

programmes bras robots exemples

brasrobot001.ino

```
#include <Servo.h>

Servo base; Servo epaule; Servo coude; Servo poignet; Servo rotapince;
Servo ouvepince;

int J1, J2, J3, J4;
int BP1, BP2;

int pulse1, pulse2, pulse3, pulse4, pulse5, pulse6;
int pas1, pas2, pas3, pas4;

void setup() {

  base.attach(6); //sortie numérique 6
  epaule.attach(3); //sortie numérique 3
  coude.attach(9); //sortie numérique 9
  poignet.attach(5); //sortie numérique 5
  rotapince.attach(11); //sortie numérique 11
  ouvepince.attach(10); //sortie numérique 10

  pinMode(4, INPUT_PULLUP); //sortie numérique 4
  pinMode(2, INPUT_PULLUP); //sortie numérique 2
  //utilise le VCC et la résistance de pullup de l'Arduino >> évite de
  rajouter un VCC et une résistance dans le montage

  J1 = J2 = J3 = J4 = 0;
  BP1 = BP2 = 1;

  //initialisation des moteurs en position de départ

  pulse1 = pulse2 = pulse3 = pulse4 = pulse5 = 1500; //90°
  pulse6 = 2000; //180°
}

void loop() {

  J1 = analogRead(A2); J3 = analogRead(A0); //Joystick gauche mouvement
horizontal et vertical
  J2 = analogRead(A3); J4 = analogRead(A1); //Joystick droit mouvement
vertical et horizontal
  BP1 = digitalRead(4); BP2 = digitalRead(2); //Bouton poussoir gauche
et droit
```

```
base.writeMicroseconds(pulse1);
epaule.writeMicroseconds(pulse2);
coude.writeMicroseconds(pulse3);
poignet.writeMicroseconds(pulse4);
rotapince.writeMicroseconds(pulse5);
ouvepince.writeMicroseconds(pulse6);

//lecture J1
if (J1 > 1000) {
  pulse6 -= 10; //action sur la fermeture de la pince
  if (pulse6 < 1650) {
    pulse6 = 1650; /*[2]*/
  }
}
else
{ //action sur la base
  pas1 = map(J1, 200, 800, -10, 10); //[0]
  if (pas1 < 5 && pas1 > -5) {} //[1]
  else
  {
    pulse1 += pas1;
    if (pulse1 > 2200) {
      pulse1 = 2200; //[2]
    }
    if (pulse1 < 800) {
      pulse1 = 800; //[3]
    }
  }
}

//lecture J2 et action sur le coude
pas3 = map(J2, 100, 900, -10, 10);
if (pas3 < 5 && pas3 > -5) {} //[1]
else
{
  pulse3 += pas3;
  if (pulse3 > 2100) {
    pulse3 = 2100; //[2]
  }
  if (pulse3 < 900) {
    pulse3 = 900; //[3]
  }
}

//lecture J3 et action sur l'épaule
if (J3 > 1000) {
  pulse6 += 10; //action sur la l'ouverture de la pince
  if (pulse6 > 2400) {
    pulse6 = 2400; /*[3]*/
  }
}
```

```

    }
}
else
{
    pas2 = map(J3, 200, 800, -10, 10);
    if (pas2 < 5 && pas2 > -5) {} //[1]
    else
    {
        pulse2 -= pas2;
        if (pulse2 > 2200) {
            pulse2 = 2200; //[2]
        }
        if (pulse2 < 1000) {
            pulse2 = 1000; //[3]
        }
    }
}

//lecture J4 et action sur le poignet
pas4 = map(J4, 100, 900, -10, 10);
if (pas4 < 5 && pas4 > -5) {} //[1]
else
{
    pulse4 -= pas4;
    if (pulse4 > 2100) {
        pulse4 = 2100; //[2]
    }
    if (pulse4 < 900) {
        pulse4 = 900; //[3]
    }
}

//lecture BP1 et BP2, et action sur la rotation de la pince
if (BP1 == LOW) {
    pulse5 -= 10;
    if (pulse5 < 600) {
        pulse5 = 600; /*[2]*/
    }
}
if (BP2 == LOW) {
    pulse5 += 10;
    if (pulse5 > 2100) {
        pulse5 = 2100; /*[3]*/
    }
}

delay(15);
}

/* [0] Un joystick peut varier de 0 à 1023. Suivant la plage qui nous

```

intéresse

l'instruction "map()" va permettre de faire une mise à l'échelle

[1] offset

[2] borne minimale : début de course

*[3] borne maximale : fin de course */*

From:

<https://www.magenealogie.chanterie37.fr/www/fablab37110/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link:

<https://www.magenealogie.chanterie37.fr/www/fablab37110/doku.php?id=start:arduino:robots:programmes&rev=1650639584>

Last update: **2023/01/27 16:08**

