module L298N : broches ENB, IN3 et IN4 car elles correspondent à l'alimentation du moteur 2, ici à gauche, la plus proche des bornes d'alimentation du moteur. Ce module pourrait donc commander un second moteur 1 en cas de besoin (broches ENA, IN1, IN2 inutilisées qui correspondent à l'alimentation située à droite que nous n'utilisons pas).



Mini-plaque de connexions rapides avec le module Bluetooth HC-05 et bouton de tests

Il vous faudra découper soigneusement ce morceau de plaque avec une scie à métaux par exemple, en respectant les zones de connexions. La LED bleue est le témoin de la connexion Bluetooth entre votre Smartphone et la voiture Arduino, elle s'allumera dès que la connexion Bluetooth aura été établie. La LED bleue est protégée par une résistance de $220\ \Omega$. Trois résistances de $1\ \text{K}\Omega$ sont nécessaires pour la protection du module HC-05 (on les voit sur le côté gauche) pour l'installation correcte du module Bluetooth HC-05 (Voir page suivante le schéma général des connexions).



La carte MEGA2560 est fixée transversalement par 2 vis sur l'arrière de la plaque en résine avant. Des LEDs diverses ont été placées à l'avant. Tous les feux avant et arrière seront connectés à la broche n° 13 de la carte.

Le pack de batteries rechargeables est fixé sur la plaque de résine arrière, près de l'interrupteur général (voir le schéma général des connexions page suivante).

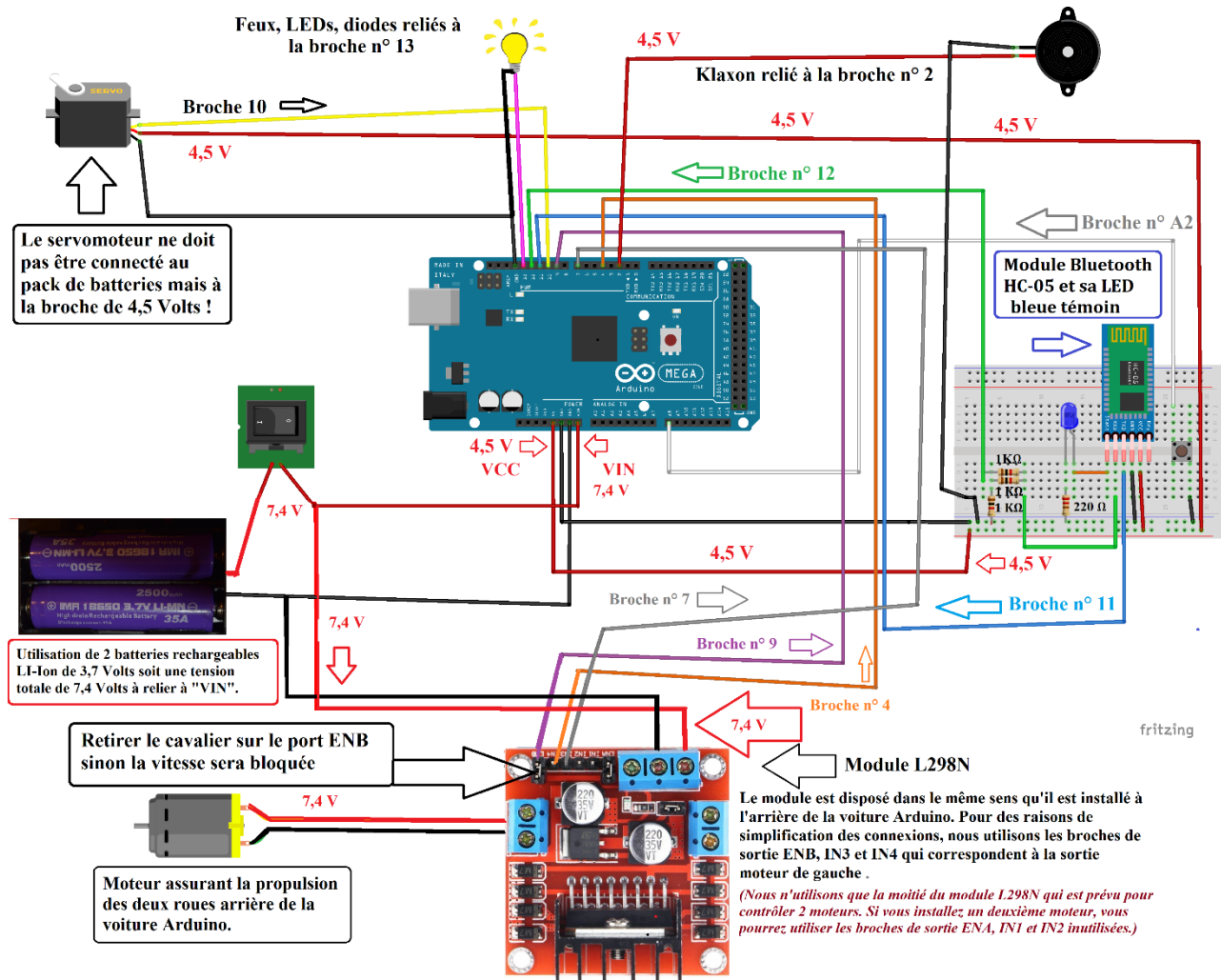
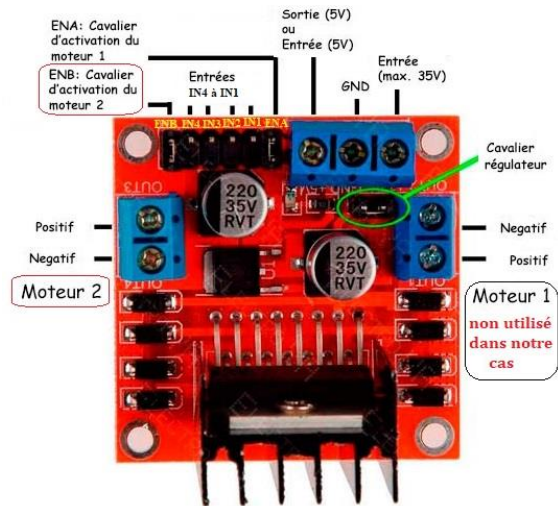
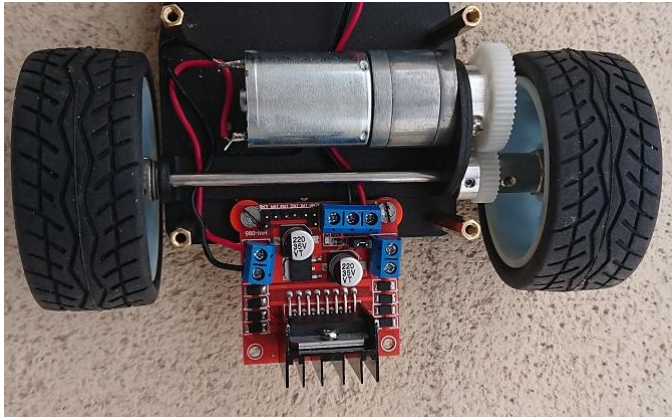


Tableau général des connexions de la voiture Arduino
<https://www.marcalaindaniel.fr/ARDUINO/ArduinoCar/Connexions.pdf>



Détail du programme Visual BASIC B4R pour matériel Arduino **Carte ARDUINO MEGA 2560**

```

1 ' NewCarArduino - VoitureArduino - NewCarArduino.B4R - Marc DANIEL - Novembre 2021
2 #Region Project Attributes
3     #AutoFlushLogs: True
4     #CheckArrayBounds: True
5     #StackBufferSize: 300
6 #End Region7
8 ' NB - L'utilisation de la bibliothèque rServo provoque un bogue sur une carte
9 ' ARDUINO UNO, le servomoteur fonctionne normalement mais les broches PWM
10 ' sont affectées et le moteur arrière ne reçoit plus rien, la voiture reste
11 ' donc immobile! Pour cette raison, je conseille l'utilisation d'une carte
12 ' MEGA2560 sur laquelle le programme est parfaitement exécuté - Marc DANIEL
13
14 ' Using the rServo library causes a bug on an ARDUINO UNO board,
15 ' the servo motor works normally but the PWM pins are affected
16 ' and the rear engine (ENA >>> PWM on D9) no longer receives anything,
17 ' so the car remains motionless! For this reason, I advise the use of
18 ' a MEGA2560 card on which the program is perfectly executed - Marc DANIEL
19
20 Sub Process_Globals
21     Public Serial1 As Serial
22     Private SoftwareSerial1 As SoftwareSerial
23     Private astream As AsyncStreams
24     ' Private IN1, IN2, ENA as Pin
25     ' Pins of the L298N to be used for a second engine
26     Private BtnTest, IN3, IN4, ENB, Lights, Klaxon As Pin
27     Private SpeedM As UInt 'Vitesse de la voiture - car speed
28     Private X=0, Y=0 As UInt
29     Private FEUX As Boolean ' Eclairage, LEDs, diodes
30     ' Lighting leds diodes
31     Public MonServo As Servo 'Servomoteur pilotant la direction
32     ' Servo motor controlling the direction of the front wheels
33     Private SON As Boolean ' Klaxon ou Buzzer si installé
34     ' Horn or Buzzer if installed
35 End Sub36
37 Private Sub AppStart
38     Serial1.Initialize(115200)
39     Log("Démarrage de la voiture")

```

```

40 IN3.Initialize(4, IN3.MODE_OUTPUT) ' Marche avant - Forward
41 IN4.Initialize(7, IN4.MODE_OUTPUT) ' Marche arrière - Backward
42 ENB.Initialize(9, ENB.MODE_OUTPUT)'Envoi du signal PWM vers le moteur
43 ' Sending the PWM signal to the rear engine
44 Lights.Initialize(13, Lights.MODE_OUTPUT)
45 'Tous les feux, LEDs blanches avant, diodes rouges arrière, doivent
46 'être reliées à la broche 13
47 'All lights, front white LEDs, rear red diodes
48 ' must be connected to pin 13
49 'Si aucun feu n'est connecté, il est possible de vérifier l'allumage
50 ' de la petite LED jaune sur la carte Arduino elle-même.
51 'If no light is connected, it is possible to check the ignition
52 ' of the small yellow LED on the Arduino board itself
53 Klaxon.Initialize(2, Klaxon.MODE_OUTPUT)
54 ' gestion de l'avertisseur sonore ou buzzer si installé
55 ' management of the audible alarm or buzzer if installed
56 MonServo.Attach2(10,30,130) 'Servomoteur contrôlant la direction
57 ' 30° est la valeur minimale, 130° est la valeur maximale
58 ' Servo motor control the front steering
59 ' 30° is the minimum value, 130° is the maximum value
60 BtnTest.Initialize(BtnTest.A0,BtnTest.MODE_INPUT_PULLUP)
61 ' Bouton poussoir qui lance le test des moteurs
62 ' Push button that starts the engine test
63 BtnTest.AddListener("BtnTEST_StateChanged")
64 SoftwareSerial1.Initialize(9600,11,12)
65 ' Software Serial port sur les broches 11 et 12
66 ' Software Serial port on pins 11 and 12 -
67 'Broche TXD de HC-05 directement reliée à Arduino 11
68 'Broche RXD de HC-05 protégée par 3 résistances de 1 Ko reliée à Arduino 12
69 astream.Initialize(SoftwareSerial1.Stream, "astream_NewData", Null)
70 End Sub
71
72 Private Sub BtnTEST_StateChanged(State As Boolean) 'Appui sur le bouton de test
73 ' Press the test button
74 If State = False Then 'BoutonTEST=0
75     Log("Bouton test: ", State)
76     Depart
77 Else 'BoutonTEST=1
78     Log("Bouton test: ", State)
79 End If
80 End Sub
81
82 Sub Depart 'tests de fonctionnement des équipements après appui sur le petit bouton test
83 'equipment operation tests after pressing the small test button
84 For x=1 To 3
85     Lights.DigitalWrite(True) 'Test des feux si installés
86     '(sinon la LED 13 de la carte permettra de vérifier le test)
87     'Test of the lights if installed
88     '(otherwise the LED 13 of the card will check the test)
89     Log("Allumage des feux")
90     FEUX = True
91     Klaxon.DigitalWrite(True) 'Test du Klaxon ou Buzzer si installé
92     ' Test the Horn or Buzzer if installed
93     SON = True
94     Log("Déclenchement du Klaxon")
95     Delay(1000)
96     Lights.DigitalWrite(False)
97     FEUX = False '(Lumières éteintes par défaut)
98     '(Lights off by default)
99     Log("Extinction des feux")

```

```

100     Klaxon.DigitalWrite(False)
101     SON = False
102     Log("Arrêt du Klaxon") '(Klaxon à l'arrêt par défaut)
103     '(Horn at idle by default)
104     Delay(1000)
105 Next
106 MonServo.Write(120) 'Tests sur la direction commandée par le Servomoteur
107 ' Tests on the steering controlled by the Servo motor
108 Log("Roues avant braquées à droite")
109 Delay(2000)
110 MonServo.Write(35)
111 Log("Roues avant braquées à gauche")
112 Delay(2000)
113 MonServo.Write(90) 'Les roues avant sont alignées en ligne droite
114 ' (Angle du bras orienté à 90°)
115 'The front wheels are aligned in a straight line (90° arm angle)
116 Log("Alignement des roues avant")
117 'Test moteur arrière (Attention la voiture va avancer puis reculer)
118 ''Rear engine test (Be careful the car will move forward and then backward)
119 IN3.DigitalWrite(True)
120 IN4.DigitalWrite(False)
121 SpeedM = 100
122 Envoi
123 Delay(3000)
124 IN3.DigitalWrite(False)
125 IN4.DigitalWrite(True)
126 SpeedM = 100
127 Envoi
128 Delay(3000)
129 SpeedM=0 'Vitesse ramenée à zéro par défaut
130 ' Speed reduced to zero by default
131 Envoi
132 For x=1 To 3
133     Lights.DigitalWrite(True) 'Test des feux si installés
134     '(sinon la LED 13 de la carte permettra de vérifier le test)
135     'Test of the lights if installed
136     '(otherwise the LED 13 of the card will check the test)
137     Log("Allumage des feux")
138     FEUX = True
139     Klaxon.DigitalWrite(True) 'Test du Klaxon ou Buzzer si installé
140     ' Test the Horn or Buzzer if installed
141     SON = True
142     Log("Déclenchement du Klaxon")
143     Delay(1000)
144     Lights.DigitalWrite(False)
145     FEUX = False '(Lumières éteintes par défaut)
146     '(Lights off by default)
147     Log("Extinction des feux")
148     Klaxon.DigitalWrite(False)
149     SON = False
150     Log("Arrêt du Klaxon") '(Klaxon à l'arrêt par défaut)
151     '(Horn at idle by default)
152     Delay(1000)
153 Next
154 End Sub 'Fin des tests de fonctionnement des équipements
155 ' End of equipment operation tests
156
157 Sub AStream_NewData (Buffer() As Byte)
158     'Fonctionnement de la voiture pilotée par Bluetooth
159     ' How the car is operated by Bluetooth
160     If Buffer.Length = 2 Then
161         X= Buffer(0)

```

```

162     Log("X: ",X)
163     Delay(10)
164     Y=Buffer(1)
165     Log("Y: ",Y)
166
167     If Y=2 And X= 2 Then
168         'Alignement des roues avant en ligne droite
169         'Alignment of the front wheels in a straight line
170         Log("Rouler tout droit")
171         MonServo.write(90)
172
173     Else If Y=4 And X= 4 Then
174         'Rotation légère à gauche - Slight rotation on the left
175         Log("Rotation légère à gauche")
176         MonServo.write(70)
177
178     Else If Y=5 And X= 5 Then
179         'Rotation nette à gauche - Net rotation on the left
180         Log("Rotation nette à gauche")
181         MonServo.write(35)
182
183     else if Y=6 And X=6 Then
184         'Rotation légère à droite - Light rotation to the right
185         Log("Rotation légère à droite")
186         MonServo.write(110)
187
188     else if Y=7 And X=7 Then
189         'Rotation nette à droite - Net rotation to the right
190         Log("Rotation nette à droite")
191         MonServo.write(125)
192
193
194
195     Else If Y < 117 And Y > 10 Then
196         ' MARCHE AVANT - FORWARD MARCH
197         'lance la voiture en avant - launches the car forward
198         IN3.DigitalWrite(True)
199         IN4.DigitalWrite(False)
200         SpeedM = 100
201         Log("Marche avant")
202         Envoi
203
204
205     else if Y > 137 Then
206         'MARCHE ARRIÈRE - BACKWARD
207         'lance la voiture en arrière - throws the car back
208         IN3.DigitalWrite(False)
209         IN4.DigitalWrite(True)
210         SpeedM = 90
211         Log("Marche arrière")
212         Envoi
213
214     Else
215         SpeedM=0
216         Log("arrêt de la voiture")
217         Envoi
218     End If
219 End If
220
221 If Y = 250 And X = 250 Then
222     ' MARCHE AVANT rapide de la voiture - Fast FORWARD OF THE CAR

```

```

223     IN3.DigitalWrite(True)
224     IN4.DigitalWrite(False)
225     SpeedM = 180
226     Log("Marche avant rapide")
227     Envoi
228
229     else if Y=200 And X=200 Then
230         'MARCHE ARRIÈRE rapide de la voiture - Fast REVERSE of the car
231         IN3.DigitalWrite(False)
232         IN4.DigitalWrite(True)
233         SpeedM = 110
234         Log("Marche arrière rapide")
235         Envoi
236
237     else if Y=9 And X=9 Then
238         ' GESTION des FEUX, LEDS, DIODES et lumières diverses si connectés broche 13
239         ' MANAGEMENT of LIGHTS, LEDS, DIODES and various lights if connected pin 13
240         If FEUX=False Then
241             Lights.DigitalWrite(True)
242             FEUX = True
243             Log("Allumage des LEDS")
244             'Allumage des LEDS si elles existent et ont été connectées sur la broche 13
245             'Ignition of LEDs if they exist and have been connected on pin 13
246         Else
247             Lights.DigitalWrite(False)
248             FEUX = False
249             Log("Extinction des LEDS")
250             'Extinction des LEDS si elles existent et ont été connectées sur la broche 13
251             'Turning off LEDs if they exist and have been connected on pin 13
252         End If
253
254     else if Y = 3 And X = 3 Then ' GESTION du «Klaxon» ou buzzer si connecté à la broche 3
255         'MANAGEMENT of the "Horn" or buzzer or similar device if connected to pin 3
256         If SON = False Then
257             Klaxon.DigitalWrite(True)
258             SON = True
259             Log("Déclenchement du Klaxon")
260         Else
261             Klaxon.DigitalWrite(False)
262             SON = False
263             Log("Arrêt du Klaxon")
264         End If
265
266     else if Y=8 And X = 8 Then
267         GrandHuit
268     End If
269 End Sub
270
271 Sub GrandHuit
272     ' La voiture va effectuer seule les manoeuvres sur un circuit de plusieurs mètres
273     ' de préférence à l'extérieur
274     'The car will make the trips alone on a circuit of several meters preferably outside
275     MonServo.Write(90)
276     IN3.DigitalWrite(True)
277     IN4.DigitalWrite(False)
278     SpeedM = 140
279     Envoi 'Démarrage - Startup
280     Delay(3000)
281     MonServo.Write(30)
282     Delay(3000)
283     MonServo.Write(90)

```

```

284 Delay(4000)
285 MonServo.Write(125)
286 Delay(3000)
287 MonServo.Write(90)
288 Delay(4000)
289 SpeedM = 0
290 Envoi 'Fin du circuit - End of the circuit'
291 End Sub
292
293 Private Sub Envoi 'Gestion vitesse de la voiture - Car speed management
294 ENB.analogWrite(SpeedM)
295 Log("Vitesse de la voiture:",SpeedM)
296 End Sub

```

Fin du programme B4R pour voiture n° 1

Téléchargement du programme B4R pour carte ARDUINO MEGA 2560 :

<https://www.marcalaindaniel.fr/ARDUINO/ArduinoCar/Programmes/NewCarArduino.zip>

Maintenant que nous disposons du programme qui va contrôler la voiture Arduino n° 1 et que celle-ci est construite et équipée de tous les éléments nécessaires à son bon fonctionnement, il est déjà possible de vérifier si tout fonctionne bien en effectuant les opérations suivantes :

- Téléverser le programme B4R dans la carte MEGA2560 à l'aide d'un câble USB approprié ;
- Mettre sous tension la voiture en utilisant le petit interrupteur arrière ;
- Presser sur le petit bouton de tests situé sur la mini-plaque de connexions :

Si tout est bien en place, la voiture va effectuer la série de tests prévus par l'algorithme du programme des ligne 82 à 155, c'est-à-dire que vous devriez assister successivement à 3 appels de phares (LEDs avant, Diodes rouges arrière) accompagnés par 3 sons de Klaxon (Buzzer) puis les roues avant vont se braquer à droite puis à gauche, puis les roues seront ramenées en ligne droite et ATTENTION ! la voiture va rouler en avant pendant 3 secondes puis reculer également pendant 3 secondes. Les tests se termineront par 3 nouveaux appels de phares + 3 coups de Klaxon.

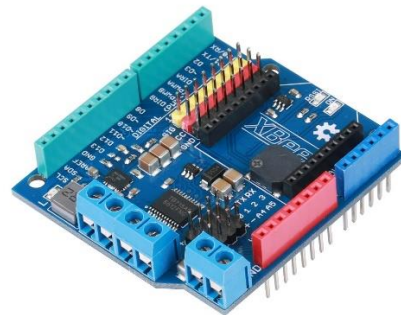
Ainsi, sans même attendre la mise au point de l'application B4A pour Smartphone Android qui va nous permettre prochainement de piloter à distance cette voiture grâce au système Bluetooth, il est possible de faire bouger la voiture et de vérifier qu'elle est prête pour continuer les travaux.

Équipement électronique des voitures

Voiture n° 2 >>> Utilisation d'une carte ARDUINO UNO

C'est la solution que je préconise, même s'il faut du temps pour obtenir la carte « shield » YFROBOT, car la structure a été prévue au départ pour la fixation d'une carte ARDUINO UNO et l'association avec la carte YFROBOT PM-R3 qui va permettre de contrôler très facilement le moteur arrière et le servomoteur avant. De plus, cette méthode a le grand avantage de diminuer spectaculairement le nombre de câbles de connexion. Tout est connecté simplement sur la carte YFROBOT (qui remplace le module L298N) : l'alimentation générale en provenance du pack de batteries rechargeables, le moteur arrière, le servomoteur.

Composants nécessaires



- Une carte Arduino UNO ou similaire ;
- Une carte shield YFROBOT PM-R3 servant à contrôler le moteur arrière et le servomoteur avant

Liens vers des fournisseurs:

<https://www.sinoning.com/product/yfrobot-pm-r3-smart-car-drive-board-r3-expansion-board-multi-function-motor-drive-board/>

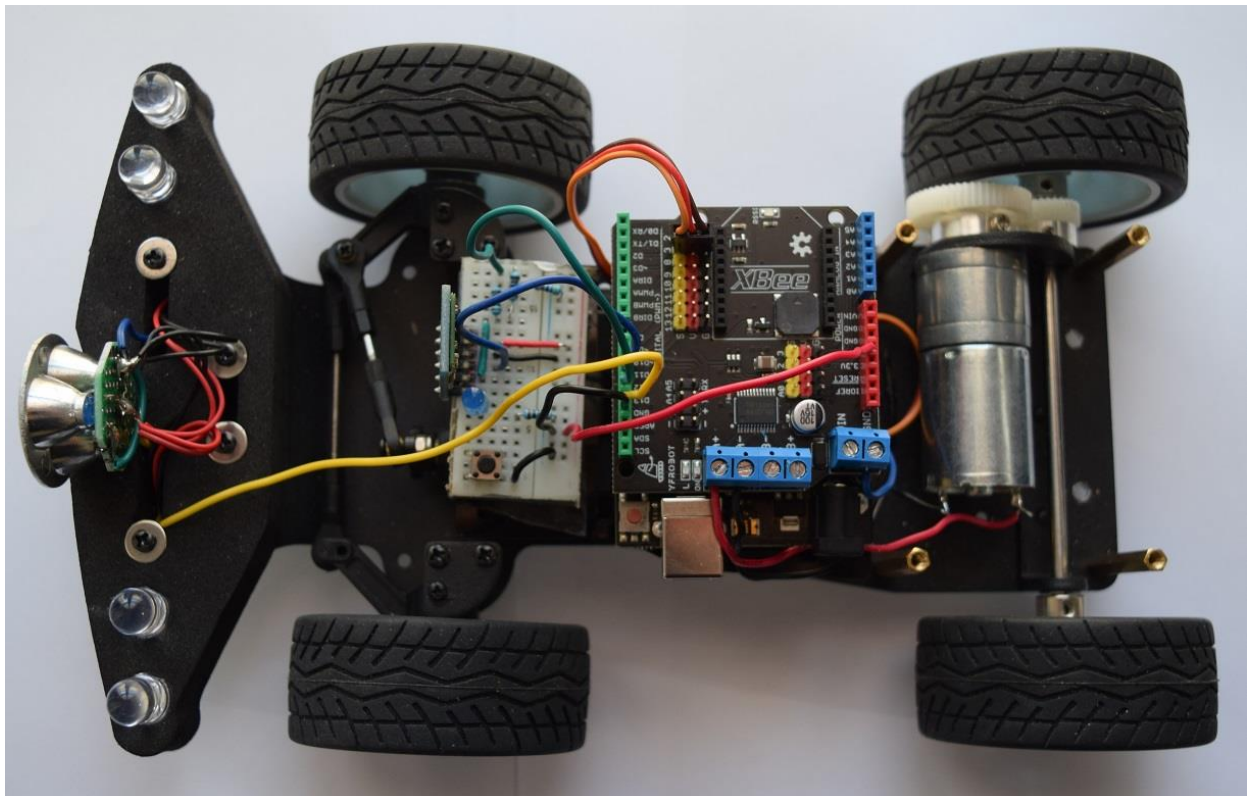
<https://www.ebay.fr/itm/182826494617>

- Une demi-plaque de connexions (que l'on va réduire aux dimensions disponibles au-dessus du servomoteur) ;
- Un module Bluetooth HC-05;
- Un support pour 2 batteries au lithium 18650 Li-ion ;
- Deux batteries rechargeables Li-Ion 18650 de 3,7 volts chacune ;
- 3 résistances de 1 Kohm, 1 résistance de 220 ohms, 1 LED bleue, des LEDs blanches pour placer à l'avant, 2 diodes rouges pour placer à l'arrière ;
- 1 buzzer qui servira de Klaxon ;
- Câbles de couleur, jumpers pour les diverses connexions entre tous ces éléments.

Rappelons que le moteur (propulsion des roues arrière), le servomoteur (gestion de la direction avant) et l'interrupteur général ont été fournis dans le kit d'origine.

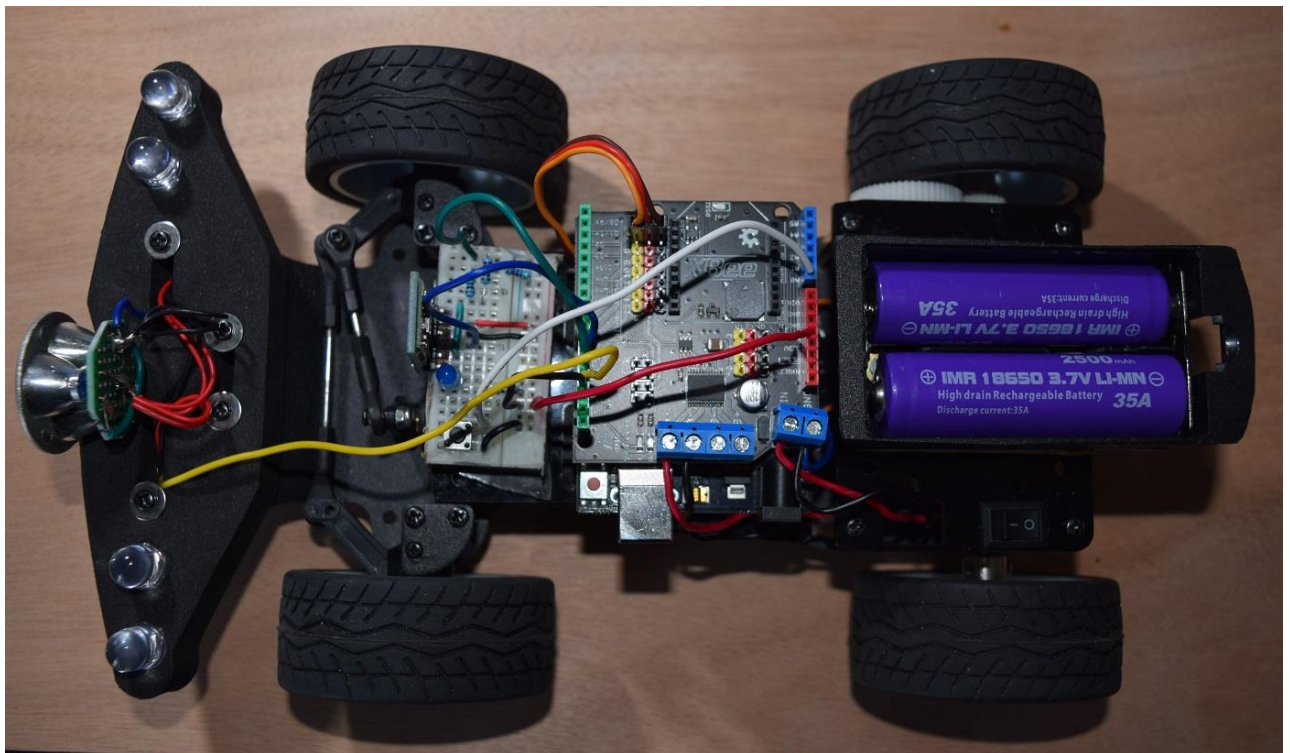


Installation des 2 cartes superposées et connexion du moteur arrière

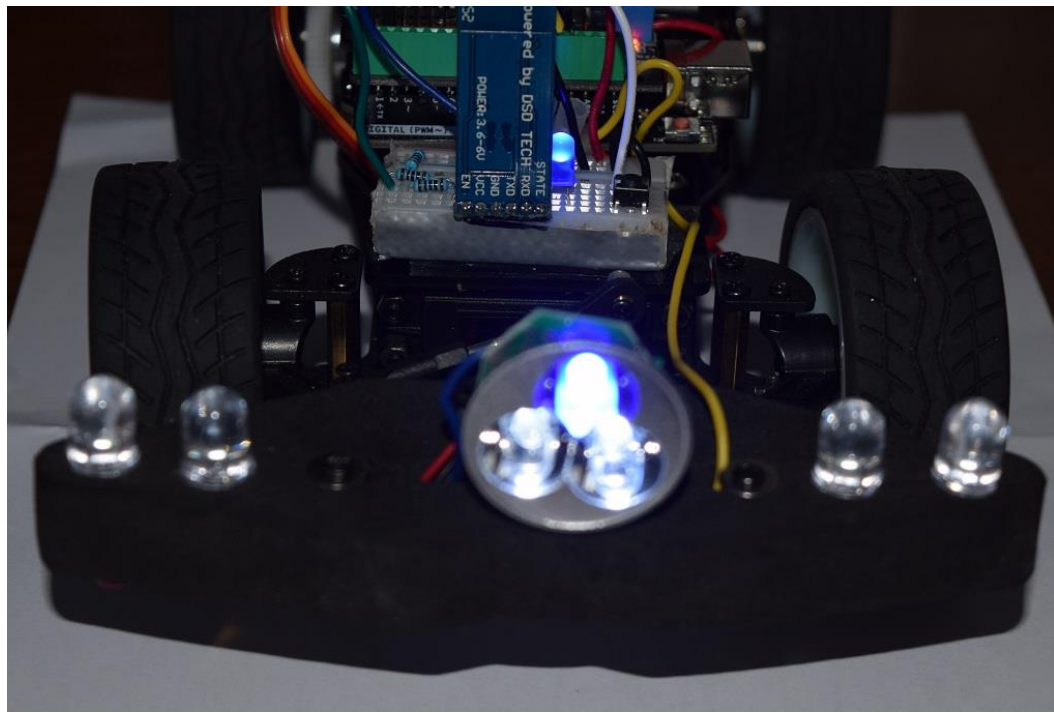


Notez que les 3 câbles jaune-rouge-brun du servomoteur sont connectés sur la 2^{ème} ligne de broches tricolores en haut et à gauche (n°3) de la carte YFROBOT.

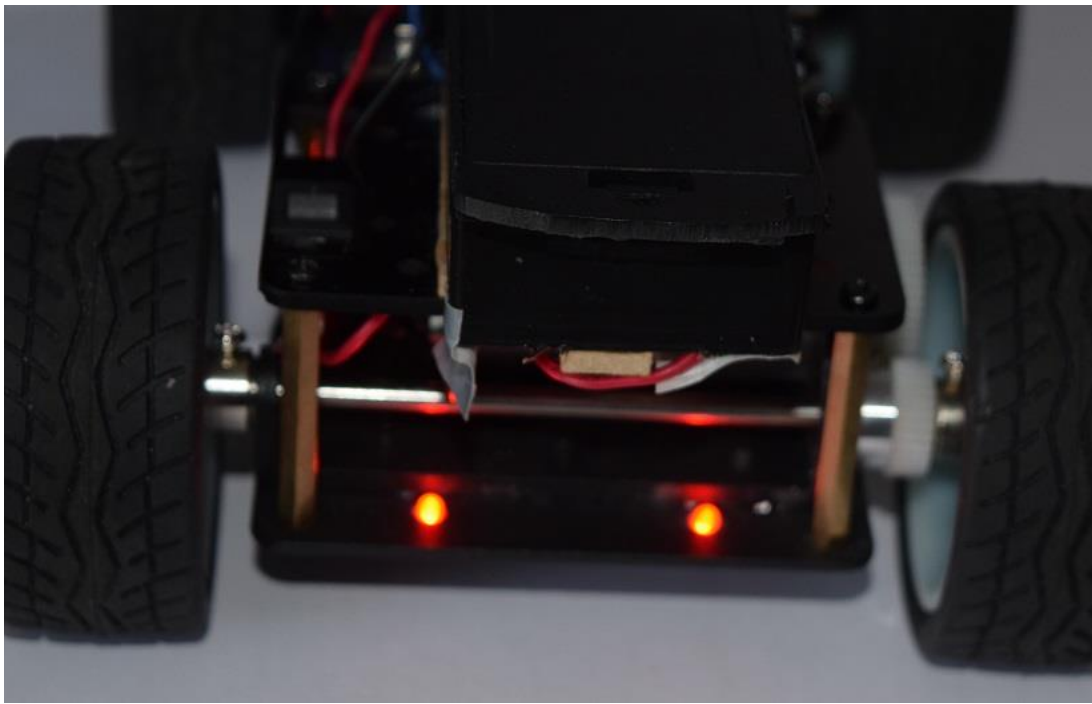
Je ne reviens pas sur l'installation du module Bluetooth et du bouton de lancement des tests, c'est exactement identique à l'installation avec Arduino Mega 2560. (Voir plus haut)



Installation des batteries sur la plaque arrière avec l'interrupteur général



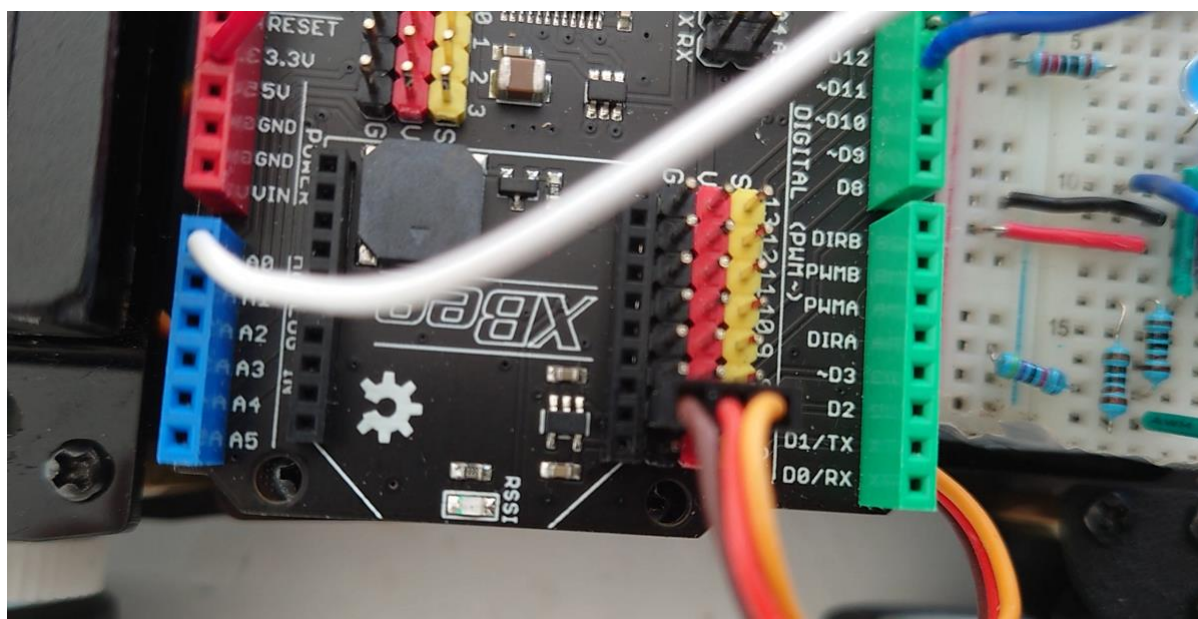
Feux avant : 4 grosses LEDs blanches + 3 LEDs centrales >>> D13



Feux rouges arrière >>> D13

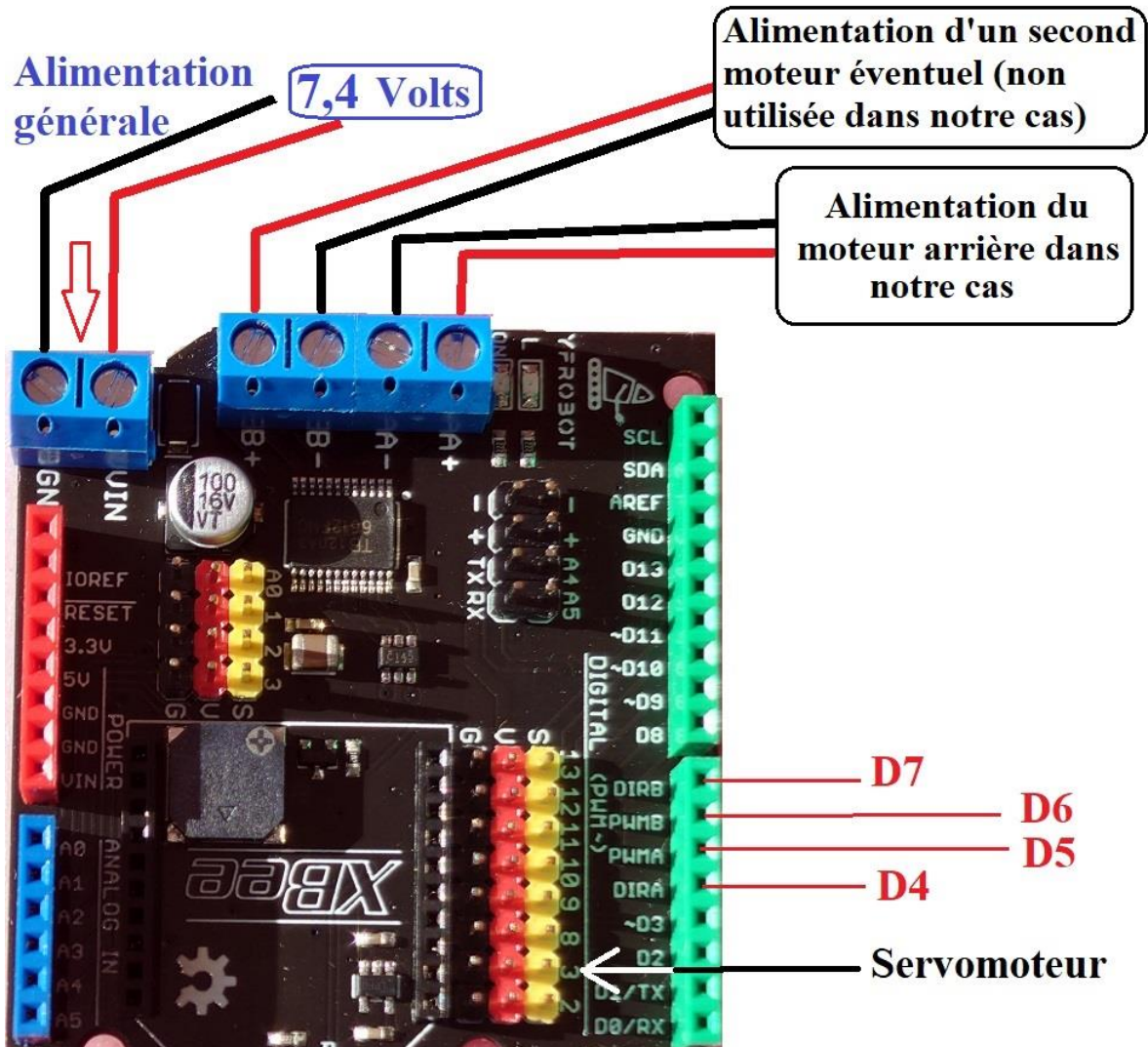
Zoom sur les connexions >>>

https://www.marcalaindaniel.fr/ARDUINO/ArduinoCar/YFROBOTarduinoCar/AvecUNO_et_YFROBOTshield.mp4



Carte Shield YROBOT XBee Motor Driver Board

utilisée clipée sur une carte Arduino UNO R3



Détail des broches et connexions de la carte YFROBOT

Cette carte qui remplace le module L298N permet en fait de contrôler deux moteurs. Dans notre cas, nous n'utilisons donc que les broches D4, D5 et D7 puisque nous n'utilisons qu'un seul moteur (Le moteur 1 en quelque sorte).

La carte permet la connexion très simple du servomoteur sur la ligne de broches tricolores (Noir-Rouge-Jaune). Dans notre cas, il faut utiliser la 2^{ème} ligne correspondante à la broche D3.

Noir (Masse GND) >>> câble brun du servomoteur

Rouge (VCC +) >>> câble rouge du servomoteur

Jaune (Signal = valeur d'angle) >>> câble jaune du servomoteur

L'alimentation générale du système se fait directement et uniquement sur les entrées VIN et GND de cette carte à partir du pack de batteries utilisées. Le gros avantage d'utiliser cette carte réside dans le fait que l'on simplifie au maximum le nombre de câbles utilisés et de connexions qui sont réduites au minimum par rapport à la version « ARDUINO MEGA 2560 » qui nécessite une « forêt » de câbles.

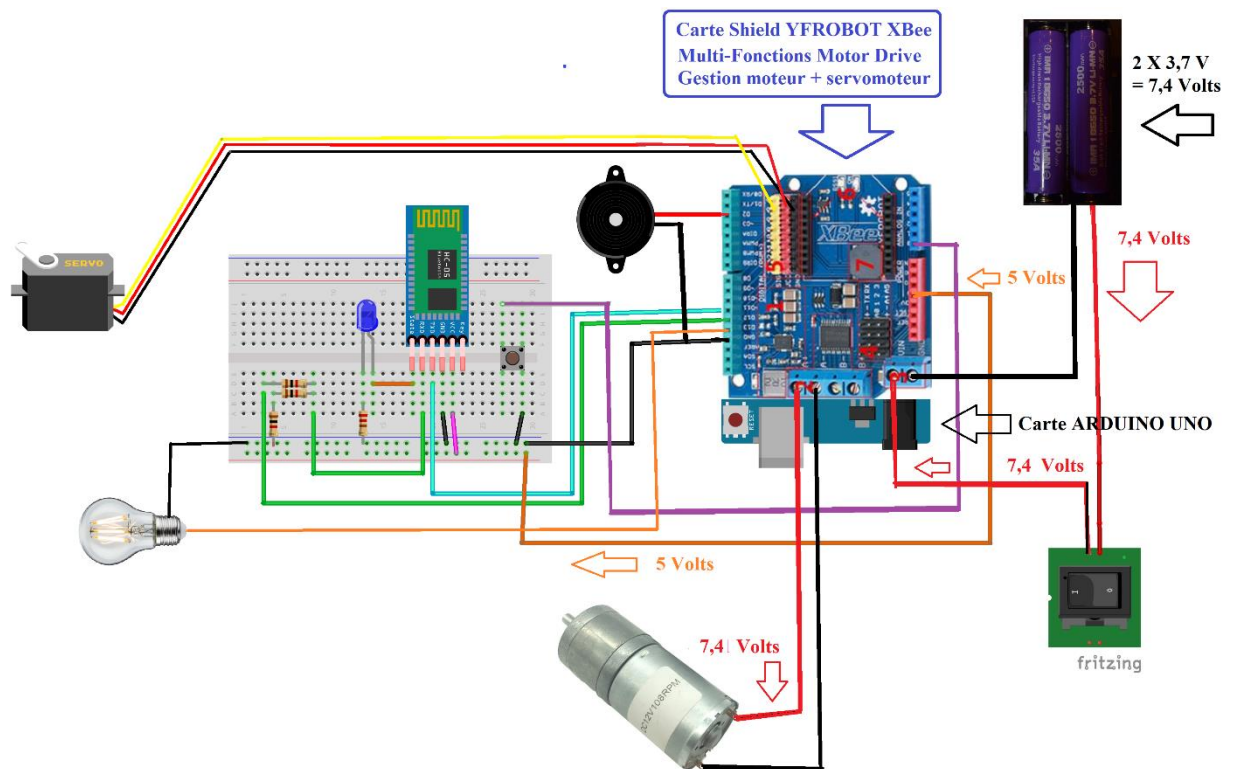


Schéma général des connexions

https://www.marcalaindaniel.fr/ARDUINO/ArduinoCar/YFROBOTarduinoCar/YFROBOT_Fritzing.pdf

Téléchargement du programme B4R pour carte ARDUINO UNO – Voiture n° 2 :

https://www.marcalaindaniel.fr/ARDUINO/ArduinoCar/YFROBOTarduinoCar/4WD_YFROBOT.zip

Le nouveau programme B4R a été adapté aux nouveaux composants utilisés mais l'algorithme est sensiblement le même que celui du programme précédent détaillé des pages 15 à 20.

Les changements essentiels sont dus aux spécificités de la carte YFROBOT utilisée dont les broches sont préconnectées en circuit interne. J'ai modifié les noms utilisés pour simplifier la programmation et l'adapter aux indications portées sur la carte elle-même. Ainsi IN3, IN4, ENB sont devenus DIR1, DIR2, PWMA ...

Broches de direction :

DIR1 >>> Broche D4

DIR2 >>> Broche D7

Vitesse du moteur :

PWMA >>> Broche D5

(Notons que si nous utilisons un second moteur, sa vitesse serait gérée par PWMB à la broche D6)

Servomoteur >>> Broche D3

Pas de changements pour la gestion des feux (Broche D13) et du Klaxon (Broche D2) ni pour le bouton de test connecté à la broche analogique A0. Le module Bluetooth HC-05 est toujours connecté aux broches D11 et D12.

```

10 Sub Process_Globals
11   Public Serial1 As Serial
12   Private SoftwareSerial1 As SoftwareSerial
13   Private astream As AsyncStreams
14   Private BtnTest, DIR1, DIR2, PWMA, Lights, Klaxon As Pin
15   'Private PWMB as pin
16   Private Speed As UInt
17   Private X=0, Y=0 As UInt
18   Private FEUX As Boolean
19   Public MonServo As Servo
20   Private SON As Boolean
21 End Sub
22
23 Private Sub AppStart
24   Serial1.Initialize(115200)
25   Log("Démarrage")
26   DIR1.Initialize(4, DIR1.MODE_OUTPUT) ' Marche avant - Forward
27   DIR2.Initialize(7, DIR2.MODE_OUTPUT) ' Marche arrière - Backward
28   PWMA.Initialize(5, PWMA.MODE_OUTPUT) ' moteur arrière
29   'PWMB.Initialize(6, PWMB.MODE_OUTPUT) ' en cas de second moteur arrière
30   MonServo.Attach2(3,30,130) ' Direction by servo motor
31   Lights.Initialize(13, Lights.MODE_OUTPUT)
32   Klaxon.Initialize(2, Klaxon.MODE_OUTPUT)
33   BtnTest.Initialize(BtnTest.A0, BtnTest.MODE_INPUT_PULLUP)
34   BtnTest.AddListener("BtnTEST_StateChanged")
35   SoftwareSerial1.Initialize(9600, 11, 12)
36   astream.Initialize(SoftwareSerial1.Stream, "astream_NewData", Null)
37 End Sub
38

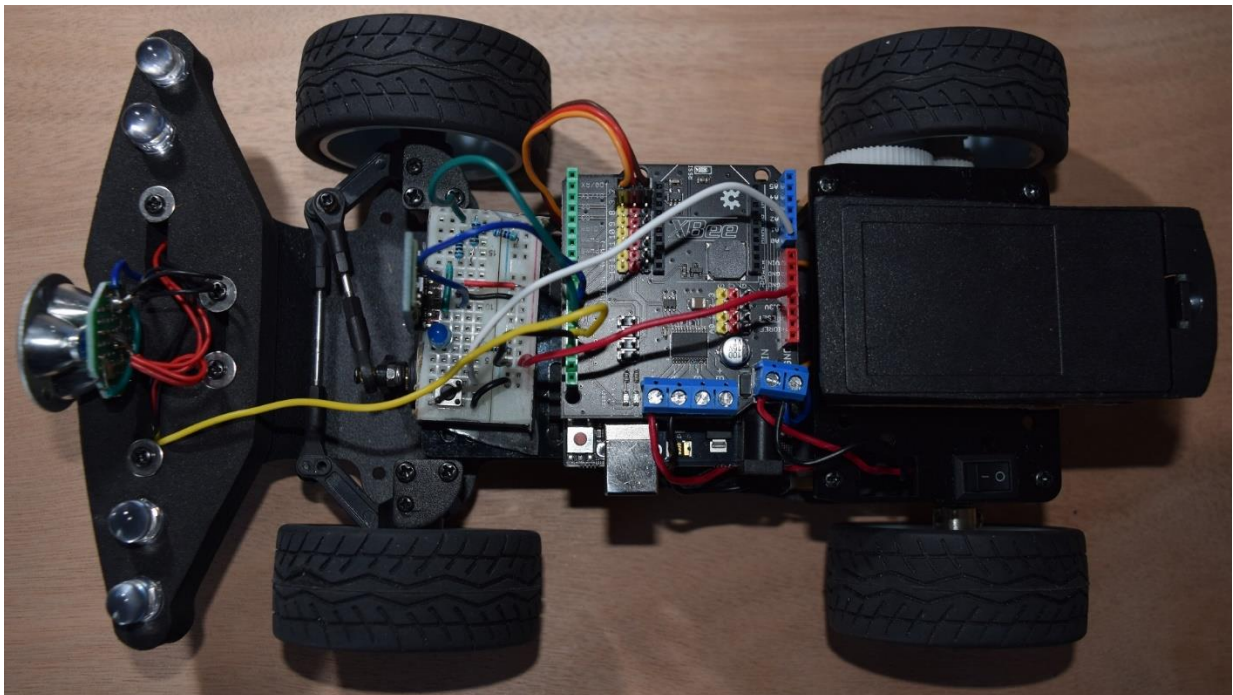
```

Les noms ont été changés pour correspondre aux indications portées en face des broches de la carte YFROBOT: DIR1, DIR2, PWMA

Extraits du nouveau programme B4R adapté à la voiture n° 2

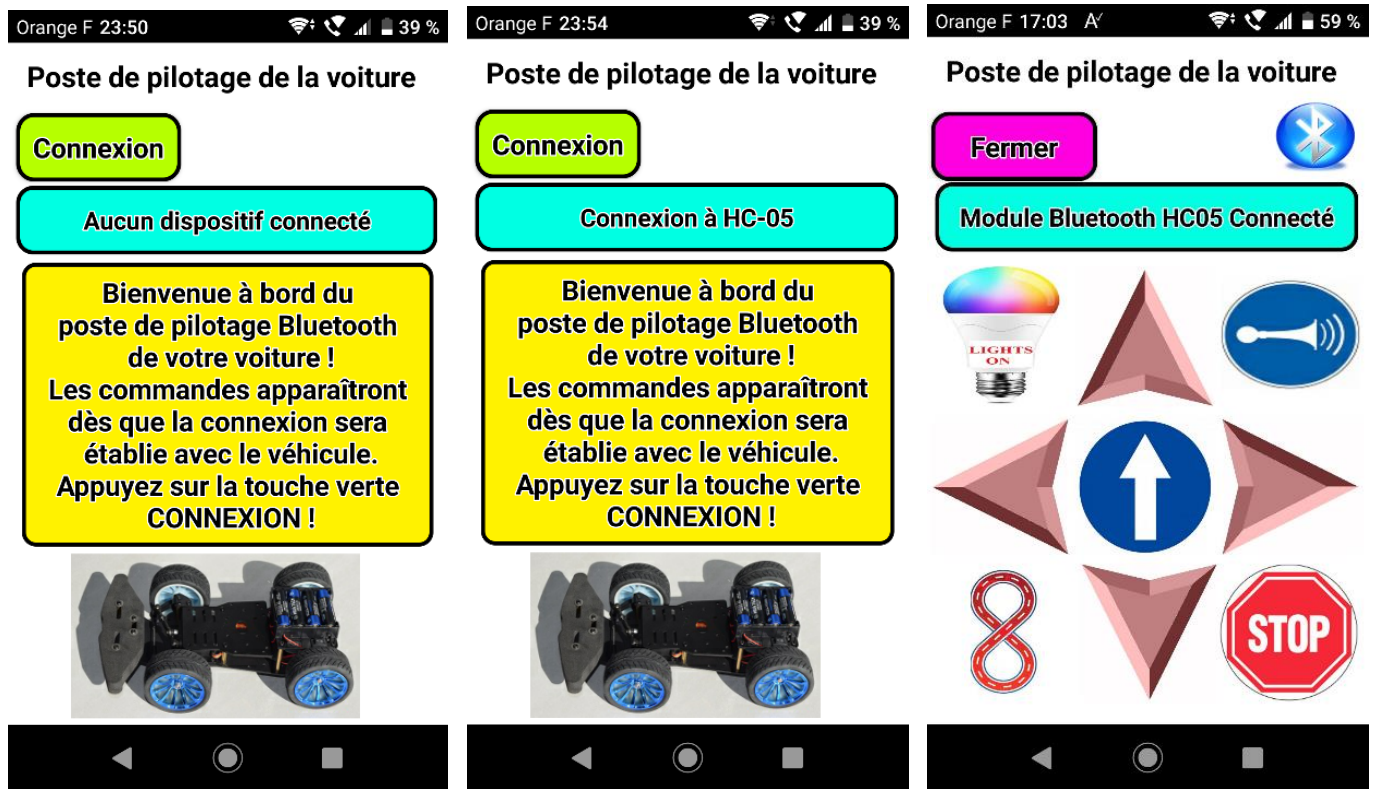
Vidéo : Voir la voiture Arduino en circulation autonome

<https://www.marcalaindaniel.fr/ARDUINO/ArduinoCar/YFROBOTarduinoCar/VideoCircuitAutonomie.mp4>



Application B4A pour Smartphone Android

Notons que cette application est la même, quelle que soit l'installation effectuée (Voiture n° 1 ou voiture n° 2) et elle fonctionnera aussi bien avec le programme B4R associé à l'utilisation d'une carte Arduino MEGA2560 qu'avec le programme B4R associé à l'utilisation de la carte Arduino UNO !



Captures d'écran de l'application B4A "NewPilote" installée sur un Smartphone

L'application a pour but de transmettre des informations depuis l'écran de votre Smartphone jusqu'à la voiture Arduino en utilisant le système Bluetooth qui va envoyer des données depuis votre Smartphone. Ces données seront reçues par l'intermédiaire du petit module Bluetooth HC-05 installé sur la voiture. Le programme B4R installé sur la voiture va interpréter ces données et les traduire par des actions directes sur la conduite et les déplacements de la voiture.

Pour cela, il faut préalablement mettre la voiture sous tension, **une mini-LED rouge se met à clignoter rapidement en bas du module HC-05** ;

Lancez l'application « NewPilote » sur votre Smartphone puis cliquez sur la touche verte « Connexion », le statut de la connexion va passer par les phases suivantes :

- Aucun dispositif connecté
- Recherche dispositif en cours
- HC-05 trouvé, veuillez patienter
- Connexion à HC-05
- Module Bluetooth HC05 Connecté >>> **Allumage de la LED bleue témoin près du module.**
Une fois la connexion faite, la mini-LED rouge clignote plus lentement 2 fois par seconde en alternance avec une période d'extinction.

Si la connexion ne se fait pas, le statut de la connexion sera le suivant :

- Module HC-05 non détecté

Et un message apparaîtra sur l'écran de votre Smartphone, vous invitant à refaire une tentative.



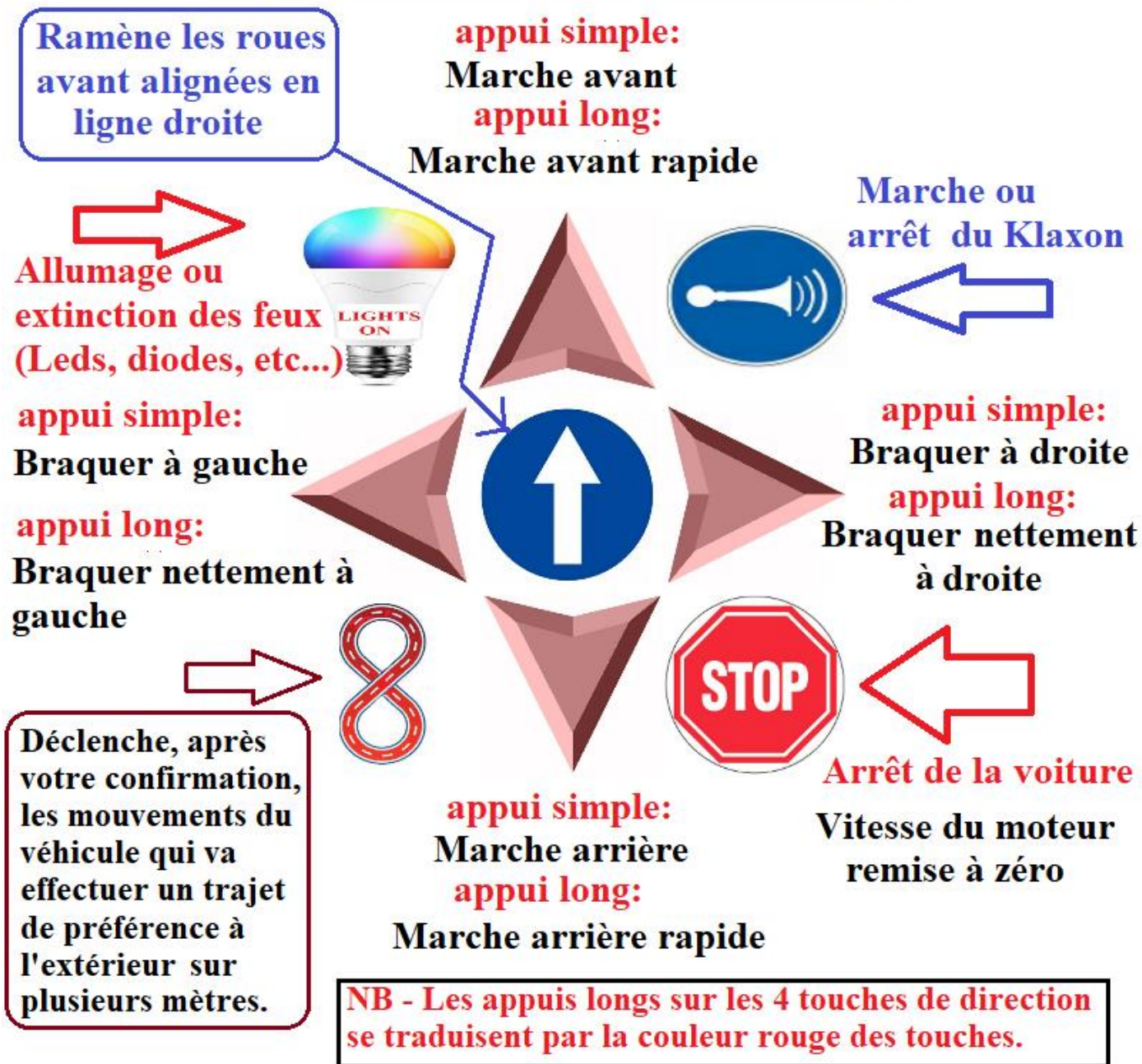
Il vous appartient alors d'éteindre puis de rallumer la voiture, de fermer puis de relancer l'application sur votre Smartphone.

Notons que lors de la première connexion, un code PIN simple (1 2 3 4) vous sera sans doute demandé sur votre Smartphone puis restera en mémoire de votre appareil et ne vous sera plus demandé sauf si vous changez le module Bluetooth HC-05.

Le « poste de pilotage » Bluetooth de votre voiture va vous permettre d'effectuer les opérations suivantes :

- Allumer ou éteindre les feux installés sur la voiture
- Actionner ou éteindre le Klaxon
- Braquer les roues avant légèrement ou carrément à droite
- Braquer les roues avant légèrement ou carrément à gauche
- Rouler en avant à vitesse moyenne ou à grande vitesse
- Reculer à petite vitesse ou plus vite
- Ramener les roues avant en position alignées en ligne droite
- Arrêter la voiture en pressant la touche STOP équivalente au frein
- Donner l'ordre à la voiture d'effectuer seule un circuit de plusieurs mètres de préférence à l'extérieur
- Quitter l'application

**Poste de pilotage de la voiture
sur votre Smartphone via Bluetooth**



Téléchargement de l'application B4A pour appareil Android :

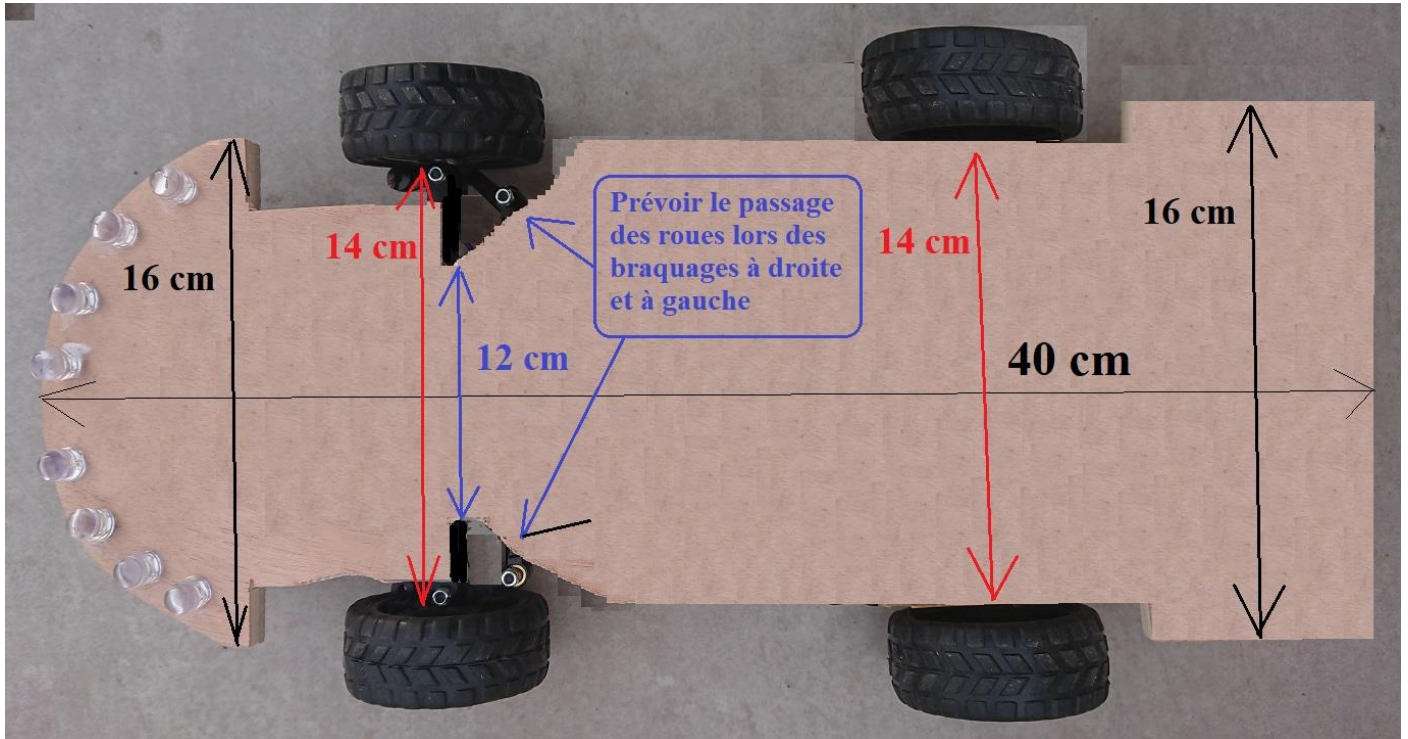
<https://www.marcalaindaniel.fr/ARDUINO/ArduinoCar/Programmes/NewPilote.zip>

Téléchargement du fichier APK directement installable sur votre appareil :

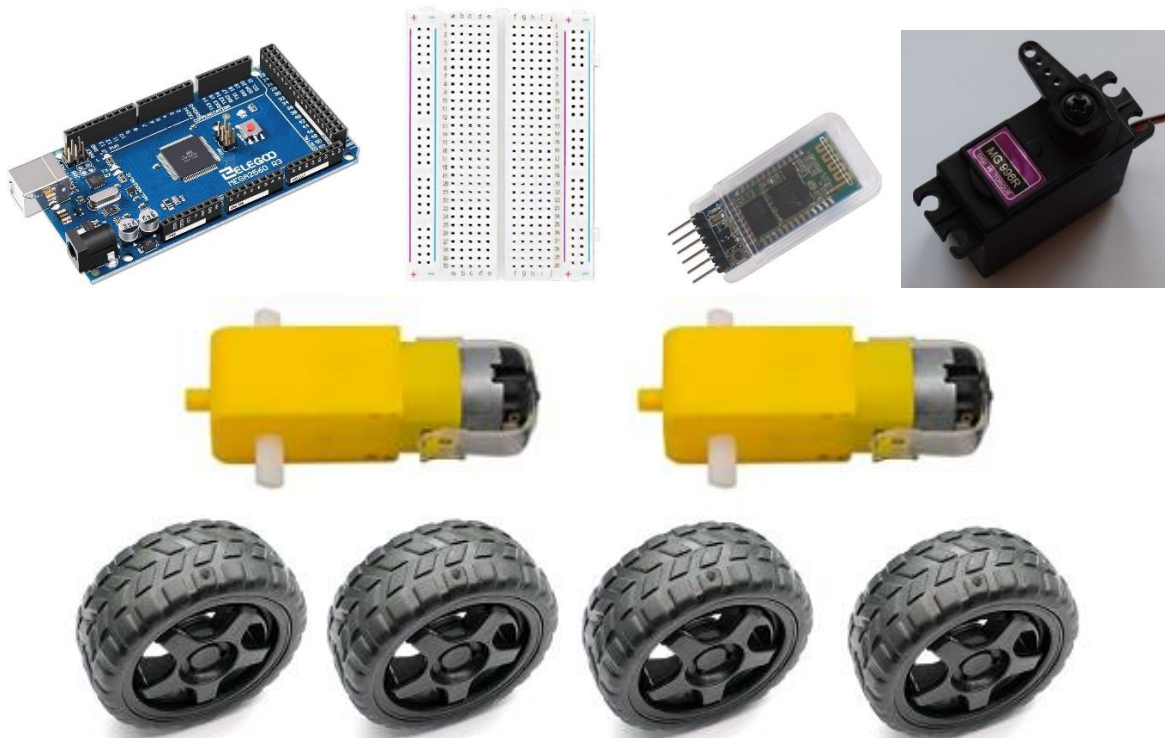
<https://www.marcalaindaniel.fr/ARDUINO/ArduinoCar/Programmes/NewPiloteAPK.zip>

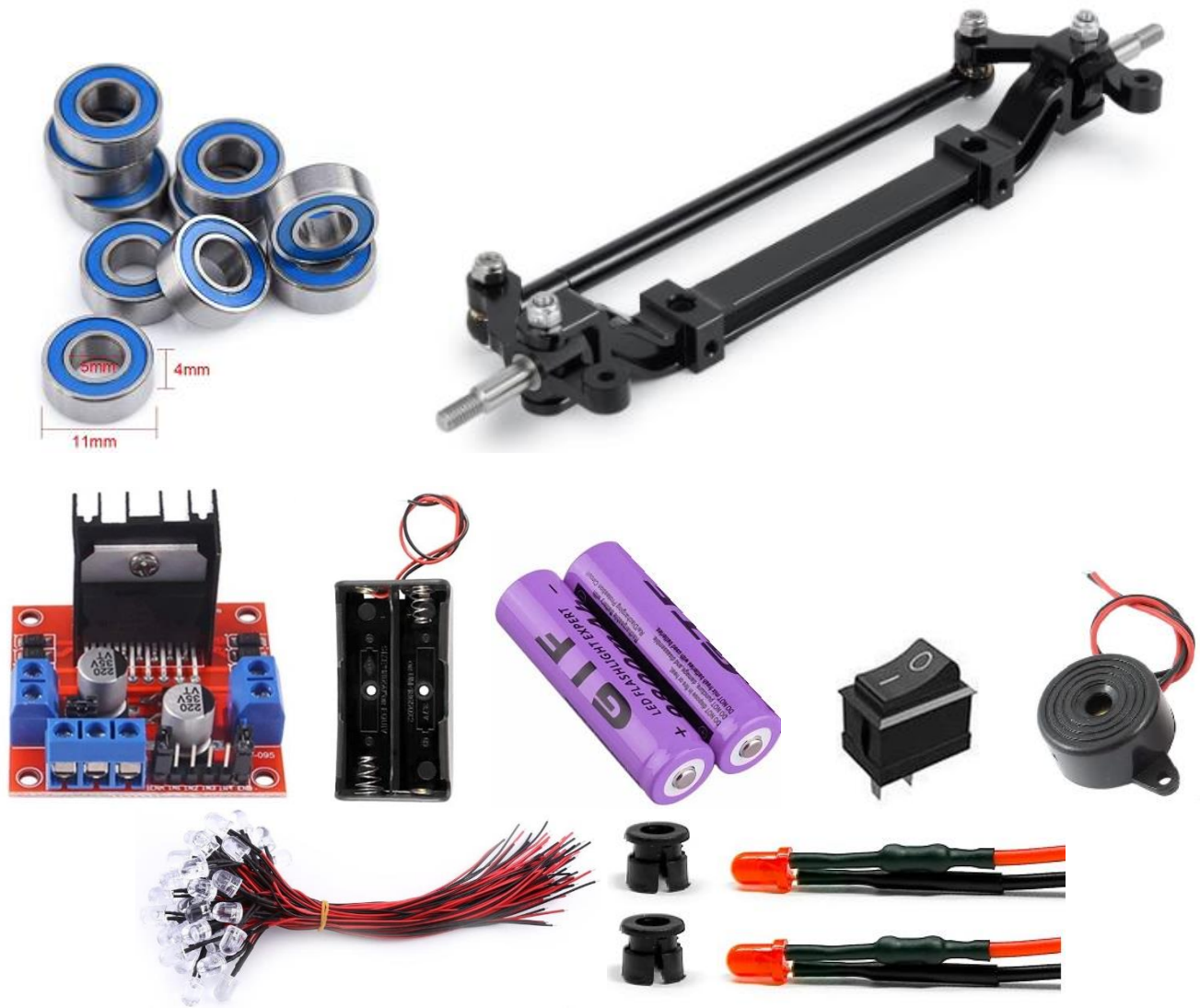
Bricolage artisanal alternatif

Je propose enfin une troisième solution pour les bricoleurs : construire soi-même sa voiture Arduino, sans le kit d'origine.



Le socle sera découpé dans une planche en contreplaqué de 1 cm d'épaisseur





- Une carte Arduino Mega 2560 ou similaire
- Une demi-plaque de connexions
- Un module Bluetooth HC-05
- Un servomoteur MG 996R avec câbles et accessoires (bras et vis) pour contrôler la direction avant (Nous le fixons sous la planche utilisée comme socle en avant de la barre de direction)
- Deux moteurs pour assurer la transmission des roues arrière
- Quatre roues Arduino
(Les 2 roues avant seront inversées, percées et modifiées avec des mini-roulements qui vont s'emboîter sur les deux axes, deux écrous seront nécessaires pour les tenir. Les roues arrière sont emboîtées simplement sur les axes moteurs à section rectangulaire).
- 4 mini-roulements à bille 5 X 11 X 4 mm (2 par roue avant)

https://www.amazon.fr/gp/product/B07CXN1143/ref=ppx_yo_dt_b_asin_image_o04_s00?ie=UTF8&th=1

- Barre de direction complète prééquipée (écartement des roues de 14 cm)

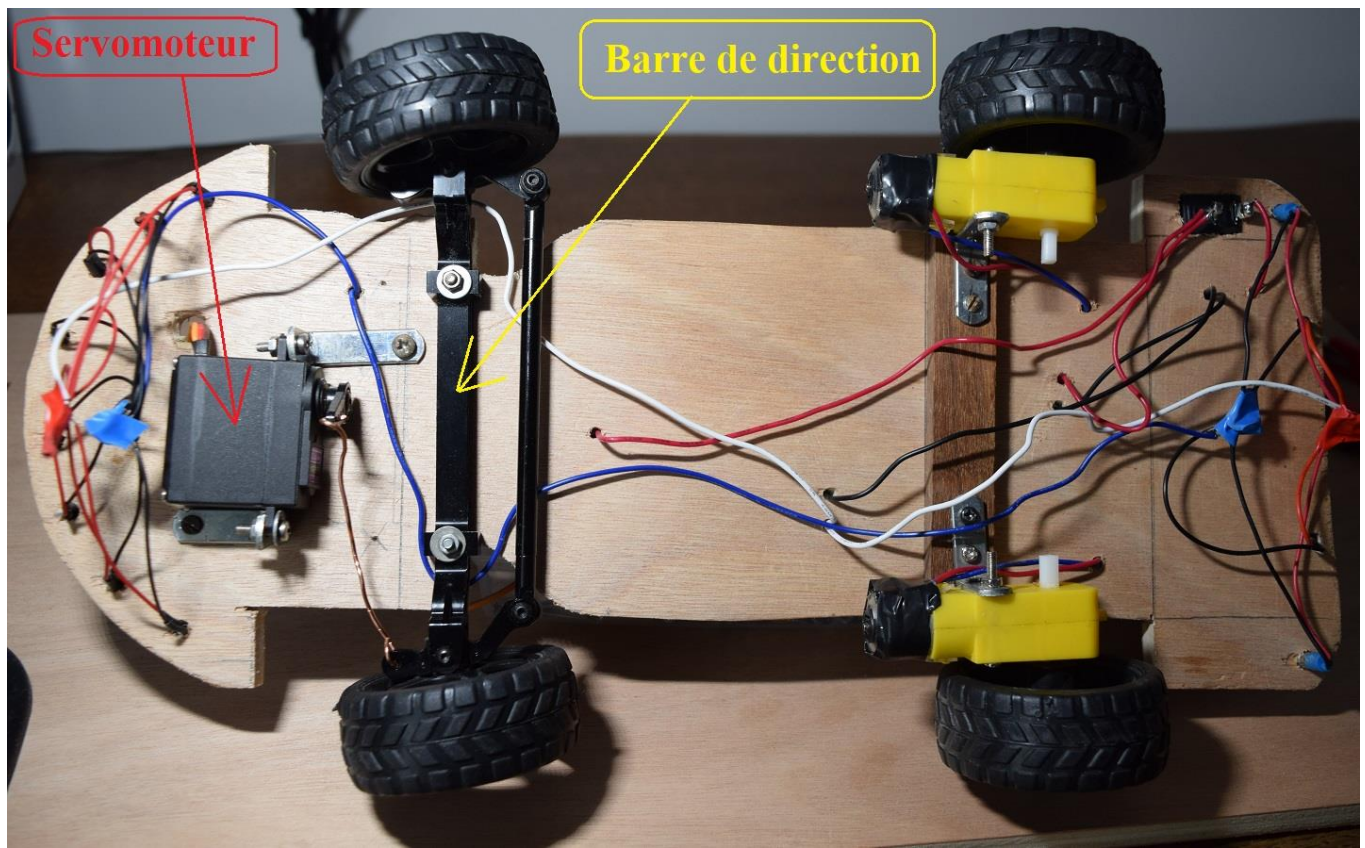
<https://www.aliexpress.com/snapshot/0.html?spm=a2g0s.9042311.0.0.3ffb6c37GPuxtQ&orderId=8140926754236110&productId=4000266374525>

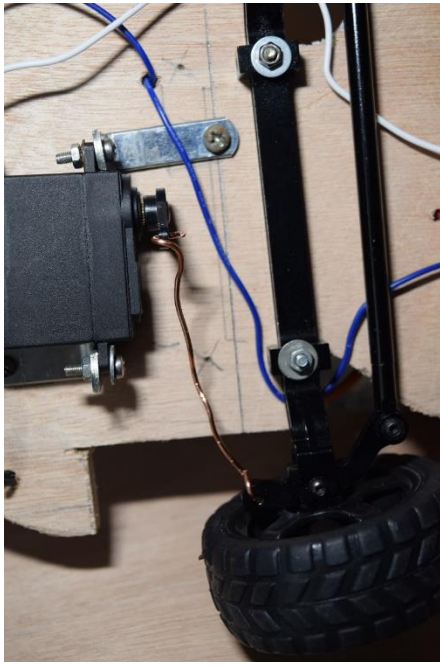
- Un module L298N Dual H-Bridge Motor Driver

- Un support pour 2 batteries au lithium 18650 Li-ion
- Deux batteries rechargeables Li-Ion 18650 de 3,7 volts chacune
- Un interrupteur général
- 1 buzzer qui servira de Klaxon
- 3 résistances de 1K Ω , 1 résistance de 220 Ω , 1 LED bleue, des LEDs blanches pour placer à l'avant, 2 diodes rouges pour placer à l'arrière
- Câbles de couleur, jumpers pour les diverses connexions entre tous ces éléments



Prévoir un écartement de 14 cm entre les roues avant et donc les roues arrière pour découper votre planche-support en conséquence



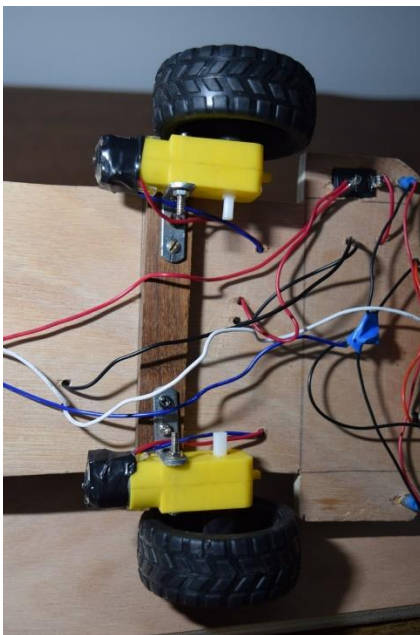


Faute d'avoir reçu des petites biellettes à écartement réglable que j'avais commandées, j'utilise en attendant une tige de cuivre (en provenance d'un câble électrique dénudé) que j'ai fixé d'un côté au « bras » du servomoteur et de l'autre à l'une des extrémités de la barre de direction.

Le servomoteur est fixé solidement sous la planche de socle en avant de la barre de direction. C'est exactement le même modèle que celui qui est utilisé avec les voitures Arduino n° 1 et n° 2.

La barre de direction est solidement fixée au socle par deux vis + écrous.

Servomoteur et barre de direction avant



Les deux roues arrière sont emboîtées sur les axes à section rectangulaire des deux moteurs DC.

Contrairement aux roues avant, ces roues n'ont subi aucune modification.

Il n'y a pas de sens privilégié pour la fixation de ces moteurs. Chacun les installe comme il le veut ou il le peut.

Simplement, si le sens de rotation ne correspond pas à la marche avant et/ou arrière, il suffit d'inverser les connexions + et - du ou des 2 moteurs au niveau des sorties du module L298N.



Les roues avant sont inversées. Elles ont été percées pour pouvoir encastrer 2 petits roulements à billes et leur largeur au niveau de l'axe a été réduite pratiquement de moitié afin de pouvoir les fixer sur l'axe prévu sur la barre de direction. Une petite rondelle et un écrou permettent de les fixer solidement. La largeur doit correspondre à l'épaisseur de 2 roulements soit environ 8 mm.



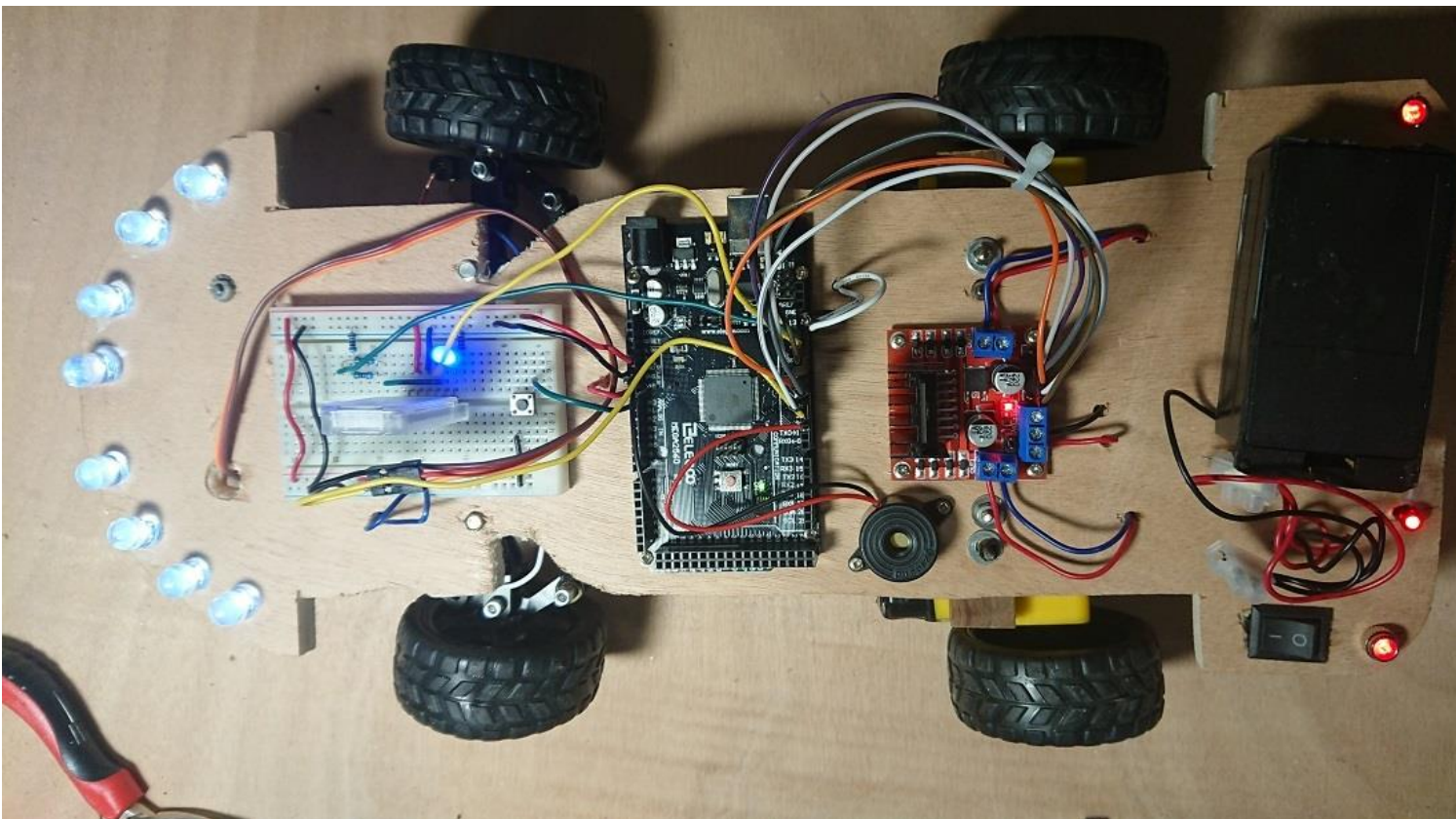
Les deux roues arrières n'ont subi aucune modification et sont simplement emboîtées sur les sections rectangulaires des axes sortant des moteurs arrières. Ce sont des roues Arduino facilement accessibles dans le commerce en ligne.

Lors de la construction du véhicule, il faut trouver la solution pour que la hauteur sol-socle soit la même à l'avant et à l'arrière du véhicule.

Le programme Arduino B4R sera légèrement modifié puisque nous gérons désormais 2 moteurs arrière au lieu d'un seul mais le scénario reste le même et donc l'application B4A sur Smartphone reste exactement la même que précédemment avec la voiture avec châssis équipé.

Le bouton de test provoquera exactement les mêmes résultats :

- Les roues avant vont se braquer vers la droite puis vers la gauche avant de s'aligner en ligne droite
- Le véhicule va avancer pendant 3 secondes puis reculer pendant 3 secondes
- De nouveau, les feux et le Klaxon vont s'allumer puis s'éteindre 3 fois



Prototype issu du bricolage, en état de fonctionner

Par contre le schéma des connexions n'est pas exactement le même puisque l'utilisation de deux moteurs arrière a pour conséquence l'utilisation complète du module L298N et de ses 6 broches ENA, IN1, IN2, IN3, IN4 et ENB

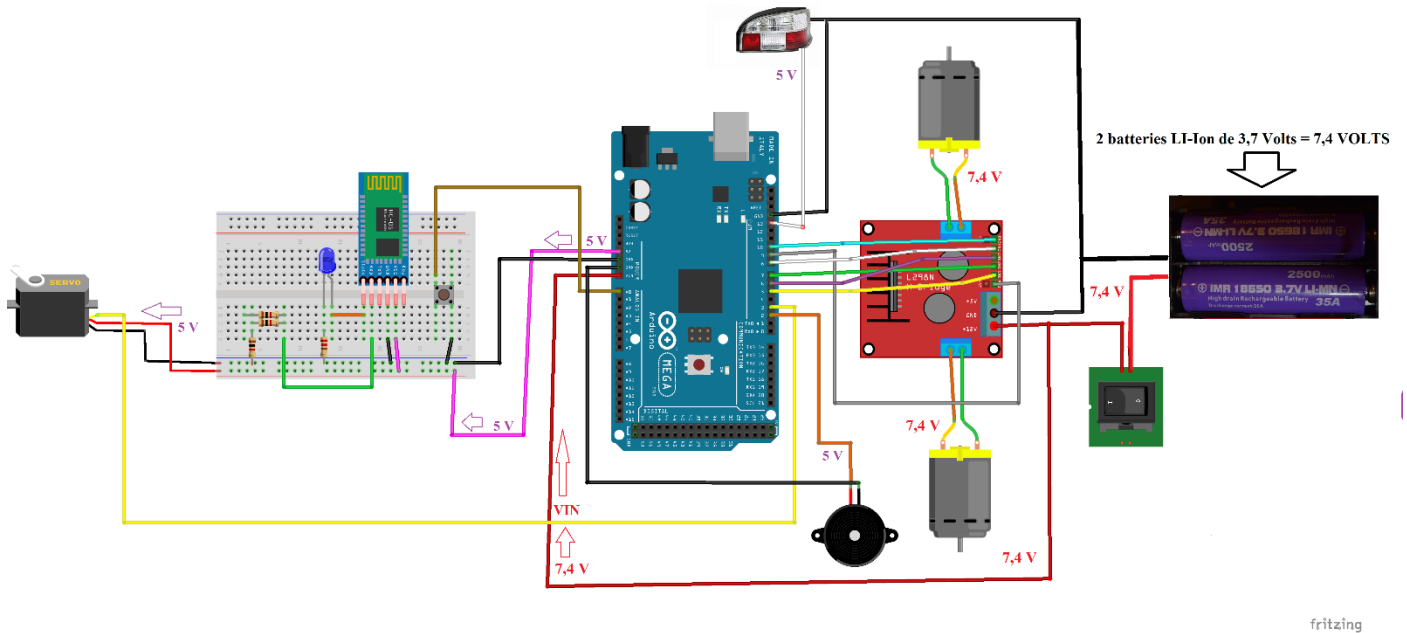
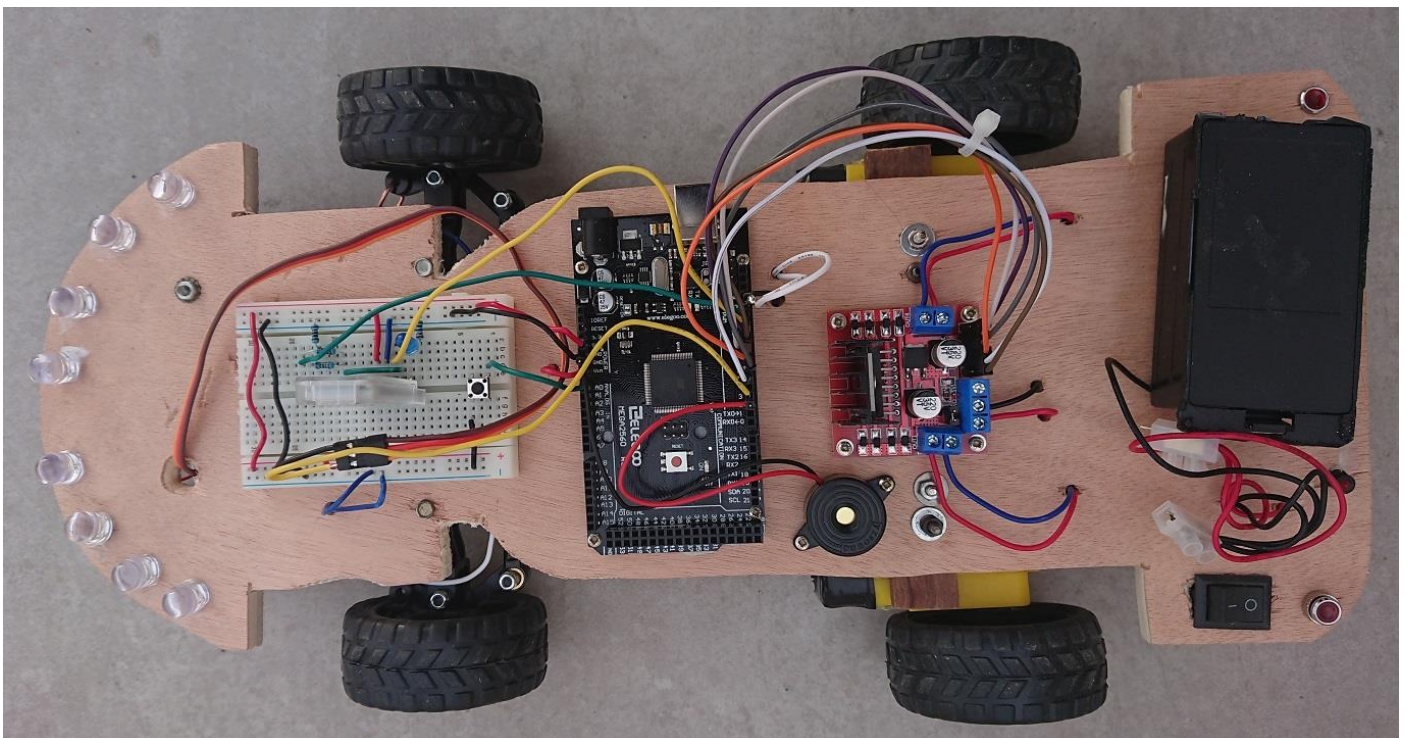


Schéma général des connexions entre les divers composants de la voiture Arduino



Vidéo >>> **TESTS** *obtenus après la pression sur le bouton de tests*

<https://www.marcalaindaniel.fr/ARDUINO/4WDArduinoCar/TESTS.mp4>

Vidéo >>> **Circuit extérieur en autonomie**

<https://www.marcalaindaniel.fr/ARDUINO/4WDArduinoCar/VideoVehicle.mp4>

Le programme B4R adapté à notre bricolage varie légèrement par rapport aux programmes utilisés avec les voitures Arduino n° 1 et n° 2 en raison du fait qu'il y a deux moteurs au lieu d'un et que nous utilisons les 6 broches du module L298N. Ce nouvel algorithme tient compte du fait que nous contrôlons deux roues arrières indépendantes alors que dans les voitures 1 et 2, le moteur arrière entraîne seul les deux roues arrières.

Lien de téléchargement >>> Téléchargement du programme adapté Arduino B4R

<https://www.marcalaindaniel.fr/ARDUINO/4WDArduinoCar/4WDArduino.zip>

Rappelons que l'application B4A sur Smartphone est la même que celle qui est utilisée avec les voitures Arduino n° 1 et n°2 étudiées plus haut :

<https://www.marcalaindaniel.fr/ARDUINO/ArduinoCar/ApplicationB4A.html>

Voir mes autres programmes Arduino ici >>>

<https://www.marcalaindaniel.fr/ARDUINO/Index.html>

*Brochure pédagogique récapitulative rédigée en novembre 2021 par Marc DANIEL
Vos observations >>> marcdaniel1@orange.fr*

[Retour Page 1](#)