



Voiture télécommandée en bluetooth par son smartphone

Créer sa propre voiture télécommandée, son appli sous App Inventor et insérer son programme dans Arduino pour la piloter

 Difficulté **Moyen**

 Durée **3 heure(s)**

 Catégories **Électronique, Jeux & Loisirs, Robotique**

 Coût **60 EUR (€)**

Sommaire

Étape 1 - Création de l'appli sous App Inventor 2

Étape 2 - Code Arduino

Étape 3 - Châssis design et câblage

Commentaires

Matériaux

Contreplaqué, 1 carte Arduino, 2 moteurs 5V cc (modélisme), 1 driver pour moteur à courant continu, une alim de 9V, un bluetooth HC05 ou HC06, 2 roues, une roue libre, cables arduino femelles, pins, une mini-breadboard

Outils

Découpe Laser
Ordinateur avec Arduino
App Inventor 2 en ligne
Fer à souder
Tournevis
Perceuse

Étape 1 - Création de l'appli sous App Inventor 2

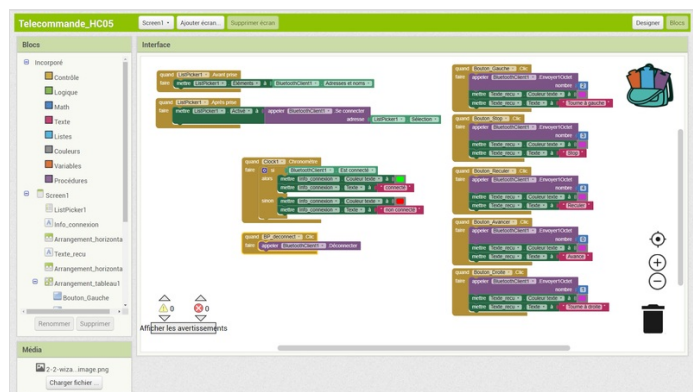
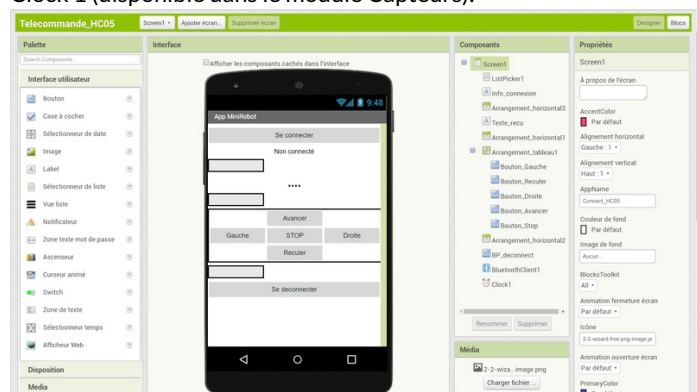
On va se créer notre propre appli qu'on installera directement sur notre téléphone. On va utiliser App Inventor 2 qui nécessite un compte gmail pour se connecter et un smarthone avec Androide pour fonctionner. (A noter l'equivalent gratuit Thinkable pour Androide et IOS, ne necessitant pas gmail).

Le programme est assez simple, on liste tous les appareils Bluetooth disponibles, on sélectionne celui dédié à notre voiture pour ne pas interférer avec celui du voisin (nécessité de renommer le HC05 auparavant via les commandes AT), puis on envoie un chiffre correspondant à notre commande (1 pour avancer, 2 pour reculer par ex.)

Ensuite on installe directement le build .apk sur notre téléphone.

Info complémentaire : le 1er bouton de notre interface "Se connecter" est un composant Listpicker qui permet de détecter les réseaux bluetooth disponibles et d'en sélectionner un.

Et les 2 éléments invisibles nécessaires au fonctionnement de l'appli sont le client Bluetooth (disponible dans le module Connectivité) et Clock 1 (disponible dans le module Capteurs).



Étape 2 - Code Arduino

Le code Arduino que l'on va charger sur la carte, va faire le lien entre ce que recevra le petit module bluetooth HC05 et les commandes à envoyer aux moteurs de droite et de gauche.

Un petit driver TB6612FNG (GOTRONIC) controle les moteurs (2 micro-moteur DC 6V 160 RPM).

On inclut la bibliothèque SoftwareSerial.h pour communiquer avec le bluetooth. Attention à bien croiser le RX et TX du bluetooth et de l'Arduino lors du câblage.

Et un simple switch case va définir les différentes configurations selon la commande reçue.

Le sens de rotation du moteur et donc de la roue sera commandée par un signal numérique (DigitalWrite) en HIGH ou LOW.

La vitesse sera modulée par un signal analogique (AnalogWrite) en PWM comprise entre 0 et 255.

```
1 #include <SoftwareSerial.h>
2
3 int RXPin = 5;
4 int DIR1 = 4;
5 int PWM2 = 6;
6 int DIR2 = 7;
7 int Vaz;
8
9 #define rxPin 10 // Broche 11 en tant que RX, à raccorder sur TX du HC-05
10 #define txPin 11 // Broche 10 en tant que TX, à raccorder sur RX du HC-05
11 SoftwareSerial hc05(rxPin, txPin);
12
13 void setup()
14 {
15   pinMode(DIR1, OUTPUT);
16   pinMode(DIR2, OUTPUT);
17
18   // definition des pin Modes pour TX, RX pins de l ARDUINO:
19   pinMode(rxPin, INPUT);
20   pinMode(txPin, OUTPUT);
21
22   // Ouverture et definition vitesses ports serie de communication
23   hc05.begin(9600);
24   Serial.begin(9600);
25   Serial.println("Bonjour");
26   delay(300);
27 }
```

```
28 void loop()
29 {
30
31   Serial.println("En attente d'instructions!");
32   delay(30);
33   while (hc05.available() > 0){
34     var = hc05.read();
35     Serial.println(var);
36     delay(30);
37
38     switch (var) {
39
40       case 0:
41         //AVANCE
42         digitalWrite(DIR1,LOW);
43         digitalWrite(DIR2,LOW);
44         analogWrite(PWM1, 200);
45         analogWrite(PWM2, 200);
46         delay(30);
47         break;
48
49       case 1:
50         //DROITE
51         analogWrite(PWM1, 0);
52
53
54     }
```

```
55   delay(30);
56   break;
57
58   case 2:
59     //GAUCHE
60     analogWrite(PWM1, 200);
61     analogWrite(PWM2, 0);
62     delay(30);
63     break;
64
65
66   case 3:
67     //RECULE
68     digitalWrite(DIR1,HIGH);
69     digitalWrite(DIR2,HIGH);
70     analogWrite(PWM1, 200);
71     analogWrite(PWM2, 200);
72     delay(30);
73     break;
74
75   case 4:
76     //STOP
77     analogWrite(PWM1, 0);
78     analogWrite(PWM2, 0);
79     delay(30);
80     break;
81   }
82 }
```

Étape 3 - Châssis design et câblage

La plupart de nos designs sont des prototypes. Dessinés sous Inkscape pour être découpés à la laser principalement, ils devaient pouvoir recevoir les différents éléments pour être facilement câblés par la suite. Pensez à mettre les connecteurs d'alimentation orientés vers l'extérieur pour ne pas être gêné par la suite. Quelques soudures sur les fils moteurs et les pins du driver pour les maintenir. Par souci de gain de place, l'alimentation a été fixée sous les châssis. Une roue libre fixée à l'avant permet un contrôle droite ou gauche en bloquant un moteur du côté opposé.

Suivre les schémas de câblage, en accord avec votre code Arduino pour les numéros de pins et voilà !

A noter que le 3.3V en sortie de l'arduino est suffisant pour alimenter notre HC05.

