

# Phase 3 - Programmation du bras robotique

## Cinématique d'un bras robot

La cinématique est un domaine de la mécanique qui se concentre sur l'étude des mouvements. Ça tombe bien, le but d'un robot étant de bouger ! L'étude cinématique consiste donc à déterminer les positions, vitesses et accélérations des différentes parties d'un robot, sans prendre en compte les forces qui causent ces mouvements (puissance des moteurs par exemple). Un robot, souvent décrit comme une machine automatisée, peut effectuer des tâches complexes apportant une réelle valeur ajoutée dans divers secteurs.

Notre robot a 4 moteurs, l'objectif est de les actionner pour :

- Fermer la pince pour saisir un objet
- Déplacer la pince pour déplacer l'objet
- Ouvrir la pince pour lâcher l'objet

Les 4 moteurs contrôlent 4 mouvements possibles :

- Moteur 1 : Base
- Moteur 2 : Coude
- Moteur 3 : Épaule
- Moteur 4 : Pince

<https://www.youtube.com/watch?v=xQkPjDEbFoU>

Attention chaque Servomoteur possède des positions maximums et minimums à bien respecter pour pas que ce dernier ne fonctionne plus.

Voici un tableau qui regroupe ces valeurs si vous l'avez monter comme dans la phase 1 :

	Tableau des valeurs maximales et minimales des servomoteurs			
	Base	Épaule	Coude	Pince
Position maximale	180	150	180	160
Position minimale	0	0	100	90

Pour ouvrir et fermer la pince c'est donc facile, on actionne le moteur 4.

Espace Cartésien : les mouvements de la pince et des objets

Espace articulaire : les mouvements des moteurs

La question principale qu'on se pose c'est comment actionner mes moteurs pour faire bouger ma pince comme je veux ? C'est le modèle cinématique inverse.

On peut aussi se demander : si je bouge mes moteurs à telle position, quelle sera la position de ma pince ? C'est le modèle cinématique direct.

Voici un code mblock qui permet de prendre un objet à une où le coude est à la position 140 et l'épaule à la position 150.

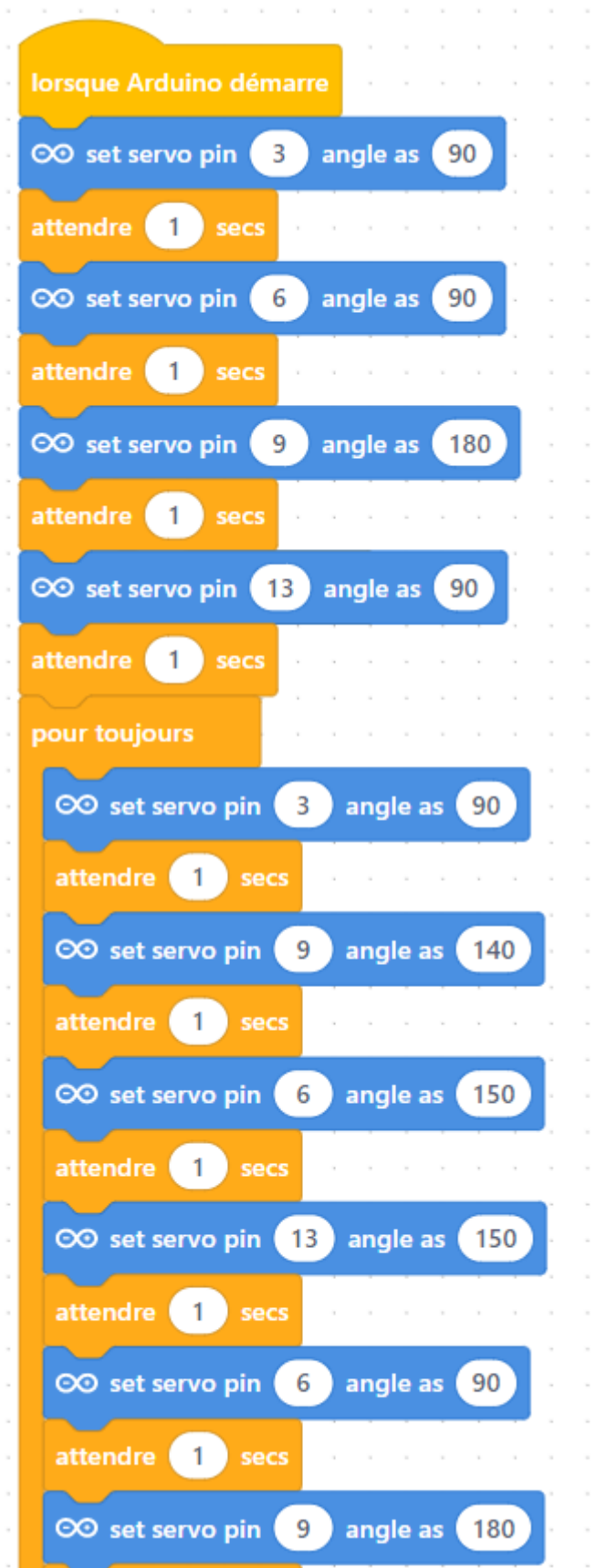
Epaule : Pin 6

Coude : Pin 9

Base : Pin 3

Pince : Pin 13

**Partie 1 :**



**Partie 2 :**



## Déplacement à travers une liste de points

```
/*  
    DIY Arduino Robot Arm Smartphone Control  
    by Dejan, www.HowToMechatronics.com  
*/  
  
#include <SoftwareSerial.h>  
#include <Servo.h>  
  
Servo servo01;
```

```

Servo servo02;
Servo servo03;
Servo servo04;
Servo servo05;
Servo servo06;

SoftwareSerial Bluetooth(3, 4); // Arduino(RX, TX) - HC-05 Bluetooth (TX, RX)

int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos; // current position
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos; // previous
position
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50];
// for storing positions/steps
int speedDelay = 20;
int index = 0;
String dataIn = "";

void setup() {
    servo01.attach(5);
    servo02.attach(6);
    servo03.attach(7);
    servo04.attach(8);
    servo05.attach(9);
    servo06.attach(10);

    Bluetooth.begin(38400); // Default baud rate of the Bluetooth module
    Bluetooth.setTimeout(1);
    delay(20);
    // Robot arm initial position
    servo1PPos = 90;
    servo01.write(servo1PPos);
    servo2PPos = 150;
    servo02.write(servo2PPos);
    servo3PPos = 35;
    servo03.write(servo3PPos);
    servo4PPos = 140;
    servo04.write(servo4PPos);
    servo5PPos = 85;
    servo05.write(servo5PPos);
    servo6PPos = 80;
    servo06.write(servo6PPos);
}

```

```

}

void loop() {
  // Check for incoming data
  if (Bluetooth.available() > 0) {
    dataIn = Bluetooth.readString(); // Read the data as string

    // If "Waist" slider has changed value - Move Servo 1 to position
    if (dataIn.startsWith("s1")) {
      String dataInS = dataIn.substring(2, dataIn.length()); // Extract only the number. E.g.
from "s1120" to "120"
      servo1Pos = dataInS.toInt(); // Convert the string into integer
      // We use for loops so we can control the speed of the servo
      // If previous position is bigger then current position
      if (servo1PPos > servo1Pos) {
        for ( int j = servo1PPos; j >= servo1Pos; j--) { // Run servo down
          servo01.write(j);
          delay(20); // defines the speed at which the servo rotates
        }
      }
      // If previous position is smaller then current position
      if (servo1PPos < servo1Pos) {
        for ( int j = servo1PPos; j <= servo1Pos; j++) { // Run servo up
          servo01.write(j);
          delay(20);
        }
      }
      servo1PPos = servo1Pos; // set current position as previous position
    }

    // Move Servo 2
    if (dataIn.startsWith("s2")) {
      String dataInS = dataIn.substring(2, dataIn.length());
      servo2Pos = dataInS.toInt();

      if (servo2PPos > servo2Pos) {
        for ( int j = servo2PPos; j >= servo2Pos; j--) {
          servo02.write(j);
          delay(50);
        }
      }
    }
  }
}

```

```

    }
    if (servo2PPos < servo2Pos) {
        for ( int j = servo2PPos; j <= servo2Pos; j++) {
            servo02.write(j);
            delay( 50);
        }
    }
    servo2PPos = servo2Pos;
}

// Move Servo 3
if (dataIn.startsWith("s3")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo3Pos = dataInS.toInt();
    if (servo3PPos > servo3Pos) {
        for ( int j = servo3PPos; j >= servo3Pos; j--) {
            servo03.write(j);
            delay( 30);
        }
    }
    if (servo3PPos < servo3Pos) {
        for ( int j = servo3PPos; j <= servo3Pos; j++) {
            servo03.write(j);
            delay( 30);
        }
    }
    servo3PPos = servo3Pos;
}

// Move Servo 4
if (dataIn.startsWith("s4")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo4Pos = dataInS.toInt();
    if (servo4PPos > servo4Pos) {
        for ( int j = servo4PPos; j >= servo4Pos; j--) {
            servo04.write(j);
            delay( 30);
        }
    }
    if (servo4PPos < servo4Pos) {
        for ( int j = servo4PPos; j <= servo4Pos; j++) {
            servo04.write(j);

```

```

        delay(30);
    }
}
servo4PPos = servo4Pos;
}
// Move Servo 5
if (dataIn.startsWith("s5")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo5Pos = dataInS.toInt();
    if (servo5PPos > servo5Pos) {
        for ( int j = servo5PPos; j >= servo5Pos; j--) {
            servo05.write(j);
            delay(30);
        }
    }
    if (servo5PPos < servo5Pos) {
        for ( int j = servo5PPos; j <= servo5Pos; j++) {
            servo05.write(j);
            delay(30);
        }
    }
    servo5PPos = servo5Pos;
}
// Move Servo 6
if (dataIn.startsWith("s6")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo6Pos = dataInS.toInt();
    if (servo6PPos > servo6Pos) {
        for ( int j = servo6PPos; j >= servo6Pos; j--) {
            servo06.write(j);
            delay(30);
        }
    }
    if (servo6PPos < servo6Pos) {
        for ( int j = servo6PPos; j <= servo6Pos; j++) {
            servo06.write(j);
            delay(30);
        }
    }
    servo6PPos = servo6Pos;
}

```



```

}
// If button "SAVE" is pressed
if (dataIn.startsWith("SAVE")) {
    servo01SP[index] = servo1PPos; // save position into the array
    servo02SP[index] = servo2PPos;
    servo03SP[index] = servo3PPos;
    servo04SP[index] = servo4PPos;
    servo05SP[index] = servo5PPos;
    servo06SP[index] = servo6PPos;
    index++; // Increase the array index
}
// If button "RUN" is pressed
if (dataIn.startsWith("RUN")) {
    runservo(); // Automatic mode - run the saved steps
}
// If button "RESET" is pressed
if (dataIn == "RESET") {
    memset(servo01SP, 0, sizeof(servo01SP)); // Clear the array data to 0
    memset(servo02SP, 0, sizeof(servo02SP));
    memset(servo03SP, 0, sizeof(servo03SP));
    memset(servo04SP, 0, sizeof(servo04SP));
    memset(servo05SP, 0, sizeof(servo05SP));
    memset(servo06SP, 0, sizeof(servo06SP));
    index = 0; // Index to 0
}
}
}

// Automatic mode custom function - run the saved steps
void runservo() {
    while (dataIn != "RESET") { // Run the steps over and over again until "RESET" button is
pressed
        for (int i = 0; i <= index - 2; i++) { // Run through all steps(index)
            if (Bluetooth.available() > 0) { // Check for incoming data
                dataIn = Bluetooth.readString();
                if (dataIn == "PAUSE") { // If button "PAUSE" is pressed
                    while (dataIn != "RUN") { // Wait until "RUN" is pressed again
                        if (Bluetooth.available() > 0) {
                            dataIn = Bluetooth.readString();
                            if (dataIn == "RESET") {

```

```

        break;
    }
}
}
}
// If speed slider is changed
if (dataIn.startsWith("ss")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    speedDelay = dataInS.toInt(); // Change servo speed (delay time)
}
}
// Servo 1
if (servo01SP[i] == servo01SP[i + 1]) {
}
if (servo01SP[i] > servo01SP[i + 1]) {
    for ( int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {
        servo01.write(j);
        delay( speedDelay);
    }
}
if (servo01SP[i] < servo01SP[i + 1]) {
    for ( int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {
        servo01.write(j);
        delay( speedDelay);
    }
}

// Servo 2
if (servo02SP[i] == servo02SP[i + 1]) {
}
if (servo02SP[i] > servo02SP[i + 1]) {
    for ( int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {
        servo02.write(j);
        delay( speedDelay);
    }
}
if (servo02SP[i] < servo02SP[i + 1]) {
    for ( int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {
        servo02.write(j);
        delay( speedDelay);
    }
}

```

```

    }
}

// Servo 3
if (servo03SP[i] == servo03SP[i + 1]) {
}
if (servo03SP[i] > servo03SP[i + 1]) {
    for ( int j = servo03SP[i]; j >= servo03SP[i + 1]; j--) {
        servo03.write(j);
        delay( speedDelay);
    }
}
if (servo03SP[i] < servo03SP[i + 1]) {
    for ( int j = servo03SP[i]; j <= servo03SP[i + 1]; j++) {
        servo03.write(j);
        delay( speedDelay);
    }
}

// Servo 4
if (servo04SP[i] == servo04SP[i + 1]) {
}
if (servo04SP[i] > servo04SP[i + 1]) {
    for ( int j = servo04SP[i]; j >= servo04SP[i + 1]; j--) {
        servo04.write(j);
        delay( speedDelay);
    }
}
if (servo04SP[i] < servo04SP[i + 1]) {
    for ( int j = servo04SP[i]; j <= servo04SP[i + 1]; j++) {
        servo04.write(j);
        delay( speedDelay);
    }
}

// Servo 5
if (servo05SP[i] == servo05SP[i + 1]) {
}
if (servo05SP[i] > servo05SP[i + 1]) {
    for ( int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {

```

```

        servo05.write(j);
        delay( speedDelay);
    }
}
if (servo05SP[i] < servo05SP[i + 1]) {
    for ( int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {
        servo05.write(j);
        delay( speedDelay);
    }
}

// Servo 6
if (servo06SP[i] == servo06SP[i + 1]) {
}
if (servo06SP[i] > servo06SP[i + 1]) {
    for ( int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {
        servo06.write(j);
        delay( speedDelay);
    }
}
if (servo06SP[i] < servo06SP[i + 1]) {
    for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {
        servo06.write(j);
        delay( speedDelay);
    }
}
}
}
}
}

```

## Pilotage bluetooth et sauvegarde de points de la trajectoire

```

/*
    DIY Arduino Robot Arm Smartphone Control
    by Dejan, www.HowToMechatronics.com

```

```

*/

#include <SoftwareSerial.h>
#include <Servo.h>

Servo servo01;
Servo servo02;
Servo servo03;
Servo servo04;
Servo servo05;
Servo servo06;

SoftwareSerial Bluetooth(3, 4); // Arduino(RX, TX) - HC-05 Bluetooth (TX, RX)

int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos; // current position
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos; // previous
position
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50];
// for storing positions/steps
int speedDelay = 20;
int index = 0;
String dataIn = "";

void setup() {
    servo01.attach(5);
    servo02.attach(6);
    servo03.attach(7);
    servo04.attach(8);
    servo05.attach(9);
    servo06.attach(10);
    Bluetooth.begin(38400); // Default baud rate of the Bluetooth module
    Bluetooth.setTimeout(1);
    delay(20);
    // Robot arm initial position
    servo1PPos = 90;
    servo01.write(servo1PPos);
    servo2PPos = 150;
    servo02.write(servo2PPos);
    servo3PPos = 35;

```

```

servo03.write(servo3PPos);
servo4PPos = 140;
servo04.write(servo4PPos);
servo5PPos = 85;
servo05.write(servo5PPos);
servo6PPos = 80;
servo06.write(servo6PPos);
}

void loop() {
  // Check for incoming data
  if (Bluetooth.available() > 0) {
    dataIn = Bluetooth.readString(); // Read the data as string

    // If "Waist" slider has changed value - Move Servo 1 to position
    if (dataIn.startsWith("s1")) {
      String dataInS = dataIn.substring(2, dataIn.length()); // Extract only the number. E.g.
      from "s1120" to "120"
      servo1Pos = dataInS.toInt(); // Convert the string into integer
      // We use for loops so we can control the speed of the servo
      // If previous position is bigger then current position
      if (servo1PPos > servo1Pos) {
        for ( int j = servo1PPos; j >= servo1Pos; j--) { // Run servo down
          servo01.write(j);
          delay(20); // defines the speed at which the servo rotates
        }
      }
      // If previous position is smaller then current position
      if (servo1PPos < servo1Pos) {
        for ( int j = servo1PPos; j <= servo1Pos; j++) { // Run servo up
          servo01.write(j);
          delay(20);
        }
      }
      servo1PPos = servo1Pos; // set current position as previous position
    }

    // Move Servo 2
    if (dataIn.startsWith("s2")) {

```

```

String dataInS = dataIn.substring(2, dataIn.length());
servo2Pos = dataInS.toInt();

if (servo2PPos > servo2Pos) {
    for ( int j = servo2PPos; j >= servo2Pos; j--) {
        servo02.write(j);
        delay(50);
    }
}
if (servo2PPos < servo2Pos) {
    for ( int j = servo2PPos; j <= servo2Pos; j++) {
        servo02.write(j);
        delay(50);
    }
}
servo2PPos = servo2Pos;
}

// Move Servo 3
if (dataIn.startsWith("s3")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo3Pos = dataInS.toInt();
    if (servo3PPos > servo3Pos) {
        for ( int j = servo3PPos; j >= servo3Pos; j--) {
            servo03.write(j);
            delay(30);
        }
    }
    if (servo3PPos < servo3Pos) {
        for ( int j = servo3PPos; j <= servo3Pos; j++) {
            servo03.write(j);
            delay(30);
        }
    }
    servo3PPos = servo3Pos;
}

// Move Servo 4
if (dataIn.startsWith("s4")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo4Pos = dataInS.toInt();

```

```

    if (servo4PPos > servo4Pos) {
        for ( int j = servo4PPos; j >= servo4Pos; j--) {
            servo04.write(j);
            delay( 30);
        }
    }
    if (servo4PPos < servo4Pos) {
        for ( int j = servo4PPos; j <= servo4Pos; j++) {
            servo04.write(j);
            delay( 30);
        }
    }
    servo4PPos = servo4Pos;
}

// Move Servo 5
if (dataIn.startsWith("s5")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo5Pos = dataInS.toInt();
    if (servo5PPos > servo5Pos) {
        for ( int j = servo5PPos; j >= servo5Pos; j--) {
            servo05.write(j);
            delay( 30);
        }
    }
    if (servo5PPos < servo5Pos) {
        for ( int j = servo5PPos; j <= servo5Pos; j++) {
            servo05.write(j);
            delay( 30);
        }
    }
    servo5PPos = servo5Pos;
}

// Move Servo 6
if (dataIn.startsWith("s6")) {
    String dataInS = dataIn.substring(2, dataIn.length());
    servo6Pos = dataInS.toInt();
    if (servo6PPos > servo6Pos) {
        for ( int j = servo6PPos; j >= servo6Pos; j--) {
            servo06.write(j);

```



```

        delay(30);
    }
}
if (servo6PPos < servo6Pos) {
    for ( int j = servo6PPos; j <= servo6Pos; j++) {
        servo06.write(j);
        delay(30);
    }
}
servo6PPos = servo6Pos;
}
// If button "SAVE" is pressed
if (dataIn.startsWith("SAVE")) {
    servo01SP[index] = servo1PPos; // save position into the array
    servo02SP[index] = servo2PPos;
    servo03SP[index] = servo3PPos;
    servo04SP[index] = servo4PPos;
    servo05SP[index] = servo5PPos;
    servo06SP[index] = servo6PPos;
    index++; // Increase the array index
}
// If button "RUN" is pressed
if (dataIn.startsWith("RUN")) {
    runservo(); // Automatic mode - run the saved steps
}
// If button "RESET" is pressed
if ( dataIn == "RESET") {
    memset(servo01SP, 0, sizeof(servo01SP)); // Clear the array data to 0
    memset(servo02SP, 0, sizeof(servo02SP));
    memset(servo03SP, 0, sizeof(servo03SP));
    memset(servo04SP, 0, sizeof(servo04SP));
    memset(servo05SP, 0, sizeof(servo05SP));
    memset(servo06SP, 0, sizeof(servo06SP));
    index = 0; // Index to 0
}
}
}

// Automatic mode custom function - run the saved steps

```

```

void runservo() {
    while (dataIn != "RESET") {    // Run the steps over and over again until "RESET" button is
pressed
        for (int i = 0; i <= index - 2; i++) {    // Run through all steps(index)
            if (Bluetooth.available() > 0) {        // Check for incoming data
                dataIn = Bluetooth.readString();
                if ( dataIn == "PAUSE") {            // If button "PAUSE" is pressed
                    while (dataIn != "RUN") {        // Wait until "RUN" is pressed again
                        if (Bluetooth.available() > 0) {
                            dataIn = Bluetooth.readString();
                            if ( dataIn == "RESET") {
                                break;
                            }
                        }
                    }
                }
            }
            // If speed slider is changed
            if (dataIn.startsWith("ss")) {
                String dataInS = dataIn.substring(2, dataIn.length());
                speedDelay = dataInS.toInt(); // Change servo speed (delay time)
            }
        }
        // Servo 1
        if (servo01SP[i] == servo01SP[i + 1]) {
        }
        if (servo01SP[i] > servo01SP[i + 1]) {
            for ( int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {
                servo01.write(j);
                delay( speedDelay);
            }
        }
        if (servo01SP[i] < servo01SP[i + 1]) {
            for ( int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {
                servo01.write(j);
                delay( speedDelay);
            }
        }
    }

    // Servo 2

```

```

if (servo02SP[i] == servo02SP[i + 1]) {
}
if (servo02SP[i] > servo02SP[i + 1]) {
    for ( int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {
        servo02.write(j);
        delay( speedDelay);
    }
}
if (servo02SP[i] < servo02SP[i + 1]) {
    for ( int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {
        servo02.write(j);
        delay( speedDelay);
    }
}

// Servo 3
if (servo03SP[i] == servo03SP[i + 1]) {
}
if (servo03SP[i] > servo03SP[i + 1]) {
    for ( int j = servo03SP[i]; j >= servo03SP[i + 1]; j--) {
        servo03.write(j);
        delay( speedDelay);
    }
}
if (servo03SP[i] < servo03SP[i + 1]) {
    for ( int j = servo03SP[i]; j <= servo03SP[i + 1]; j++) {
        servo03.write(j);
        delay( speedDelay);
    }
}

// Servo 4
if (servo04SP[i] == servo04SP[i + 1]) {
}
if (servo04SP[i] > servo04SP[i + 1]) {
    for ( int j = servo04SP[i]; j >= servo04SP[i + 1]; j--) {
        servo04.write(j);
        delay( speedDelay);
    }
}

```

```

}
if (servo04SP[i] < servo04SP[i + 1]) {
    for ( int j = servo04SP[i]; j <= servo04SP[i + 1]; j++) {
        servo04.write(j);
        delay( speedDelay);
    }
}

// Servo 5
if (servo05SP[i] == servo05SP[i + 1]) {
}
if (servo05SP[i] > servo05SP[i + 1]) {
    for ( int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {
        servo05.write(j);
        delay( speedDelay);
    }
}
if (servo05SP[i] < servo05SP[i + 1]) {
    for ( int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {
        servo05.write(j);
        delay( speedDelay);
    }
}

// Servo 6
if (servo06SP[i] == servo06SP[i + 1]) {
}
if (servo06SP[i] > servo06SP[i + 1]) {
    for ( int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {
        servo06.write(j);
        delay( speedDelay);
    }
}
if (servo06SP[i] < servo06SP[i + 1]) {
    for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {
        servo06.write(j);
        delay( speedDelay);
    }
}
}

```

```
}  
}  
}
```

# Sources

Cinématique des robots série :

<https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/>

<https://poppy.discourse.group/t/presentation-dun-travail-de-cinematique-2d-avec-le-robot-poppy-ergo-jr-1ere-s/2611>

<https://poppy.discourse.group/t/algorithme-de-cinematique-inverse-applique-a-poppy-inverse-kinematic-algorithm-applied-to-poppy/1582>

<https://learn.ros4.pro/fr/theory/>

<https://files.ros4.pro/theorie.pdf>

<https://www.youtube.com/watch?v=vKD20BTkXhk>

Robot série à parallélogramme :

<https://www.youtube.com/watch?v=GnrRk9mWv7A>

<https://www.youtube.com/watch?v=xQkPjDEbFoU>

[https://www.youtube.com/watch?v=Z7HWoh\\_MR1s](https://www.youtube.com/watch?v=Z7HWoh_MR1s)

<https://drive.google.com/drive/folders/1zaOw2QcZatylyqQoVnYhIK7iysyCOY2d>

<https://www.youtube.com/watch?v=Q9JOKQaIR1w>

Un peu complexe :

[https://eduscol.education.fr/sti/ressources\\_pedagogiques/analyse-et-performance-cinematique-dun-robot-bi-article#fichiers-liens](https://eduscol.education.fr/sti/ressources_pedagogiques/analyse-et-performance-cinematique-dun-robot-bi-article#fichiers-liens)

<https://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/pedagogiques/14480/14480-1-robot-bi-articule-doc-ressource.pdf>

---

Revision #10

Created 16 January 2025 15:44:13 by Quentin Petrazoller

Updated 11 February 2025 11:37:12 by Quentin Petrazoller