

```

#include <Servo.h>
float rc_pulse3, rc_pulse5, rc_pulse6, motor_pwm9;
Servo esc1;
Servo esc2;
int positionchannel;
void setup() {
  esc1.attach(9);
  esc2.attach(10);
  delay(15);
  Serial.begin(115200);
  esc1.write(0);
  esc2.write(0);
  delay(3000);
  esc1.write(70);
  esc2.write(70);
  delay(5);

  pinMode(3, INPUT);
  pinMode(5, INPUT);
  pinMode(6, INPUT);
}

int position(int val){
  // position SA, SB, SC, SD, SF, min = 979, middle = 1480, max = 1992
  // ces valeurs oscillent dans un intervalle de +- 10 autour de la valeur reel

  if ( val > 950 and val < 1050 ) {
    return 0;
  }
  else if ( val > 1400 and val < 1500 ) {
    return 1;
  }
  else if ( val > 1900 and val < 2000 ) {
    return 2;
  }
  else {
    return 3; // gerer si jamais il y a une erreur
  }
}

float map(float x, float a, float b, float c, float d) {
  return c + (x - a) * (d - c) / (b - a);
}

void runMotor(float cmdl, float cmdr){
  esc1.write(cmdl);
  esc2.write(cmdr);

  Serial.print(" You sent me : ");
  Serial.print(" data 1 : ");
  Serial.print(cmdl);
  Serial.print(" data 2 : ");
  Serial.println(cmdr);
}

```

→ Librairie permettant de contrôler les servomoteurs.

→ permet de connaître la position du bouton SD de la télécommande

→ paramétrage de la valeur min à 70.

} position bas

} position du milieu

} position haut

→ Fonction permettant de convertir une valeur de l'intervalle [a,b] dans l'intervalle [c,d].

→ envoi de la valeur cmdl au moteur de gauche et cmdr au moteur de droite.

```
void loop() {
```

```
if( Serial.available()>0 ){ //if( Serial.available()>0 and position(rc_pulse6) != 1
String data = Serial.readStringUntil("\n"); → lecture des données
```

```
Serial.print(data.length());
```

```
if ( data.length() > 2 ) { →
```

```
int index = data.indexOf('$');
```

```
String data1 = data.substring(0,index); → extraction de cmdl
```

```
String data2 = data.substring(index+1); → extraction de cmdr
```

```
//float cmdl = data1.toFloat();
```

```
//float cmdr = data2.toFloat();
```

```
float cmdl = map ( data1.toFloat() , 0 , 100 , 70 , 80 );
```

```
float cmdr = map ( data2.toFloat() , 0 , 100 , 70 , 80 );
```

```
runMotor(cmdl, cmdr);
```

```
}
```

```
else {
```

```
Serial.print(" les vitesses sont : es1 ");
```

```
Serial.print(esc1.read());
```

```
Serial.print(" esc 2 : ");
```

```
Serial.println(esc2.read());
```

```
}
```

```
}
```

Si la longueur est supérieure à 2 (cela signifie que la donnée a été envoyée par la fonction)

extraction de cmdl
extraction de cmdr

Conversion des valeurs dans l'intervalle [70,80], puisque les valeurs de l'utilisateur seront comprises entre 0 et 100 (intervalle plus maniable que [70,80]).

lorsque data.length ≤ 2 alors la donnée est envoyée par get_arduino_cmdl_cmdr et donc l'utilisateur veut les vitesses cmdl et cmdr.

Pilotage des moteurs grâce à la réception par l'intermédiaire de l'arduino

```
rc_pulse6 = pulseIn(6, HIGH);
```

```
positionchannel = position(rc_pulse6); → position de SD
```

```
while( positionchannel == 1 ) { →
```

```
rc_pulse3 = pulseIn(3, HIGH);
```

```
rc_pulse5 = pulseIn(5, HIGH);
```

```
rc_pulse6 = pulseIn(6, HIGH); // SD
```

```
Serial.print(rc_pulse3);
```

```
Serial.print(" rc_pulse5 ");
```

```
Serial.print(rc_pulse5);
```

```
Serial.print(" rc_pulse6 ");
```

```
Serial.println(rc_pulse6);
```

```
if (position(rc_pulse3) == 1 ) { →
```

```
if ( rc_pulse5 > 1480 ) {
```

```
esc1.write(map ( rc_pulse5 , 1500 , 2000 , 70 , 80 )); → on tourne à droite
```

```
}
```

```
else {
```

```
esc2.write(map ( -rc_pulse5 , -1500 , -980 , 70 , 80 )); → on tourne à gauche
```

```
}
```

```
}
```

```
else {
```

```
esc1.write(map ( rc_pulse3 , 1500 , 2000 , 70 , 80 ));
```

```
esc2.write(map ( rc_pulse3 , 1500 , 2000 , 70 , 80 ));
```

```
}
```

```
positionchannel = position(rc_pulse6);
```

```
}
```

```
}
```

→ la commande CH3 n'est pas actionnée position CH3 au milieu

Commande CH3 actionnée, on force tout droit

Commandes des moteurs grâce à la télécommande