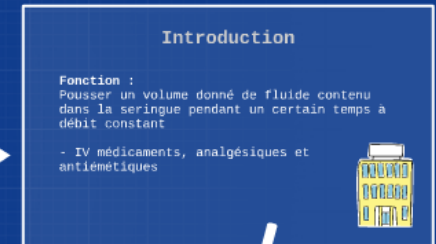
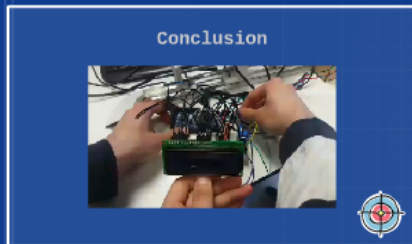


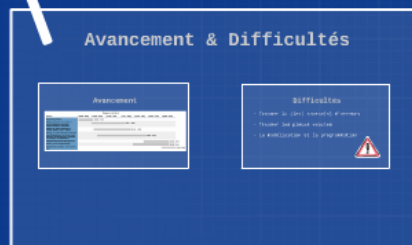
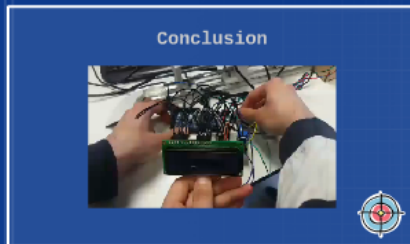
ARE : POUSSE-SERINGUE

BAKKAR, CLEMENTE, GABOLAEV,
KOPENEN & WOO



ARE : POUSSE-SERINGUE

BAKKAR, CLEMENTE, GABOLAEV,
KOPENEN & WOO



Sommaire

Introduction

Répartition des tâches

Avancement & difficultés

Conclusion

Introduction

Fonction :

Pousser un volume donné de fluide contenu dans la seringue pendant un certain temps à débit constant

- IV médicaments, analgésiques et antiémétiques



Répartition des tâches

Communication & Coordination

- Meneur
- Organisation des créneaux de travail
- Documentation
- Gestion du budget
- Surveillance



Aspect Physique

Débit massique : mg/ml

Débit volumique : ml/h

Modifié à l'aide du pavé numérique qui entraîne une fonction modifiant la pression dans le seringue.

Incertitudes :

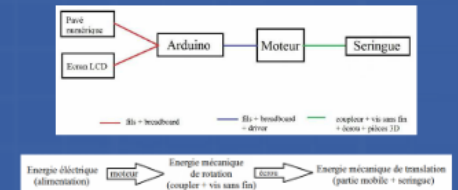
- distances
- volumes



Programmation



Aspect Mécanique



Modélisation 3D

- Imprimer les supports en 3D
- Supports extérieurs en bois
- Connexions des mécanismes

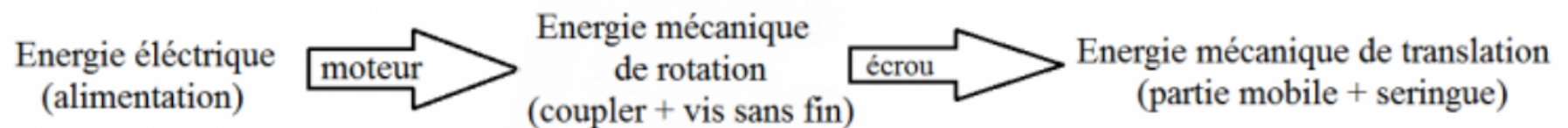
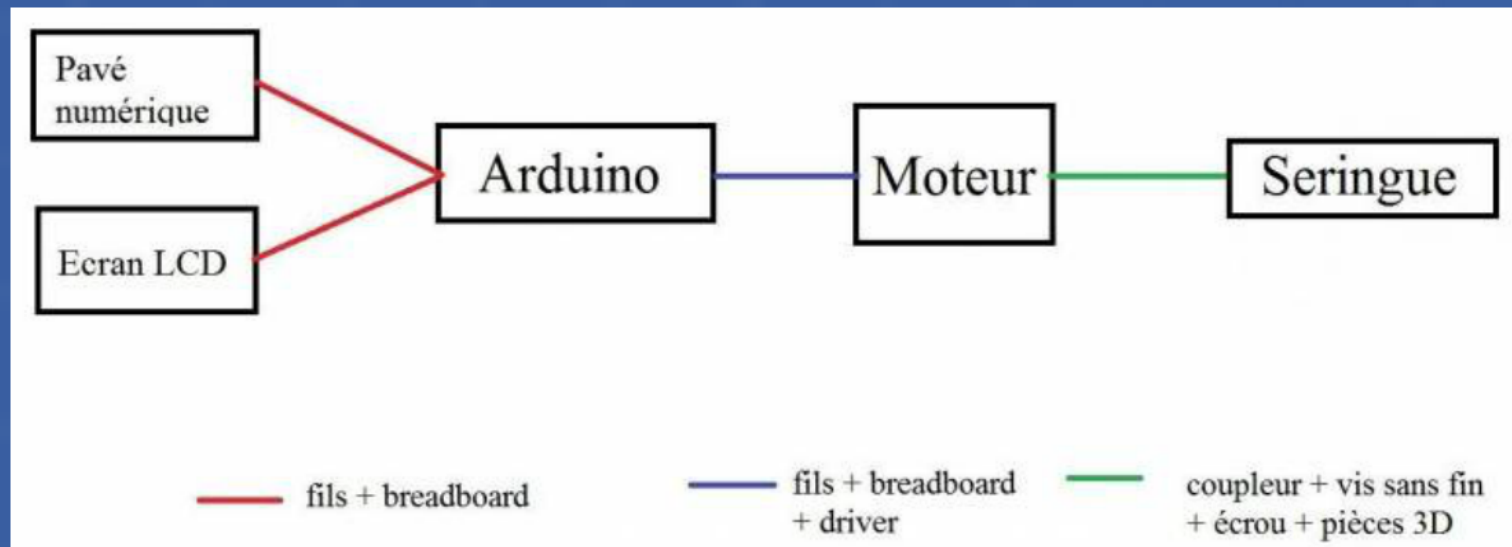


Communication & Coordination

- Meneur
- Organisation des créneaux de travail
- Documentation
- Gestion du budget
- Surveillance



Aspect Mécanique



Calcul de pas

Vis :

$$1\text{cm} = 8\text{tours}$$

$$1\text{tour} = 200\text{pas}$$

$$1\text{cm} = 1600\text{pas}$$

Seringue:

$$10\text{ml} = 2,6\text{cm}$$

$$1\text{ml} = 2,6/10\text{cm}$$

Finalemment :

$$\begin{aligned} 1\text{ml} &= 1600 * 2,6/10\text{pas} \\ &= 416\text{pas} \end{aligned}$$

Réel :

$$1\text{ml} = 425\text{pas}$$

Aspect Physique

Débit massique : mg/ml

Débit volumique : ml/h

Modifié à l'aide du pavé numérique qui entraîne une fonction modifiant la pression dans le seringue.

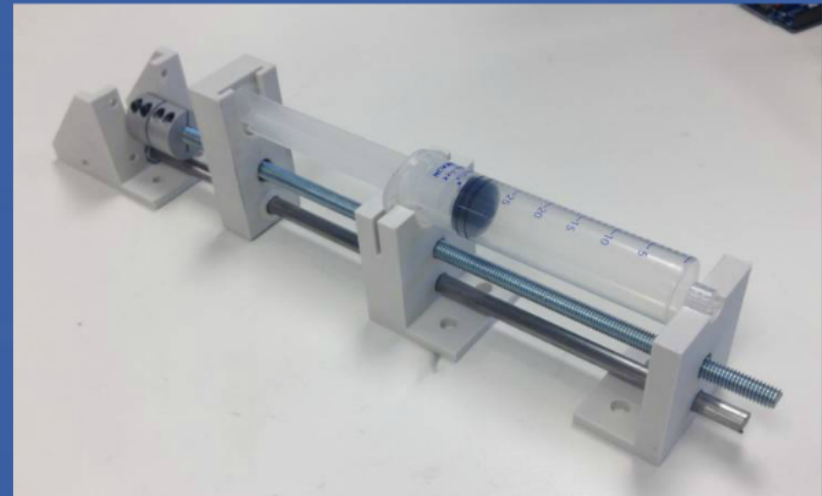
Incertitudes :

- distances
- volumes

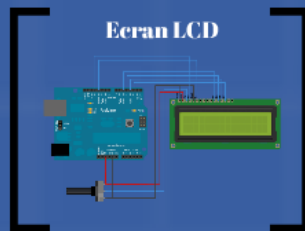
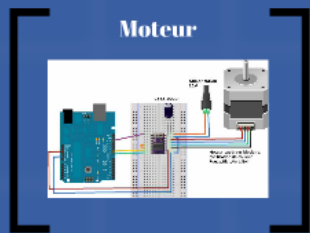


Modélisation 3D

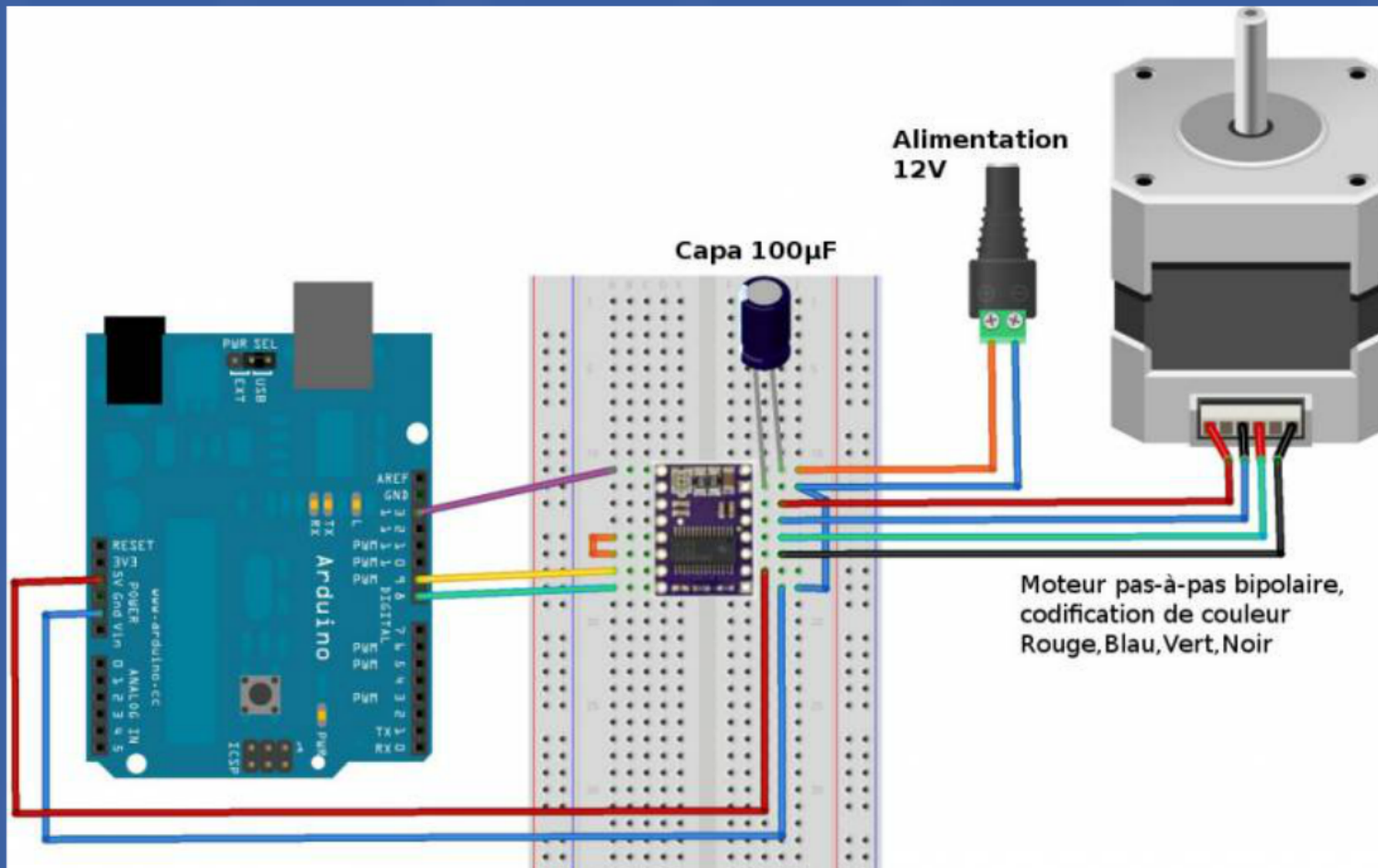
- Imprimer les supports en 3D
- Supports extérieurs en bois
- Connexions des mécanismes



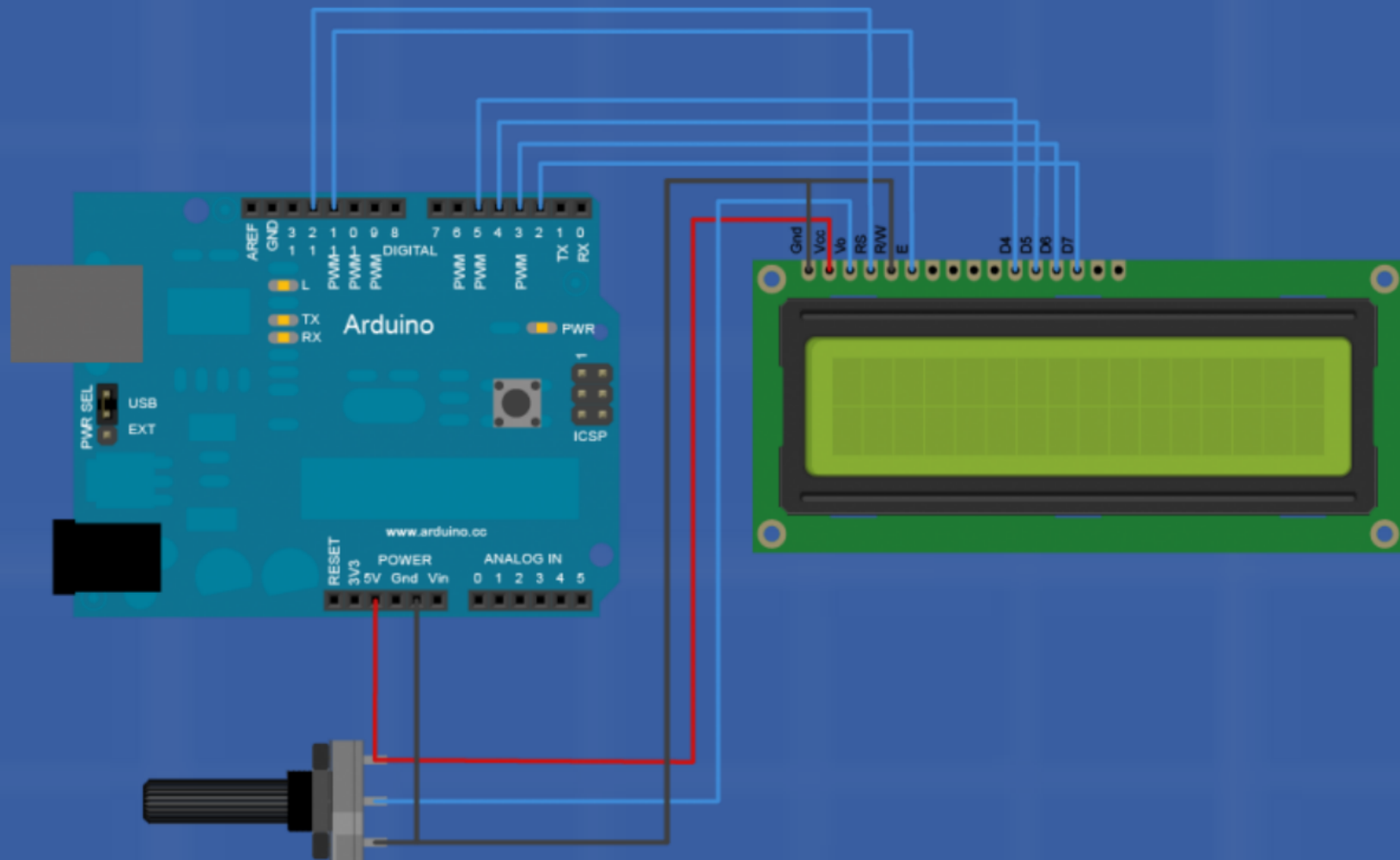
Programmation



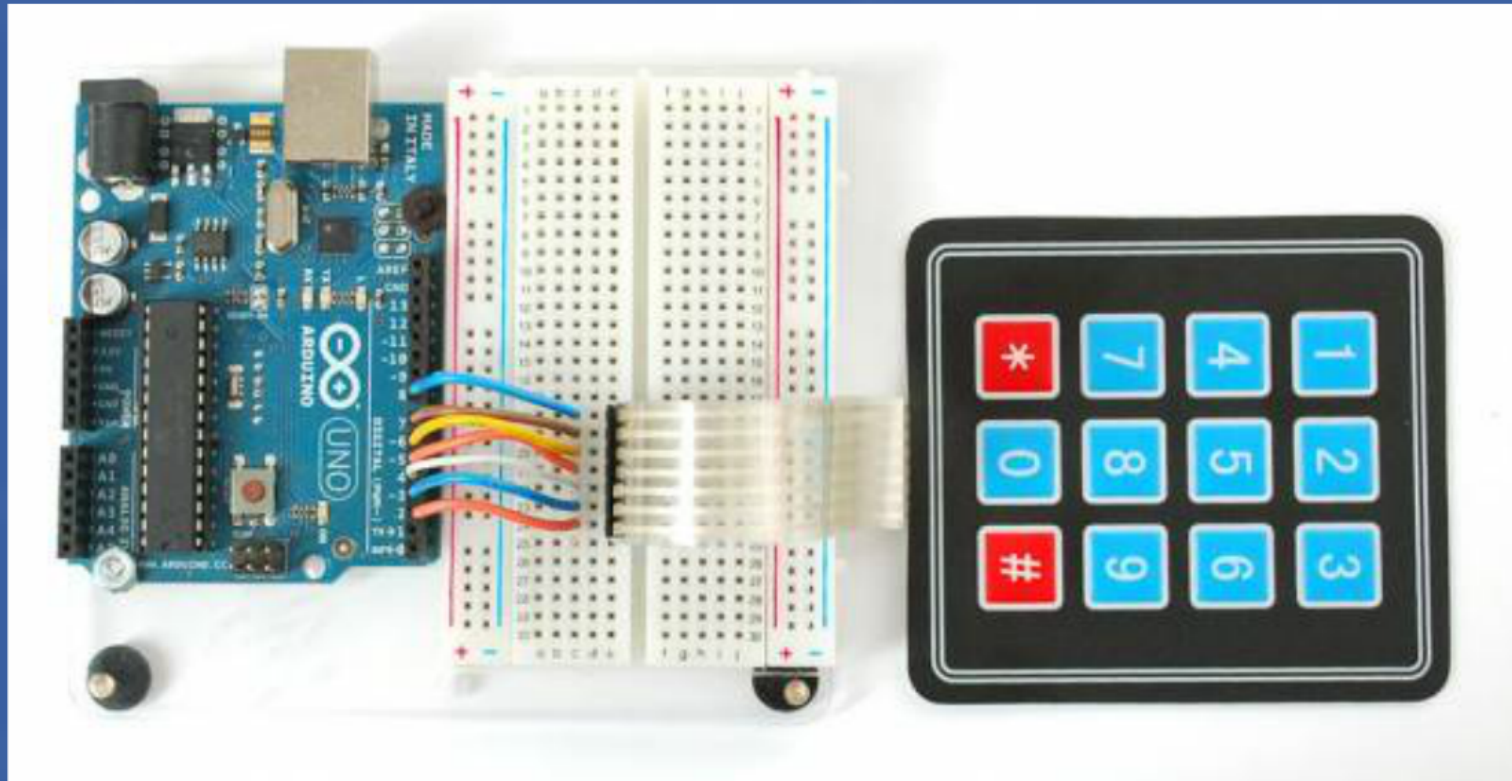
Moteur



Ecran LCD



Pavé numérique




```

#include <LiquidCrystal.h>
#include <Keypad.h>

int x;
#define BAUD (9600)

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','+'},
  {'4','5','6','-'},
  {'7','8','9','*'},
  {'X','0','=','/'},
};
byte rowPins[ROWS] = {53,51,49,47}; //connect to row pinouts
byte colPins[COLS] = {45,43,41,39}; //connect to column pinouts
Keypad myKeypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

//variables declaration
boolean valOnePresent = false;
boolean next = false;
boolean final = false;
boolean calcul termine = false;
String num1, num2;
int ans = 200;
int temps = 0;
int direction = 0;
char op;

```

```

void setup(){
  pinMode(6,OUTPUT); // Enable
  pinMode(5,OUTPUT); // Step
  pinMode(4,OUTPUT); // Dir
  digitalWrite(6, HIGH); // Set Enable low & eliminate the motor's initial vibration noise
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("Debit Volumique");
  delay(2000);
  lcd.clear(); //clears the LCD screen and positions the cursor in the upper-left corner.
  Serial.begin(BAUD);
}

void loop(){
  char key = myKeypad.getKey();
  if ((calcul termine == false) && key != NO_KEY &&
(key=='1'||key=='2'||key=='3'||key=='4'||key=='5'||key=='6'||key=='7'||key=='8'||key=='9'||key=='0')){
  if (valOnePresent != true){
    num1 = num1 + key;
    int numLength = num1.length();
    lcd.setCursor(15 - numLength, 0); //to adjust one whitespace for operator
    lcd.print(num1);
  }
  else {
    num2 = num2 + key;
    int numLength = num2.length();

```



```

}
if (calcul termine) {
  delay(2000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Direction");
  key = NO_KEY;
  while ( key == NO_KEY) {
    key = myKeypad.getKey();
  }
  if (key == '+') {
    direction = 1;
  }
  else if (key == '-') {

    direction = 2;
  }
}
if ((key != NO_KEY) && (key == 'X')){
  lcd.clear();
  valOnePresent = false;
  final = false;
  num1 = "";
  num2 = "";
  ans = 0;
  op = '';
}

```

```

if (direction == 1){
    lcd.clear();
    lcd.print("Direction +ve");
    digitalWrite(6,LOW); // Set Enable low
    digitalWrite(4,HIGH); // Set Dir high
    for(int x = 0; x < ans; x++) // Loop ans times
    {
        digitalWrite(5,HIGH); // Output high
        delay(temps);
        digitalWrite(5,LOW); // Output low
        delay(temps);
    }
    delay(1000); // pause one second
    direction = 0;
    calcul_termine = false;
    lcd.clear();
    num1 = "";
    num2 = "";
    ans = 0;
    op = '+';
    digitalWrite(6,HIGH); // Set Enable low
}

```

```

else if (direction == 2){
    lcd.clear();
    lcd.print("Direction -ve");
    digitalWrite(6,LOW); // Set Enable low
    digitalWrite(4,LOW); // Set Dir low
    for(int x = 0; x < ans; x++) // Loop ans times
    {
        digitalWrite(5,HIGH); // Output high
        delay(temps);
        digitalWrite(5,LOW); // Output low
        delay(temps);
    }
    delay(1000); // pause one second
    direction = 0;
    calcul_termine = false;
    lcd.clear();
    num1 = "";
    num2 = "";
    ans = 0;
    op = '-';

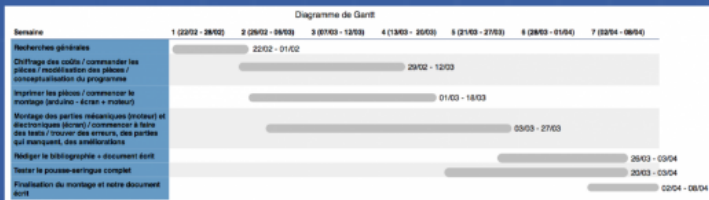
    digitalWrite(6,HIGH); // Set Enable low

}
}

```

Avancement & Difficultés

Avancement



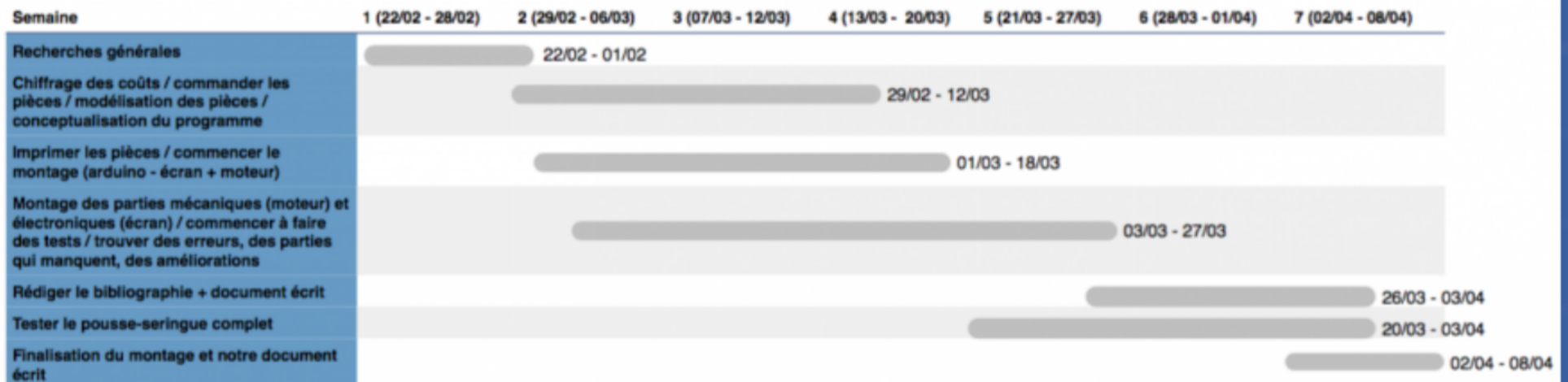
Difficultés

- Trouver la (les) source(s) d'erreurs
- Trouver les pièces exactes
- La modélisation et la programmation



Avancement

Diagramme de Gantt



Difficultés

- Trouver la (les) source(s) d'erreurs
- Trouver les pièces exactes
- La modélisation et la programmation



Conclusion

