

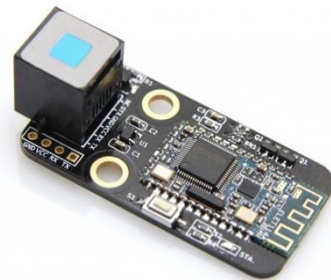


Pilotage mini robot à partir d'Android

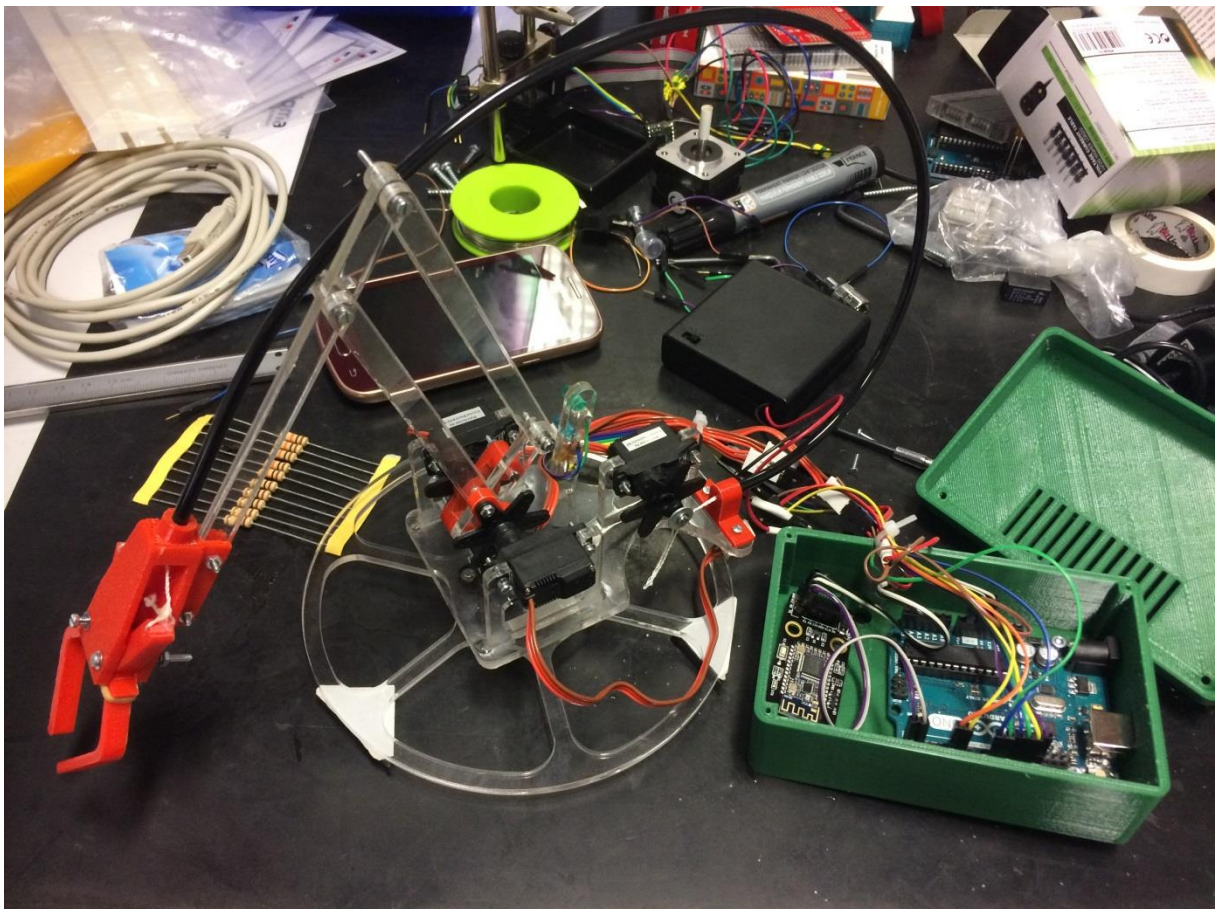
1. Câblage :

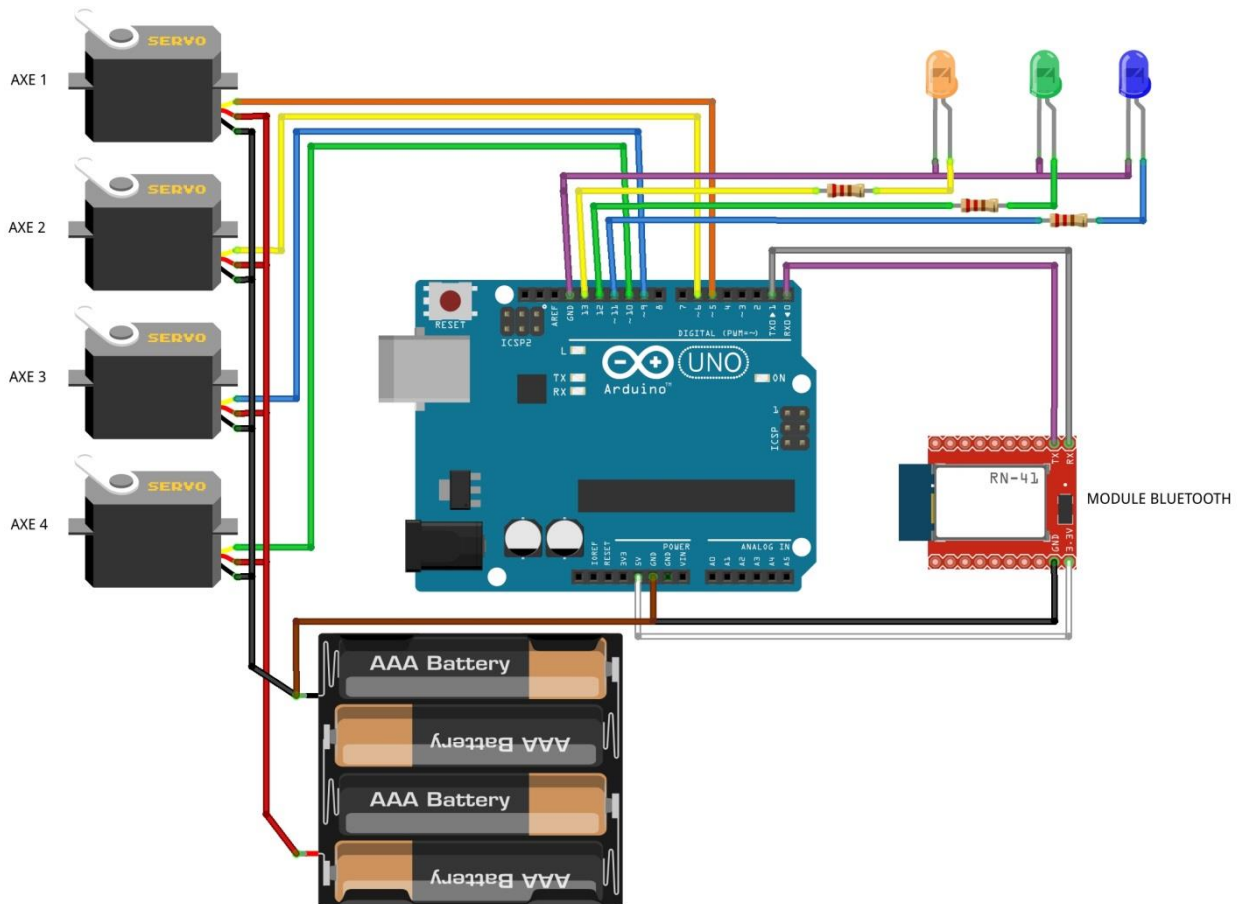
J'ai utilisé le module Bluetooth de Makebot, mais n'importe lequel peut faire l'affaire. Il n'y a, pour gérer la partie communication que 4 branchements à faire.

Pin module Bluetooth	Description	Branchement Pin Arduino
TX	Chip Select / Data3	0 (RX)
RX	MOSI / Command	1 (TX)
GND	Power Supply Ground	GND
VCC	Power Supply (5V)	5V



Le mini robot en lui-même est un petit robot 3 axes avec une pince de saisie (pilotés par 4 servo moteurs).





fritzing

2. Code :

Le choix du protocole de communication, est l'envoi d'un caractère alphanumérique pour piloter chaque mouvement ou action du robot.

Lettre	Action	Lettre	Action	Lettre	Action
w	LED bleue allumée	l	Servo1 « - »	u	Servo3 « + »
x	LED bleue éteinte	r	Servo1 « + »	d	Servo3 « - »
t	LED verte allumée	f	Servo2 « + »	o	Servo4 « + »
v	LED verte éteinte	b	Servo2 « - »	c	Servo4 « - »
y	LED jaune allumée				
z	LED jaune éteinte				

```
#include <Servo.h>
char incomingByte = 0;
int LEDBlue = 11;
int LEDGreen = 12;
int LEDYellow = 13;
int Servo1Pos = 90;
int Servo2Pos = 90;
int Servo3Pos = 90;
int Servo4Pos = 90;
int ServoStep = 2;
Servo Servo1;
Servo Servo2;
Servo Servo3;
```

```

Servo Servo4;

void setup() {
  Serial.begin(115200);
  setupBlueToothConnection();
  pinMode(LEDBlue, OUTPUT);
  pinMode(LEDGreen, OUTPUT);
  pinMode(LEDYellow, OUTPUT);
  Servo1.attach(5);
  Servo2.attach(6);
  Servo3.attach(9);
  Servo4.attach(10);
}

void loop() {
  if (Serial.available() > 0) {
    // read the incoming byte:
    incomingByte = Serial.read();
    if (incomingByte == 'w') {
      digitalWrite(LEDBlue, HIGH);
    }
    if (incomingByte == 'x') {
      digitalWrite(LEDBlue, LOW);
    }
    if (incomingByte == 't') {
      digitalWrite(LEDGreen, HIGH);
    }
    if (incomingByte == 'v') {
      digitalWrite(LEDGreen, LOW);
    }
    if (incomingByte == 'y') {
      digitalWrite(LEDYellow, HIGH);
    }
    if (incomingByte == 'z') {
      digitalWrite(LEDYellow, LOW);
    }
    //
    // Servo 1 pilot with left (l) and right (r)
    if (incomingByte == 'l') {
      Servo1Pos = Servo1Pos - ServoStep;
    }
    if (incomingByte == 'r') {
      Servo1Pos = Servo1Pos + ServoStep;
    }
    Servo1.write(Servo1Pos);
    //
    // Servo 2 pilot with forward (f) and backward (b)
    if (incomingByte == 'f') {
      Servo2Pos = Servo2Pos + ServoStep;
    }
    if (incomingByte == 'b') {
      Servo2Pos = Servo2Pos - ServoStep;
    }
    Servo2.write(Servo2Pos);
    //
    // Servo 3 pilot with up (u) and down (d)
    if (incomingByte == 'u') {
      Servo3Pos = Servo3Pos + ServoStep;
    }
    if (incomingByte == 'd') {
      Servo3Pos = Servo3Pos - ServoStep;
    }
  }
}

```

```

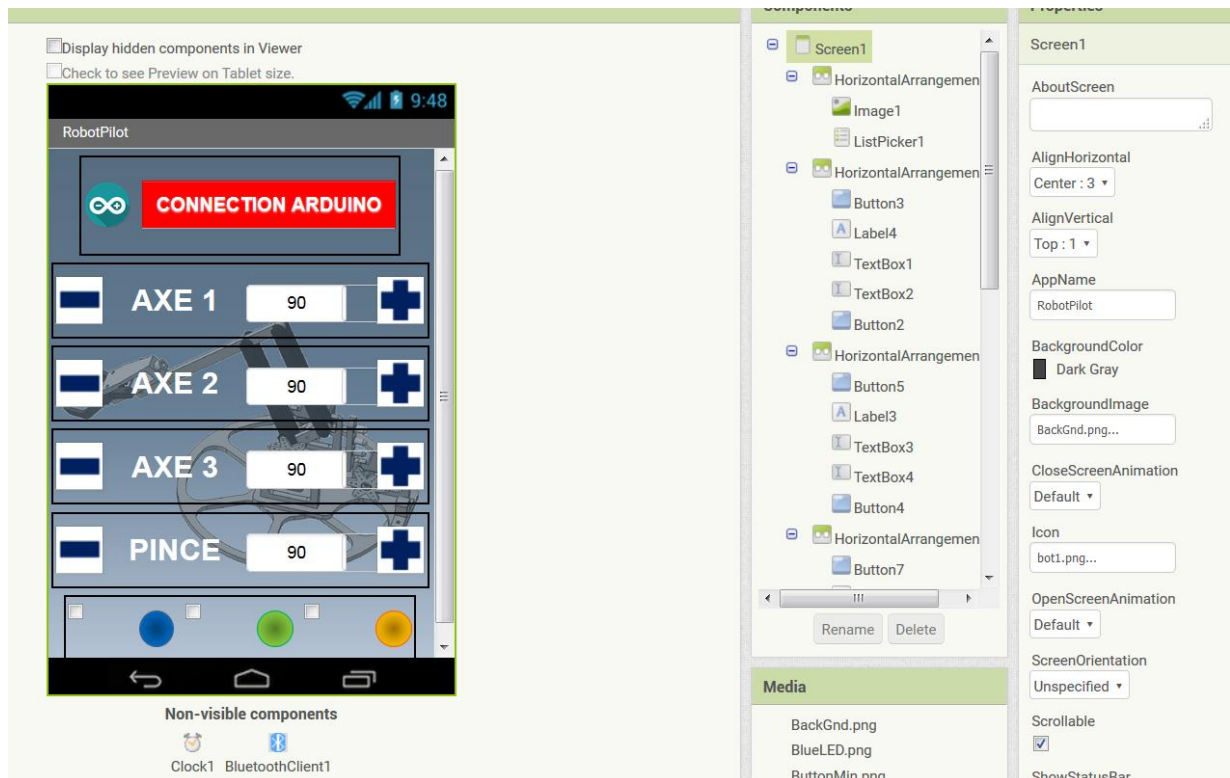
    }
    Servo3.write(Servo3Pos);
    //
    // Servo 4 pilot with open (o) and close (c)
    if (incomingByte == 'o') {
        Servo4Pos = Servo4Pos + ServoStep;
    }
    if (incomingByte == 'c') {
        Servo4Pos = Servo4Pos - ServoStep;
    }
    Servo4.write(Servo4Pos);
    //
    // Homing all servos positions
    if (incomingByte == 'h') {
        Servo1Pos = 90;
        Servo2Pos = 90;
        Servo3Pos = 90;
        Servo4Pos = 90;
    }
}
}

void setupBlueToothConnection() // fonction de configuration du module
bluetooth
{
    Serial.begin(115200); //vitesse de bluetooth
    Serial.print("\r\n+STBD=115200\r\n"); // fixe la vitesse du bluetooth
    Serial.print("\r\n+STWMOD=0\r\n"); //bluetooth en mode esclave
    Serial.print("\r\n+STNA=Arduino"); //nom de l'appareil
    Serial.print("\r\n+STPIN=0000\r\n");//code pin "0000"
    Serial.print("\r\n+STOAUT=1\r\n"); // Permit Paired device to connect me
    Serial.print("\r\n+STAUTO=0\r\n"); // Auto-connection should be forbidden
here
    delay(2000); // This delay is required.
    Serial.print("\r\n+INQ=1\r\n"); //make the slave bluetooth inquirable
    delay(2000); // This delay is required.
    Serial.flush();
}
}

```

3. Programme App Inventor

Voici le Screen design choisi :



On a donc au total un ListPicker (CONNECTION ARDUINO), 4 boutons « + » et 4 boutons « - » pour les 4 axes pilotés, 4 TextBox pour l'affichage de positions des axes, 3 CheckBox pour l'allumage des LEDs et puis quelques images et un peu de Layout pour rendre l'ensemble visuellement un peu plus sympa. En plus en invisible, on ajoute une Clock et un BluetoothClient pour la partie communication. Il faut faire attention à forcer les dimensions des éléments entre 40x40 et 60x60 pixels, sinon il faut ensuite avoir des tout petits doigts pour utiliser l'appli sur le smartphone...

Pour la programmation, on a ajouté une option appui long qui correspond à un mouvement de 5 steps contre un mouvement de 1 step pour un appui court sur chacun des boutons -/+ des axes. La valeur du step tel que définie dans le programme Arduino correspond à un pas angulaire de 2°.

E

t voilà le bloc diagram correspondant au pilotage au calcul des angles et à l'affichage :

```

initialize global Servo1 to 90
initialize global Servo2 to 90
initialize global Servo3 to 90
initialize global Servo4 to 90

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do evaluate but ignore result call BluetoothClient1 .Connect
address ListPicker1 . Selection

when Button3 .Click
do call BluetoothClient1 .SendText
text "l"
set global Servo1 to get global Servo1 - 2
set TextBox1 . Text to get global Servo1

when Button3 .LongClick
do call BluetoothClient1 .SendText
text "lllll"
set global Servo1 to get global Servo1 - 10
set TextBox1 . Text to get global Servo1

when Button2 .Click
do call BluetoothClient1 .SendText
text "r"
set global Servo1 to get global Servo1 + 2
set TextBox1 . Text to get global Servo1

when Button2 .LongClick
do call BluetoothClient1 .SendText
text "rrrrr"
set global Servo1 to get global Servo1 + 10
set TextBox1 . Text to get global Servo1

```

```

when Button5 .Click
do call BluetoothClient1 .SendText
text "b"
set global Servo2 to get global Servo2 - 2
set TextBox3 . Text to get global Servo2

when Button5 .LongClick
do call BluetoothClient1 .SendText
text "bbbb"
set global Servo2 to get global Servo2 - 10
set TextBox3 . Text to get global Servo2

when Button4 .Click
do call BluetoothClient1 .SendText
text "f"
set global Servo2 to get global Servo2 + 2
set TextBox3 . Text to get global Servo2

when Button4 .LongClick
do call BluetoothClient1 .SendText
text "ffff"
set global Servo2 to get global Servo2 + 10
set TextBox3 . Text to get global Servo2

when Button7 .Click
do call BluetoothClient1 .SendText
text "d"
set global Servo3 to get global Servo3 - 2
set TextBox5 . Text to get global Servo3

when Button7 .LongClick
do call BluetoothClient1 .SendText
text "dddd"
set global Servo3 to get global Servo3 - 10
set TextBox5 . Text to get global Servo3

```

```

when Button6 .Click
do
  call BluetoothClient1 .SendText
  text "u"
  set global Servo3 to get global Servo3 + 2
  set TextBox5 .Text to get global Servo3

```

```

when Button6 .LongClick
do
  call BluetoothClient1 .SendText
  text "uuuuu"
  set global Servo3 to get global Servo3 + 10
  set TextBox5 .Text to get global Servo3

```

```

when Button9 .Click
do
  call BluetoothClient1 .SendText
  text "c"
  set global Servo4 to get global Servo4 - 2
  set TextBox7 .Text to get global Servo4

```

```

when Button9 .LongClick
do
  call BluetoothClient1 .SendText
  text "ccccc"
  set global Servo4 to get global Servo4 - 10
  set TextBox7 .Text to get global Servo4

```

```

when Button8 .Click
do
  call BluetoothClient1 .SendText
  text "o"
  set global Servo4 to get global Servo4 + 2
  set TextBox7 .Text to get global Servo4

```

```

when Button8 .LongClick
do
  call BluetoothClient1 .SendText
  text "ooooo"
  set global Servo4 to get global Servo4 + 10
  set TextBox7 .Text to get global Servo4

```

```

when CheckBox1 .Changed
do
  evaluate but ignore result
  if CheckBox1 .Checked
  then
    do call BluetoothClient1 .SendText
    text "w"
    result true
  else
    do call BluetoothClient1 .SendText
    text "x"
    result false

```



```

when CheckBox3 .Changed
do
  evaluate but ignore result
  if CheckBox3 .Checked
  then
    do call BluetoothClient1 .SendText
    text "y"
    result true
  else
    do call BluetoothClient1 .SendText
    text "z"
    result false

```

```

when CheckBox2 .Changed
do
  evaluate but ignore result
  if CheckBox2 .Checked
  then
    do call BluetoothClient1 .SendText
    text " "
    result true
  else
    do call BluetoothClient1 .SendText
    text " "
    result false

```

