

Notes à propos du robot Elegoo

Bolide Arduino

- [Site projets Arduino](#)
- [Formation en ligne](#)
- Logiciel <https://bitbloq.bq.com/>

Attention : pour que Bitbloq fonctionne on ne doit pas connecter le module bluetooth. Il est impossible de téléverser avec arduino lorsque le module bluetooth est en place. Il faut retirer le module bluetooth avant de téléverser et par la suite, on peut le remettre si on veut l'utiliser avec un iPad.

Test du logiciel <https://bitbloq.bq.com> dans un réseau hors CSBE et ça fonctionne le 8 février PM.

- Installer cette application
<https://chrome.google.com/webstore/detail/bitbloq/ddpdpmbfdhifigignfdiggfbcmfbfj>

Vidéo rapide (ne pas me juger SVP ;-))

https://drive.google.com/open?id=1qflkCLUWDd_yLRqtssOTilu_Bjo8hUsS

Exemple de code pour faire avancer 2s, arrêter 2s, et recommencer : <https://goo.gl/HyVsHH>

Tableau des moteurs

Pin 5 (moteur gauche) et pin 6 (moteur droit) doivent être à HIGH tout le temps (ça active les moteurs, vitesse maximale).

Pin7	pin8	pin9	pin11	Tâche
Low	Low	Low	Low	Arrêt
High	Low	Low	High	Avance
Low	High	High	Low	Recule
Low	High	Low	High	Tourne gauche
High	Low	High	Low	Tourne droite

Varié la vitesse des moteurs : pin 5 (ENA), pin 6 (ENB) à faire varier de 0 à 255 (analogwrite), voir exemple de code <https://goo.gl/YpzJiV>

Radar

Exemple <http://recitmst.qc.ca/arduino/radar-distance/>

Code pris dans ce fichier : <https://drive.google.com/open?id=1UPuqjtZJUHY005KQgCiYDI2K44DyRdkN>

<pre>//Ultrasonic distance measurement Sub function //trig = A5, echo=A4 int Distance_test() { digitalWrite(Trig, LOW); delayMicroseconds(2); digitalWrite(Trig, HIGH); delayMicroseconds(20); digitalWrite(Trig, LOW); float Fdistance = pulseIn(Echo, HIGH); Fdistance= Fdistance / 58; return (int)Fdistance; }</pre>	c'est une fonction qui mesure la distance (en cm) devant le radar. sortie = trig = A5 entrée = echo = A4 on met à 0 le trig on attend on met à 1 le trig 20 microsecondes on met à 0 le trig on place la durée de l'aller retour de l'onde dans Fdistance on calcule la distance en cm on renvoie la distance
---	---

Exemple de code Bitbloq : <https://goo.gl/aMgww3>

Suiveur de ligne

<pre>//www.elegoo.com //Line Tracking IO define #define LT_R !digitalRead(10) #define LT_M !digitalRead(4) #define LT_L !digitalRead(2) #define ENA 5 #define ENB 6 #define IN1 7 #define IN2 8 #define IN3 9 #define IN4 11 #define carSpeed 150 void forward(){ analogWrite(ENA, carSpeed); analogWrite(ENB, carSpeed); digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW); digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH); Serial.println("go forward!"); }</pre>	capteur de droite capteur du milieu capteur de gauche pour les moteurs vitesse des moteurs (0 à 255) pour avance
--	--

<pre>void back(){ analogWrite(ENA, carSpeed); analogWrite(ENB, carSpeed); digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH); digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW); Serial.println("go back!"); }</pre>	pour reculer
<pre>void left(){ analogWrite(ENA, carSpeed); analogWrite(ENB, carSpeed); digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH); digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH); Serial.println("go left!"); }</pre>	tourner à gauche
<pre>void right(){ analogWrite(ENA, carSpeed); analogWrite(ENB, carSpeed); digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW); digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW); Serial.println("go right!"); }</pre>	tourner à droite
<pre>void stop(){ digitalWrite(ENA, LOW); digitalWrite(ENB, LOW); Serial.println("Stop!"); }</pre>	arrêt
<pre>void setup(){ Serial.begin(9600); pinMode(LT_R,INPUT); pinMode(LT_M,INPUT); pinMode(LT_L,INPUT); }</pre>	communication par le moniteur série les entrées
<pre>void loop() { if(LT_M){ forward(); } else if(LT_R) { right(); } }</pre>	si le capteur du milieu voit noir, on avance si le capteur de droite voit noir, on tourne à droite

<pre> while(LT_R); } else if(LT_L) { left(); while(LT_L); } } </pre>	tant que le capteur de droite voit noir si le capteur de gauche voit noir, on tourne à gauche tant que le capteur de gauche voit noir
--	---

Contrôler la vitesse

<pre> //www.elegoo.com #define ENA 5 #define ENB 6 #define IN1 7 #define IN2 8 #define IN3 9 #define IN4 11 void setup() { pinMode(IN1,OUTPUT); pinMode(IN2,OUTPUT); pinMode(IN3,OUTPUT); pinMode(IN4,OUTPUT); pinMode(ENA,OUTPUT); pinMode(ENB,OUTPUT); } void loop() { //go forward digitalWrite(IN1,HIGH); digitalWrite(IN2,LOW); digitalWrite(IN3,LOW); digitalWrite(IN4,HIGH); //reduce the speed for(int i = 255; i >= 0; i--){ analogWrite(ENB,i); analogWrite(ENA,i); delay(20); } } </pre>	les moteurs les sorties avancer à chaque passage dans la boucle, on enlève 1 à la force de départ (255), on attend 20 millisecondes avant de sortir de la boucle et de recommencer.
---	---

<pre>//stop analogWrite(ENB,0); //speed = 0 analogWrite(ENA,0); delay(1000); //runing back digitalWrite(IN1,LOW); digitalWrite(IN2,HIGH); digitalWrite(IN3,HIGH); digitalWrite(IN4,LOW); //accelerate for(int i = 0; i <= 255; i++){ analogWrite(ENB,i); analogWrite(ENA,i); delay(20); } //stop digitalWrite(ENB,LOW); //Motor is off digitalWrite(ENA,LOW); delay(2000); }</pre>	arrêt pendant 1 s on recule en accélérant à chaque passage dans la boucle on ajoute 1 à 0 (force de départ) arrêt 2s
---	--

Éviter des obstacles (radar)

<pre>//www.elegoo.com #include <Servo.h> //servo library Servo myservo; // create servo object to control servo int Echo = A4; int Trig = A5; #define ENA 5 #define ENB 6 #define IN1 7 #define IN2 8 #define IN3 9 #define IN4 11 #define carSpeed 150 int rightDistance = 0, leftDistance = 0, middleDistance</pre>	on doit inclure une librairie (du code supplémentaire) pour gérer le servo moteur radar travaille avec ces deux broches, une qui envoie l'onde l'autre qui attend l'écho les moteurs vitesse 150 (entre 0 et 255) on crée des variables pour les ditances de droite,
---	--

<pre> = 0; void forward(){ analogWrite(ENA, carSpeed); analogWrite(ENB, carSpeed); digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW); digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH); Serial.println("Forward"); } void back() { analogWrite(ENA, carSpeed); analogWrite(ENB, carSpeed); digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH); digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW); Serial.println("Back"); } void left() { analogWrite(ENA, carSpeed); analogWrite(ENB, carSpeed); digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH); digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH); Serial.println("Left"); } void right() { analogWrite(ENA, carSpeed); analogWrite(ENB, carSpeed); digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW); digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW); Serial.println("Right"); } void stop() { digitalWrite(ENA, LOW); digitalWrite(ENB, LOW); Serial.println("Stop!"); } //Ultrasonic distance measurement Sub function int Distance_test() { digitalWrite(Trig, LOW); delayMicroseconds(2); digitalWrite(Trig, HIGH); </pre>	gauche, centre. On les met à 0. avancer reculer tourner à gauche tourner à droite arrêter fonction qui mesure la distance on s'assure que le Trig (broche A5) est à 0 on attend un peu on met la broche Trig à 1
---	---

<pre> delayMicroseconds(20); digitalWrite(Trig, LOW); float Fdistance = pulseIn(Echo, HIGH); Fdistance= Fdistance / 58; return (int)Fdistance; } void setup() { myservo.attach(3); // attach servo on pin 3 to servo object Serial.begin(9600); pinMode(Echo, INPUT); pinMode(Trig, OUTPUT); pinMode(IN1, OUTPUT); pinMode(IN2, OUTPUT); pinMode(IN3, OUTPUT); pinMode(IN4, OUTPUT); pinMode(ENA, OUTPUT); pinMode(ENB, OUTPUT); stop(); } void loop() { myservo.write(90); //set servo position according to scaled value delay(500); middleDistance = Distance_test(); if(middleDistance <= 20) { stop(); delay(500); myservo.write(10); delay(1000); rightDistance = Distance_test(); delay(500); myservo.write(90); delay(1000); myservo.write(180); delay(1000); leftDistance = Distance_test(); delay(500); myservo.write(90); delay(1000); if(rightDistance > leftDistance) { right(); delay(360); } else if(rightDistance < leftDistance) { </pre>	20 microsecondes on met la broche trig à 0 on crée la variable Fdistance que l'on met à la valeur du temps qu'a mis le son à faire l'aller retour On calcule les cm. On renvoie la distance où est le servo, broche 3 on initie la communication en port série (voir le moniteur série) les entrées et sorties on met le servo à 90° on attend un peu on mesure la distance, que l'on place dans la variable middleDistance si middleDistance est plus petit ou égale à 20 (cm), arrêt attend on oriente le servo à 10° attend 1s on place la valeur de distance dans la rightDistance attend on oriente le servo à 90° attend on oriente le servo à 180° on attend on place la distance lue dans leftDistance attend oriente le servo à 90° attend si rightDistance est plus grand que leftDistance on tourne a droite si rightDistance est plus petite que leftDistance on
--	--

<pre> left(); delay(360); } else if((rightDistance <= 20) (leftDistance <= 20)) { back(); delay(180); } else { forward(); } } else { forward(); } } </pre>	tourne à gauche si les deux distances sont plus petites que 20 cm, on recule sinon on avance
---	---

Utilisation du iPad pour contrôler le véhicule avec la manette de l'application



ELEGOO BLE TOOL

OU



Avant d'utiliser l'application, on doit retirer le module bluetooth et brancher le véhicule à l'aide du câble USB. Il faut par la suite télécharger le programme suivant qui suit. Une fois le téléversement terminé, débrancher le câble, installer le module bluetooth, mettre l'interrupteur à ON et utiliser l'application de votre choix sur votre iPad.

(Il est disponible dans le CD qui vient avec le kit. E:\Francias\Leçon 2 - Voiture bluetooth\bluetooth_car)

<pre> //www.elegoo.com #define ENA 5 #define ENB 6 #define IN1 7 #define IN2 8 #define IN3 9 #define IN4 11 #define LED 13 unsigned char carSpeed = 150; bool state = LOW; char getstr; </pre>	Contrôler la vitesse de moteurs pin 7 avancer roue gauche pin 8 reculer roue gauche pin 9 reculer roue droite pin 11 avancer roue droite Contrôle la vitesse des moteurs lorsqu'on tourne à droite ou à gauche. La valeur de 150 est un peu basse, je recommande de la mettre à 180.
--	---

Avancer

Reculer

Tourner à gauche

Tourner à droite

Arrêter

<pre>void setup() { Serial.begin(9600); pinMode(LED, OUTPUT); pinMode(IN1,OUTPUT); pinMode(IN2,OUTPUT); pinMode(IN3,OUTPUT); pinMode(IN4,OUTPUT); pinMode(ENA,OUTPUT); pinMode(ENB,OUTPUT); stop(); }</pre>	Définir les sorties
<pre>void loop() { getstr = Serial.read(); switch(getstr){ case 'f': forward(); break; case 'b': back(); break; case 'l': left(); break; case 'r': right(); break; case 's': stop(); break; case 'a': stateChange(); break; default: break; } }</pre>	Identification des boutons pour la manette de l'application iPad.

Utilisation la manette IR pour contrôler le véhicule

En premier ajouter la bibliothèque IRremote disponible sur le CD (E:\Francias\Leçon 3 - Controle Voiture Infrarouge) dans l'IDE Arduino.

Brancher le véhicule avec le câble USB et téléverser le code qui suit. Il est disponible sur le CD (E:\Francias\Leçon 3 - Controle Voiture Infrarouge\Infrared_remote_control_car)

Un fois le téléversement terminé, débrancher le câble USB et utiliser le manette pour diriger le véhicule. Se contrôle bien si on est derrière le véhicule mais pas si on est devant.

<pre>//www.elegoo.com #include <IRremote.h> ////////// IR REMOTE CODES ////////// #define F 16736925 // FORWARD #define B 16754775 // BACK #define L 16720605 // LEFT</pre>	Code pour le contrôle IR
---	--------------------------

```
#define R 16761405 // RIGHT
#define S 16712445 // STOP
#define UNKNOWN_F 5316027 //
FORWARD
#define UNKNOWN_B 2747854299 // BACK
#define UNKNOWN_L 1386468383 // LEFT
#define UNKNOWN_R 553536955 // RIGHT
#define UNKNOWN_S 3622325019 // STOP
```

```
#define RECV_PIN 12
```

```
/*define channel enable output pins*/
#define ENA 5 // Left wheel speed
#define ENB 6 // Right wheel speed
/*define logic control output pins*/
#define IN1 7 // Left wheel forward
#define IN2 8 // Left wheel reverse
#define IN3 9 // Right wheel reverse
#define IN4 11 // Right wheel forward
#define carSpeed 150 // initial speed of car >=0 to
<=255
```

```
IRrecv irrecv(RECV_PIN);
decode_results results;
unsigned long val;
unsigned long preMillis;
```

```
/**
 * BEGIN DEFINE FUNCTIONS
 */
```

```
void forward(){
  digitalWrite(ENA,HIGH);
  digitalWrite(ENB,HIGH);
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
  Serial.println("go forward!");
}
```

```
void back(){
  digitalWrite(ENA,HIGH);
  digitalWrite(ENB,HIGH);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
  Serial.println("go back!");
}
```

```
void left(){
  analogWrite(ENA,carSpeed);
  analogWrite(ENB,carSpeed);
  digitalWrite(IN1,LOW);
```

Contrôler la vitesse de moteurs

pin 7 avancer roue gauche
 pin 8 reculer roue gauche
 pin 9 reculer roue droite
 pin 11 avancer roue droite
 Contrôle la vitesse des moteurs lorsqu'on tourne à droite ou à gauche. La valeur de 150 est un peu basse, je recommande de la mettre à 180.

Avancer

Reculer


```
if(millis() - preMillis > 500){
  stop();
  preMillis = millis();
}
}
```

Archives

Voici celui pour ralentir:

```
for(int i=255; i>=0;i--){
  analogWrite(ENB,i);
  analogWrite(ENA,i);
  delay(20);}
```

et celui pour accélérer:

```
for(int i=0; i<=255;i++){
  analogWrite(ENB,i);
  analogWrite(ENA,i);
  delay(20);}
```

Pouvez-vous donc me dire si je comprends bien? Prenons l'exemple du ralentissement. int pour intervalle. Le int est pour «variable nombre entier», auquel on ajoute 1 (de 0 à 255 possibilités pour le analogwrite) à chaque fois qu'on recommence la boucle.

le premier i est la vitesse initiale, le 2e dit qu'on s'en va vers 0.

Oui, vitesse initiale = max = 255, puis elle diminue à chaque 20 ms de 1.

le -- dit qu'on diminue entre le 1er i et le 2e i?

En fait on soustrait 1 à i à chaque fois qu'on entre dans la boucle.

on met les moteurs en analog car en digital on peut juste être ouvert ou fermé.
oui.

et on met le i après ENA et ENB pour dire que ces 2 moteurs vont suivre la variable i.

c'est ça, analogwrite (port, puissance) où le port doit être défini dans le haut du programme (ENA et ENB).

Est-ce bien ça? et savez-vous pourquoi un i et pas une autre lettre?

Ça pourrait probablement être une autre variable que i. Comme la variable eeee suivante :

int eeee; //on définit eeee comme entière

```
void setup() {  
}
```

```
void loop() {  
  for (eeee = 1; eeee <= 10; eeee++) {  
    faire quelque chose  
  }  
}
```