

Sommaire

1. Introduction
2. Répartition du travail
3. Mécanique
 1. Réflexion
 2. Conception
4. Électricité
 1. Recherches
 2. Programme
5. Améliorations possibles
 1. Matérielles
 2. Logicielles
6. Annexe

Introduction

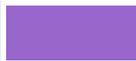
Cette année, en Sciences de l'Ingénieur, notre projet personnel consistait à créer un système permettant de contrôler une webcam selon trois axes : rotation de la webcam selon deux axes et déplacement vertical de l'ensemble du système. Plus tard, nous avons dû nous soumettre à une nouvelle contrainte, celle de devoir gérer une rotation d'au moins 180° sur les deux premiers axes. Le tout devait être gérable par l'intermédiaire d'un ordinateur, grâce à différentes touches du clavier.

Individuellement, j'ai travaillé majoritairement sur la partie électricité, bien qu'ayant contribué à plusieurs étapes de la partie mécanique.

Remerciements :

- A l'équipe éducative de Sciences de l'Ingénieur du Lycée St-Joseph Dijon
- A la communauté Arduino

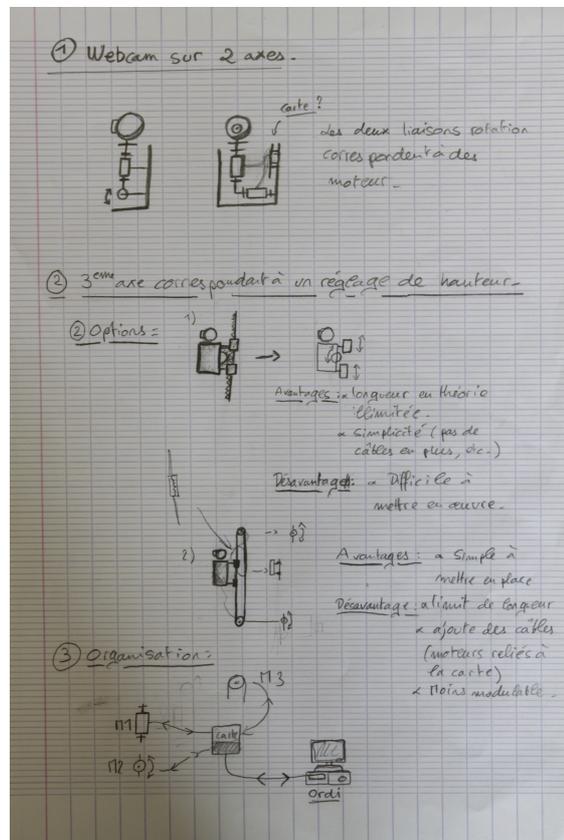
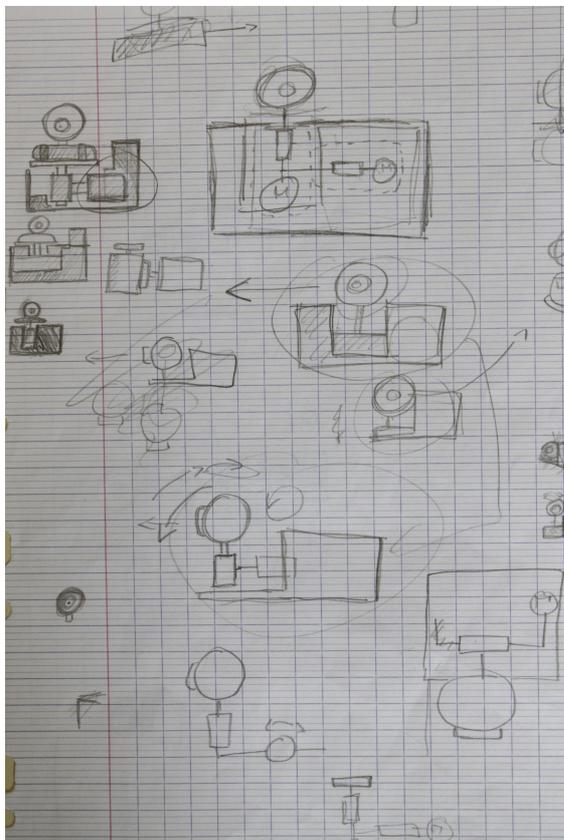
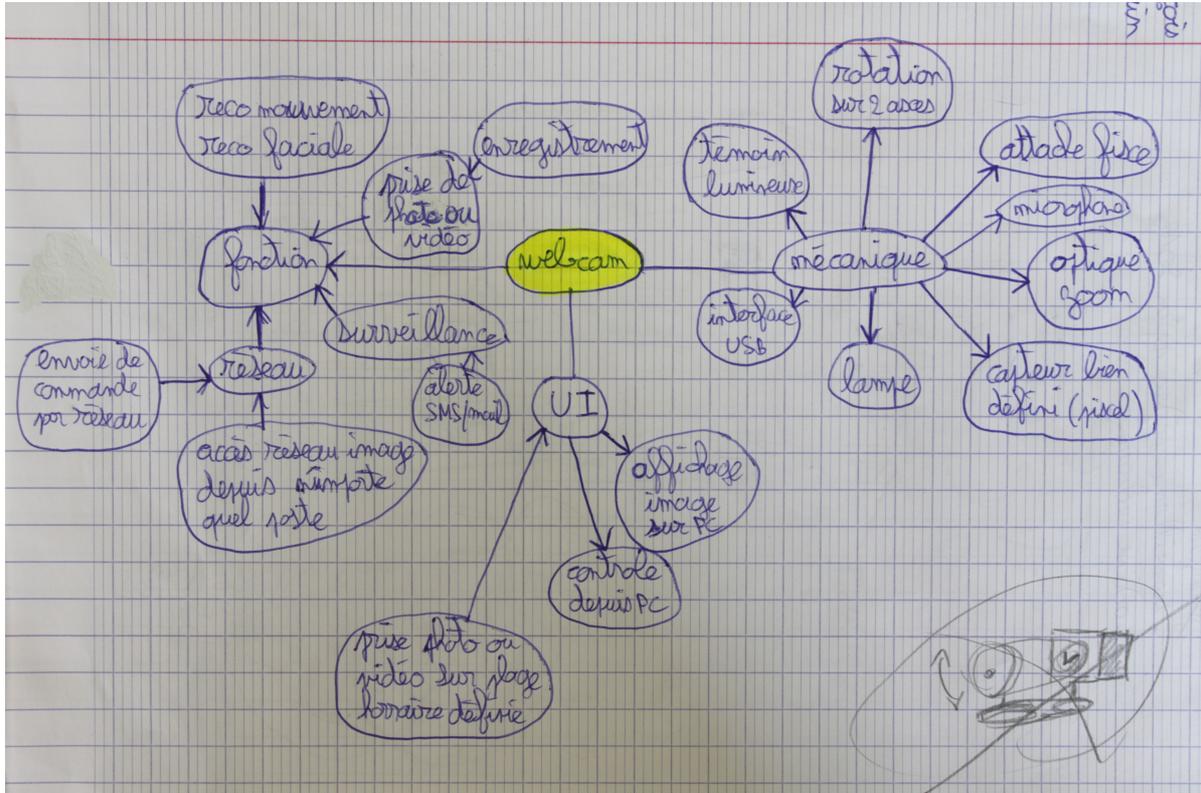
Répartition du travail

Étude de la chaîne cinématique	
Propositions de solutions de déplacement	
Recherches sur l'USB et le contrôle clavier	
Programmation de l'interface de contrôle	

- L'étude de la chaîne cinématique consistait à tous réfléchir sur le projet dans sa globalité, ce que nous allions devoir mettre en place sur le plan mécanique...
- Les propositions de déplacement sont les différentes solutions que j'ai pu imaginer pour le système, concernant les axes de rotation. C'est un travail individuel, certes, mais chaque personne du groupe a proposé au moins une solution.
- Les recherches sur l'USB et le contrôle du clavier ont consisté à savoir comment envoyer des informations provenant d'événements claviers à travers les différents ports de l'ordinateur pour les transmettre au micro-contrôleur.
- La programmation m'a amené à écrire l'interface de notre système, permettant de contrôler tous nos moteurs grâce aux touches du clavier.

Mécanique

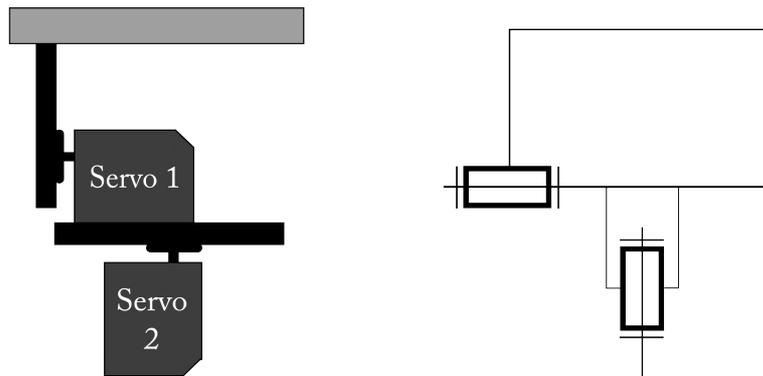
Réflexion



Nous avons choisis d'utiliser des servomoteurs pour les axes de rotations : en effet, ils sont compacts, économes et possèdent de très bonnes performances en accord avec nos attentes. Nous sommes donc partis sur des servomoteurs simples avant que l'on nous impose l'usage de servomoteurs à rotation continus, légèrement différents.

Le modèle final choisi pour les deux axes de rotation est le suivant :

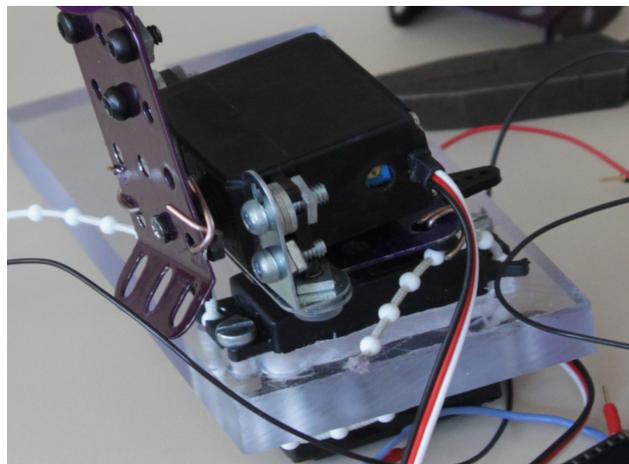
PPE - Webcam
Luca Mailhol



Conception



Nous avons fait le choix de construire notre bâti entièrement en mécano, non par nostalgie mais surtout car il s'agissait de la solution la plus simple et la plus rapide à mettre en œuvre !



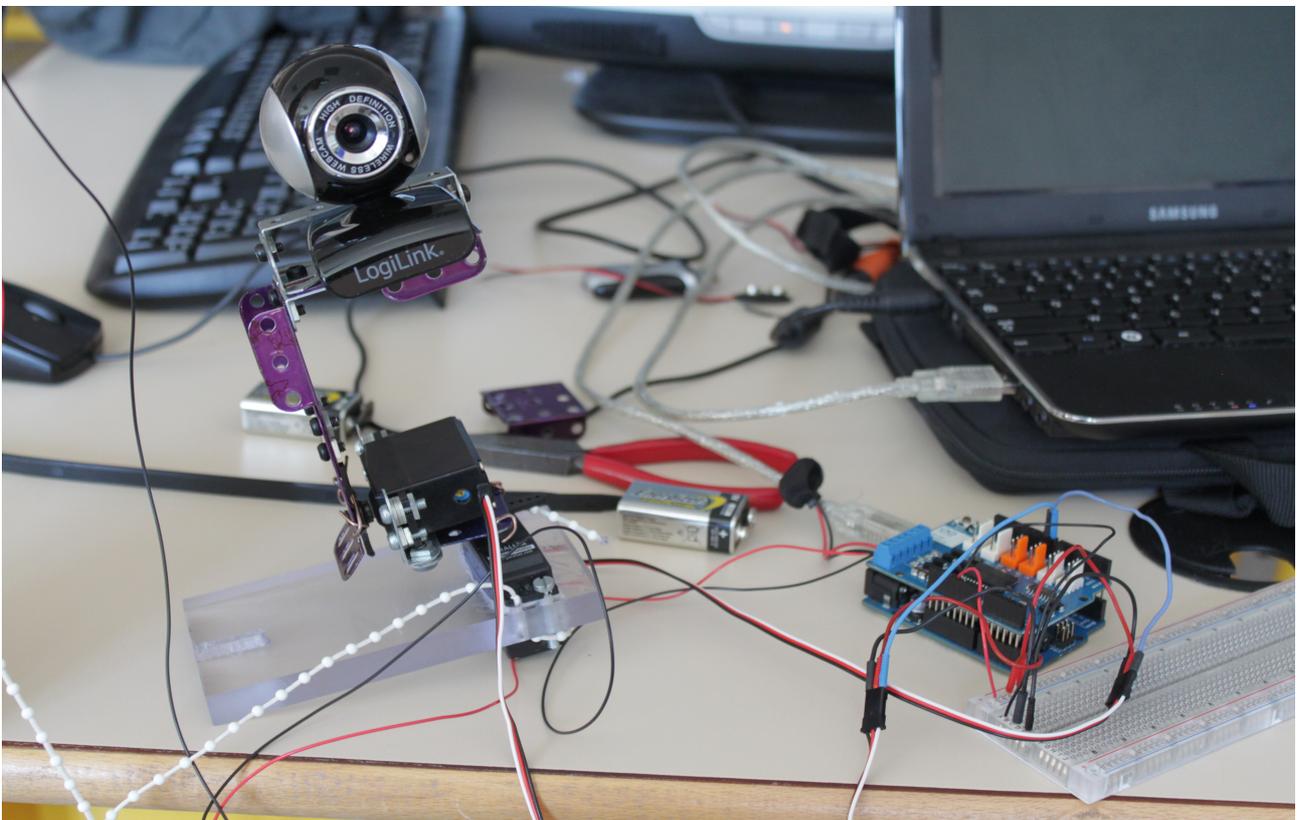
Électricité

Recherches

François et moi nous sommes penchés sur les événements clavier et leur transmission via un port USB. Nous avons alors trouvé la librairie C LibUSB. Pour autant, l'usage de cette librairie allait nous obliger à l'assimiler tout d'abord, ce qui s'annonçait assez long.

Concernant la carte, nous avons choisi d'utiliser une carte Arduino. Nous l'avons demandé car elle correspond parfaitement à nos attentes en terme matériel et il est possible de la coder directement en langage C ou C++. Nous avons eu accès à cette carte un peu plus tard dans l'année. Ainsi, tandis que François continuait ses recherches concernant LibUSB, j'ai trouvé une solution directement liée à Arduino : Processing. Il s'agit d'un langage de programmation couplé à un IDE, permettant de dialoguer facilement avec la carte. J'ai donc du apprendre cela sur le tas, ce qui s'est avéré fort intéressant.

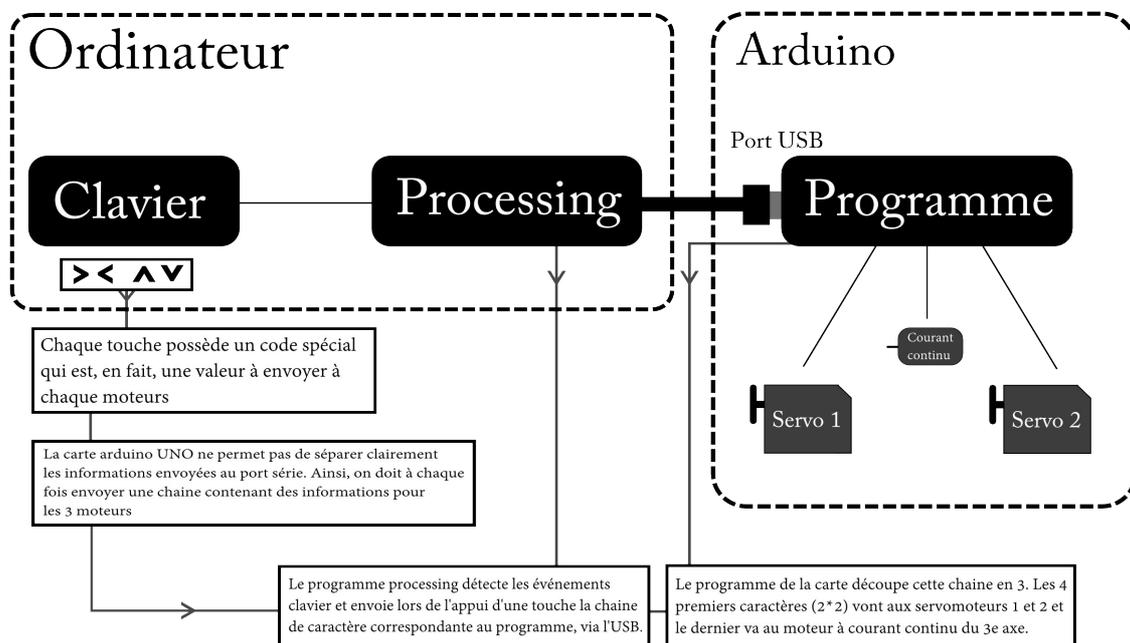
Nous avons couplé notre carte Arduino a un contrôleur possédant un pont en H, pour les branchements du moteur à courant continue du troisième axe.



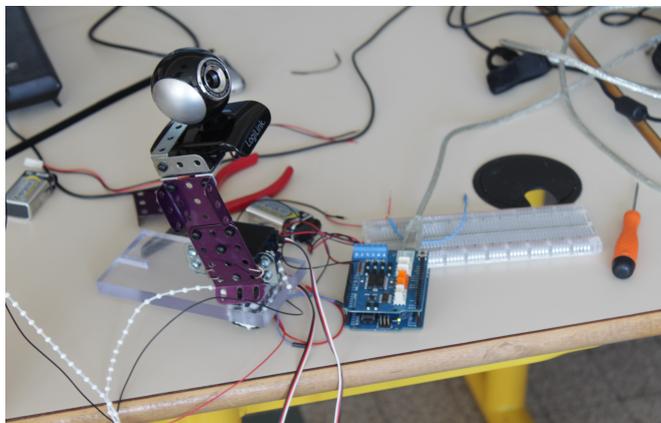
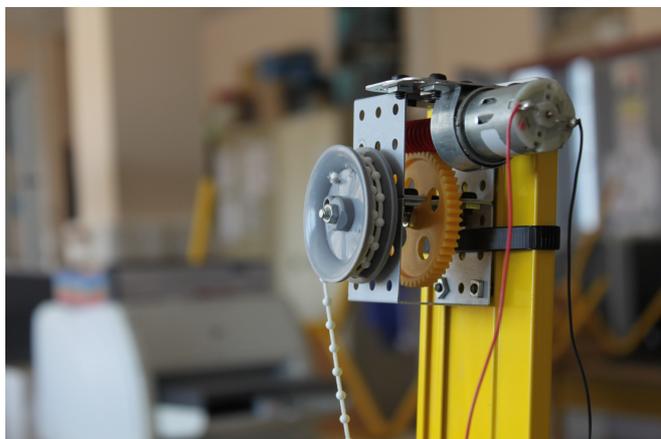
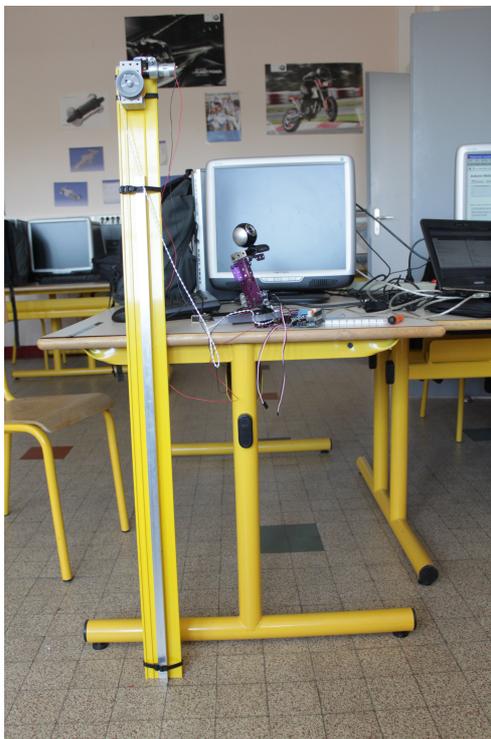
Programme

PPE - Webcam

Luca Mailhol



Le code et son détail se trouvent en annexe.



Améliorations possibles

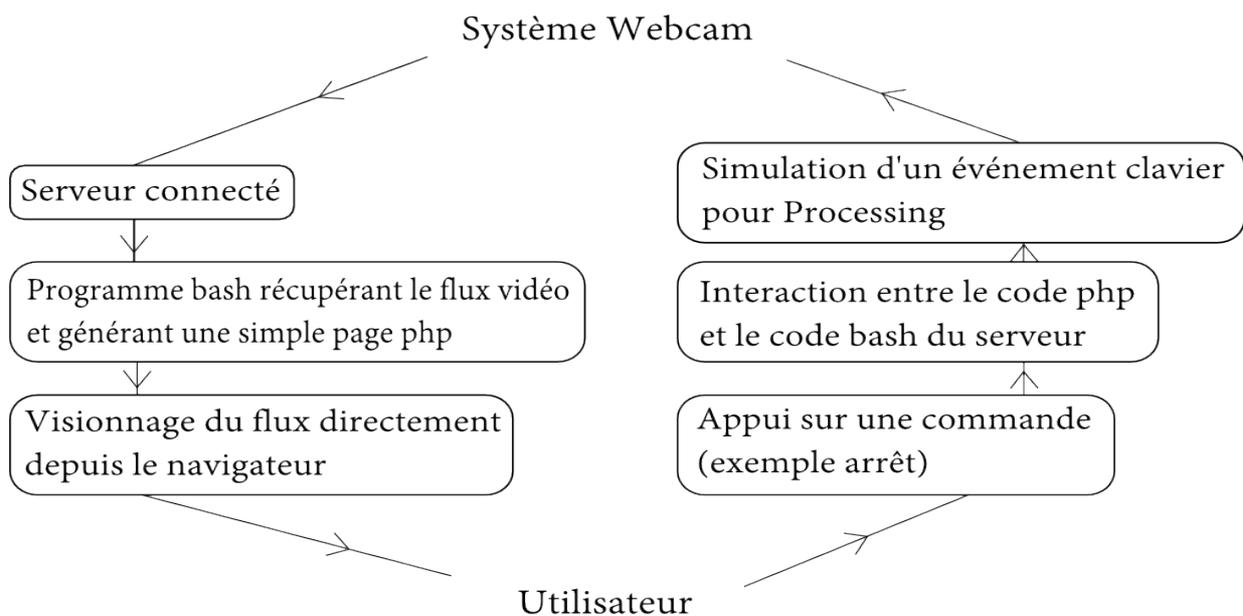
Voici une liste des améliorations possibles, qui seront détaillées à l'oral.

Matérielles

- Un retour à des servomoteurs normaux
- Un bâti plus compact

Logicielles

- Reconnaissance de mouvement
- Une capture d'écran de la webcam à intervalle régulier ou lors de la reconnaissance d'un mouvement
- Contrôle de la webcam via le réseau et partage du flux video sur internet pour la surveillance
- Alerte automatique en cas de mouvement, par mail par exemple
- Une amélioration du programme interface pour, par exemple, pouvoir gérer les différents mouvements en même temps



Annexe

Code Processing

```
import processing.serial.*;
Serial SerialPort;
int y = 0;
String valeur;

void setup(){
  size(200,200);
  textSize(80);
  println("Bienvenue");
  SerialPort = new Serial(this, Serial.list()[0], 115200);
}

void draw() { background(204); } //Fonction obligatoire pour gérer les evenements clavier

void keyPressed() {
  if (key == CODED) {
    if (keyCode == UP) { //"UP" et "DOWN" pour controler la rotation sur z
      valeur = "87902"; //on envoie une valeur de vitesse au servomoteur tandis qu'on stoppe le reste
      println(valeur);
      SerialPort.write(valeur);
      delay(100);
    }
    else if (keyCode == DOWN) {
      valeur = "93902";
      println(valeur);
      SerialPort.write(valeur);
      delay(100);
    }
    else if (keyCode == RIGHT) { //"RIGHT" et "LEFT" pour controler la rotation sur x
      valeur = "90922";
      println(valeur);
      SerialPort.write(valeur);
      delay(100);
    }
    else if (keyCode == LEFT) {
      valeur = "90882";
      println(valeur);
      SerialPort.write(valeur);
      delay(100);
    }
  }
}

if (key == 'p') { //"p" et "m" pour controler le deplacement sur y
  valeur = "90900";
  println(valeur);
  SerialPort.write(valeur);
  delay(100);
}

if (key == 'm') {
  valeur = "90901";
  println(valeur);
  SerialPort.write(valeur);
  delay(100);
}

if (key == 'a') { //"a" pour tout stopper
  valeur = "90902";
  println(valeur);
  SerialPort.write(valeur);
  delay(100);
}
}}
```

Code Arduino

```
#include <Servo.h>

#define SPEED 3 //broche de vitesse et de direction donnée par la documentation de notre controleur moteur
#define DIRECTION 12

Servo myservo; //deux servomoteurs
Servo myservo2;

int servoSpeed; //vitesse de rotation du servomoteur continue
int servoSpeed2; //vitesse de rotation du servomoteur continue
String readSerial, servo1, servo2, motor;

int motorSpeed; //vitesse du moteur
int motorDirection; //direction du moteur

void setup() {
  Serial.begin(115200);
  pinMode(8, OUTPUT); //broche en sortie servo1
  pinMode(9, OUTPUT); //broche en sortie servo2

  pinMode(DIRECTION, OUTPUT); //broche direction
  pinMode(SPEED, OUTPUT); //broche vitesse

  motorSpeed = 125; //vitesse du moteur
}

void loop() {
  while (Serial.available()) {
    delay(1);
    if (Serial.available() > 0) {
      char c = Serial.read(); //recuperation de la valeur envoyee au port serie
      readSerial += c; //rendre la chaine lisible
    }
  }
  if (readSerial.length() > 0) {
    Serial.println("Chaine : ");
    Serial.println(readSerial); //xxxxxxx

    servo1 = readSerial.substring(0, 2); //recuperation des deux premiers caracteres (servo1)
    servo2 = readSerial.substring(2, 4); //recuperation des deux suivants (servo2)
    motor = readSerial.substring(4, 5); //recuperation du dernier (moteur a courant continu)

    Serial.println("Servomoteur 1 : "); //affichage dans la console pour verifier visuellement
    Serial.println(servo1);
    Serial.println("Servomoteur 2 : ");
    Serial.println(servo2);
    Serial.println("Motor Courant Continu : ");
    Serial.println(motor);

    //conversion en int de chaques chaines (possibilite d'utiliser toInt(), aussi, manifestement)

    char carray1[5];
    servo1.toCharArray(carray1, sizeof(carray1));
    servoSpeed = atoi(carray1);

    char carray2[5];
    servo2.toCharArray(carray2, sizeof(carray2));
    servoSpeed2 = atoi(carray2);

    char carray3[5];
    motor.toCharArray(carray3, sizeof(carray3));
    motorDirection = atoi(carray3);
```

```
myservo.attach(8); //on attache les servomoteurs
myservo2.attach(9);
myservo.write(servoSpeed); //on envoie la valeur de vitesse
myservo2.write(servoSpeed2);

analogWrite(SPEED, motorSpeed); //controle du moteur a courant continu

if(motorDirection == 0) { //si la direction est de 0 on fait tourner dans un sens, sinon dans l'autre
  digitalWrite(DIRECTION,HIGH);
} else if(motorDirection == 1) {
  digitalWrite(DIRECTION,LOW);
} else if(motorDirection == 2) {
  analogWrite(SPEED, 0);
}

//remise a zero de la valeur de recuperation
readSerial = "";
}
}
```

L'ensemble du code – et ses éventuelles évolutions futures – est disponible sur un dépôt Github, à l'adresse suivante : <https://github.com/Radek411/webcam-arduino>