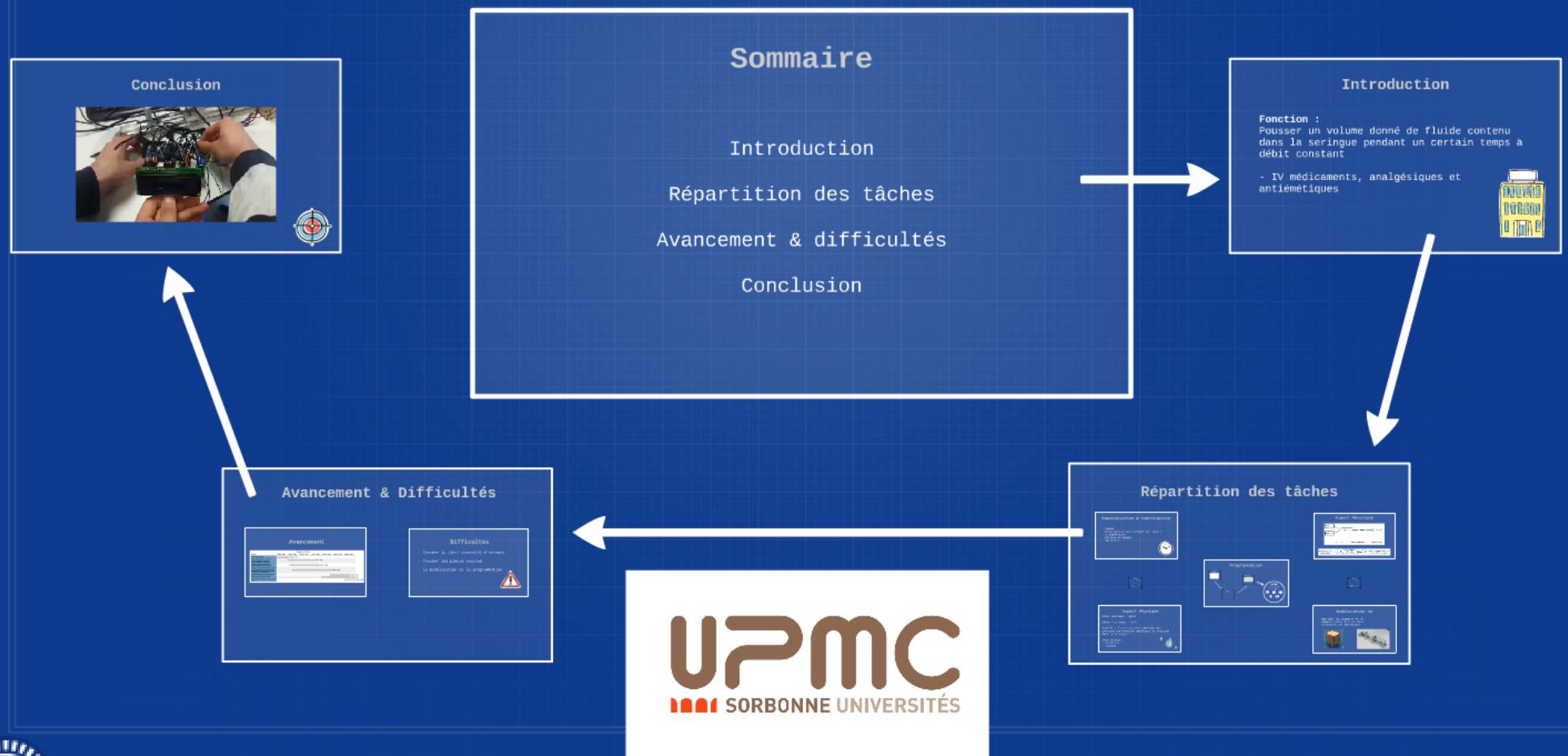


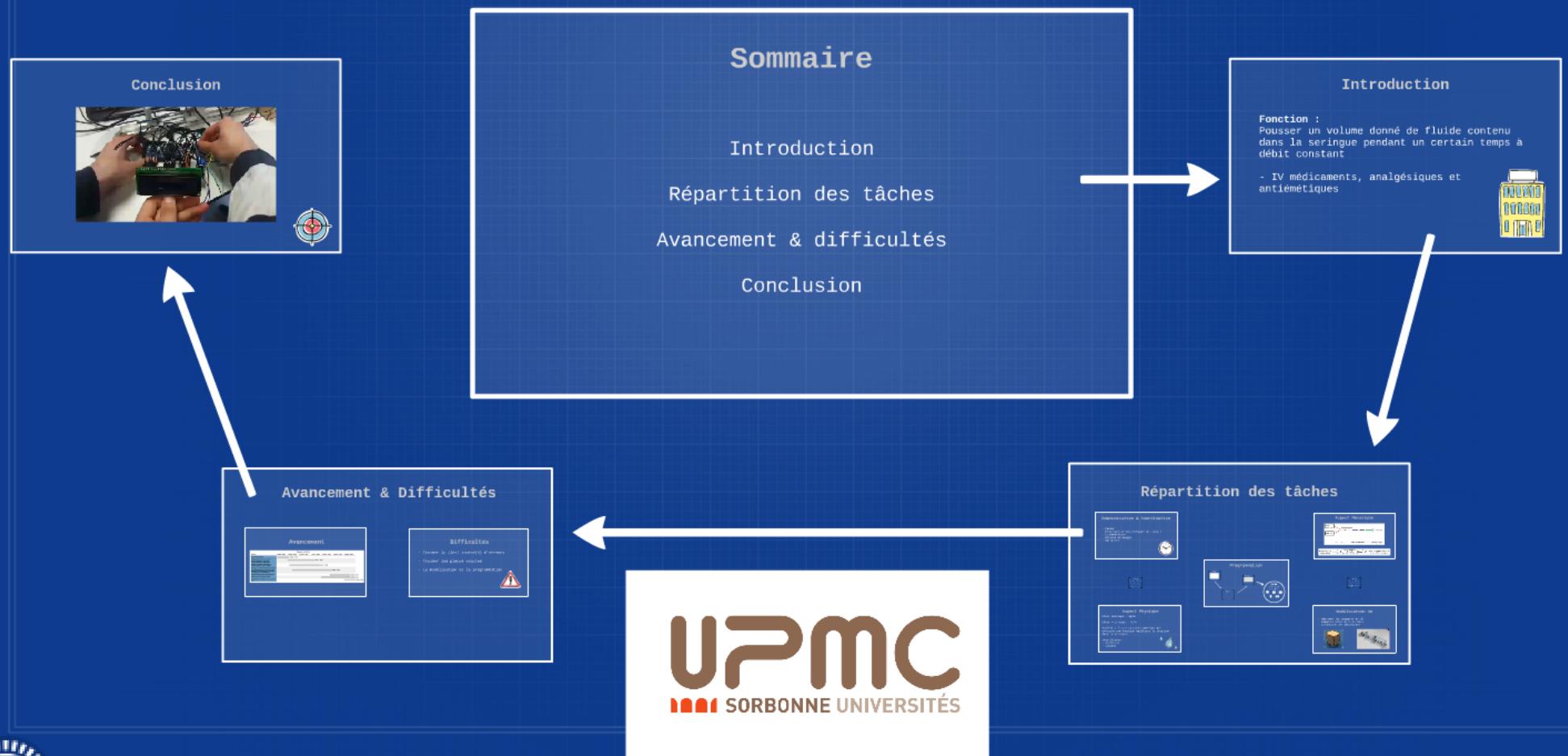
ARE : POUSSE-SERINGUE

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

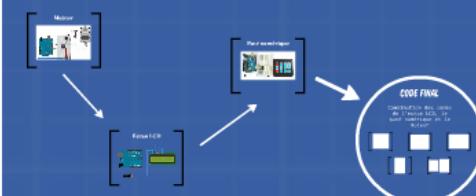
Débit volumique : ml/h

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

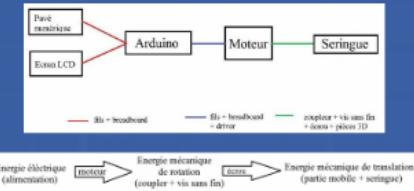
Incertitudes :
- distances
- volumes



Programmation



Aspect Mécanique



Modélisation 3D

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

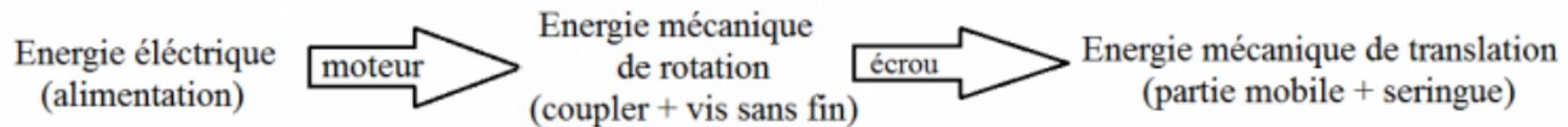
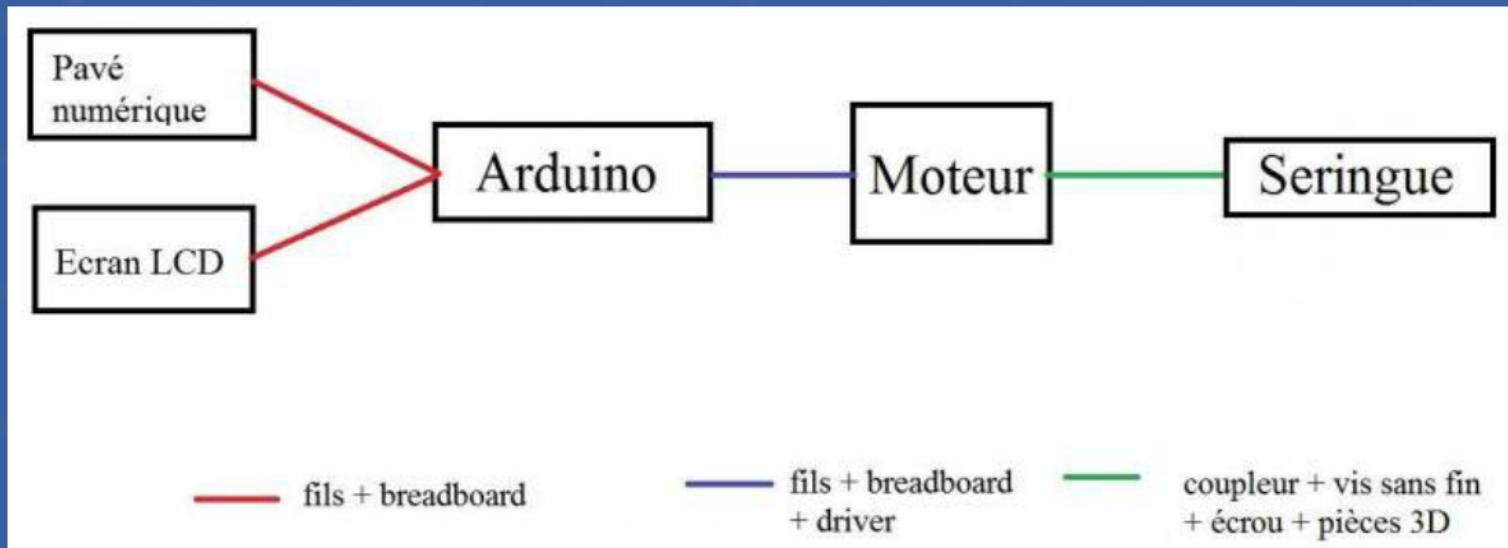


Communication & Coordination

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



Aspect Mécanique



Calcul de pas

Vis :

1cm = 8tours

1tour = 200pas

1cm = 1600pas

Seringue:

10ml = 2,6cm

1ml = 2,6/10cm

Finalement :

1ml = $1600 * 2,6 / 10$ pas
= 416pas

Réel :

1ml = 425pas

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



Calcul du débit

Débit :
volume/temps
 $= \text{vitesse} * \text{surface}$

Dimensions:
Vol. max - 30ml
diamètre - 24mm

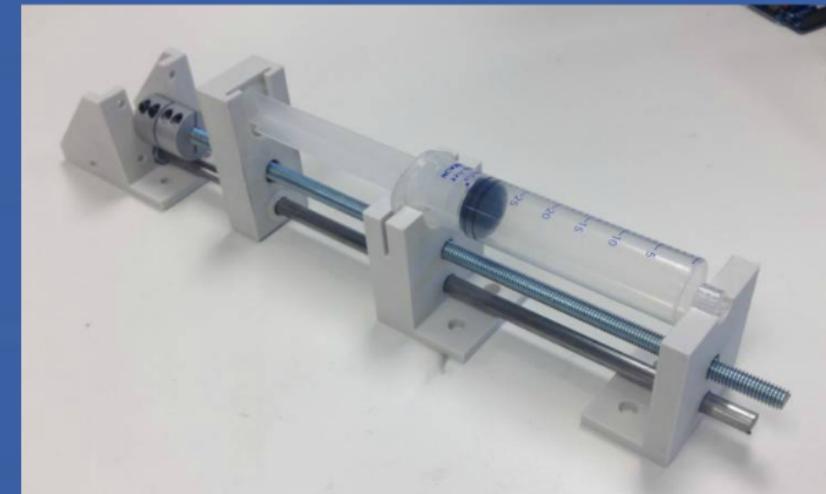
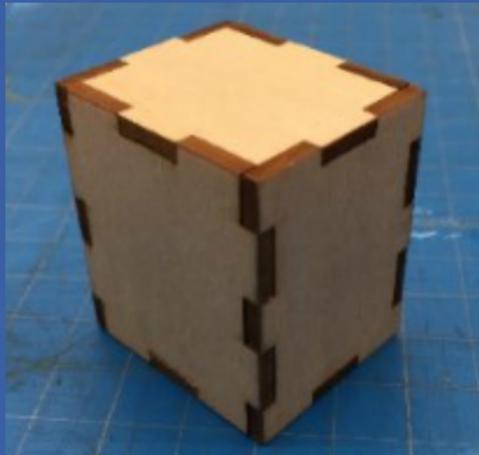
Vitesse = distance/temps
 $\Rightarrow 8\text{tours} = 1\text{cm} \Rightarrow 1\text{tour} = 1/8\text{cm}$

Embout (diamètre) $\rightarrow 2 \text{ mm} * V_f = V_i * 24 \text{ mm}$

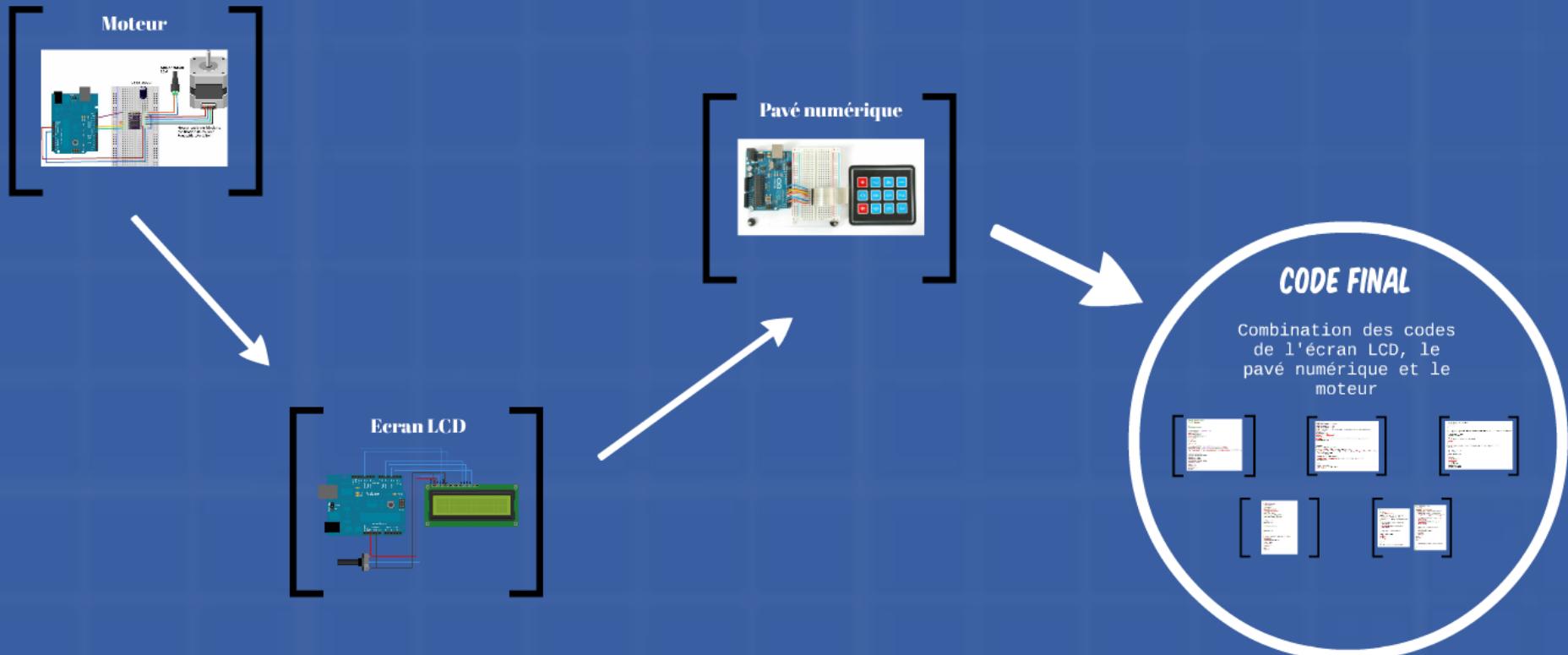
Incertitudes (distances) =
erreurs de mesure/lecture +/- 0,5 mm

Modélisation 3D

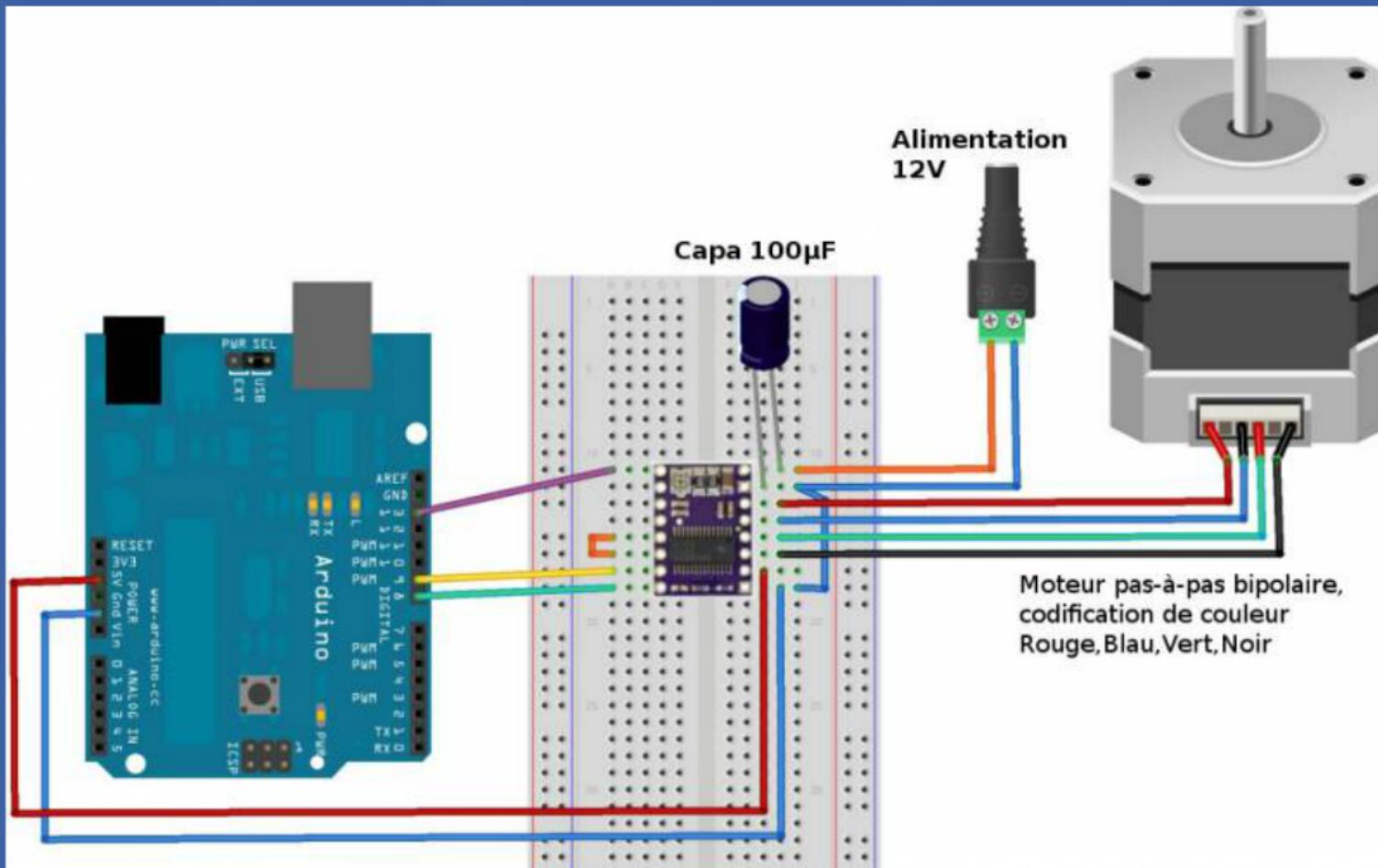
- Imprimer les supports en 3D
- Supports extérieurs en bois
- Connexions des mécanismes



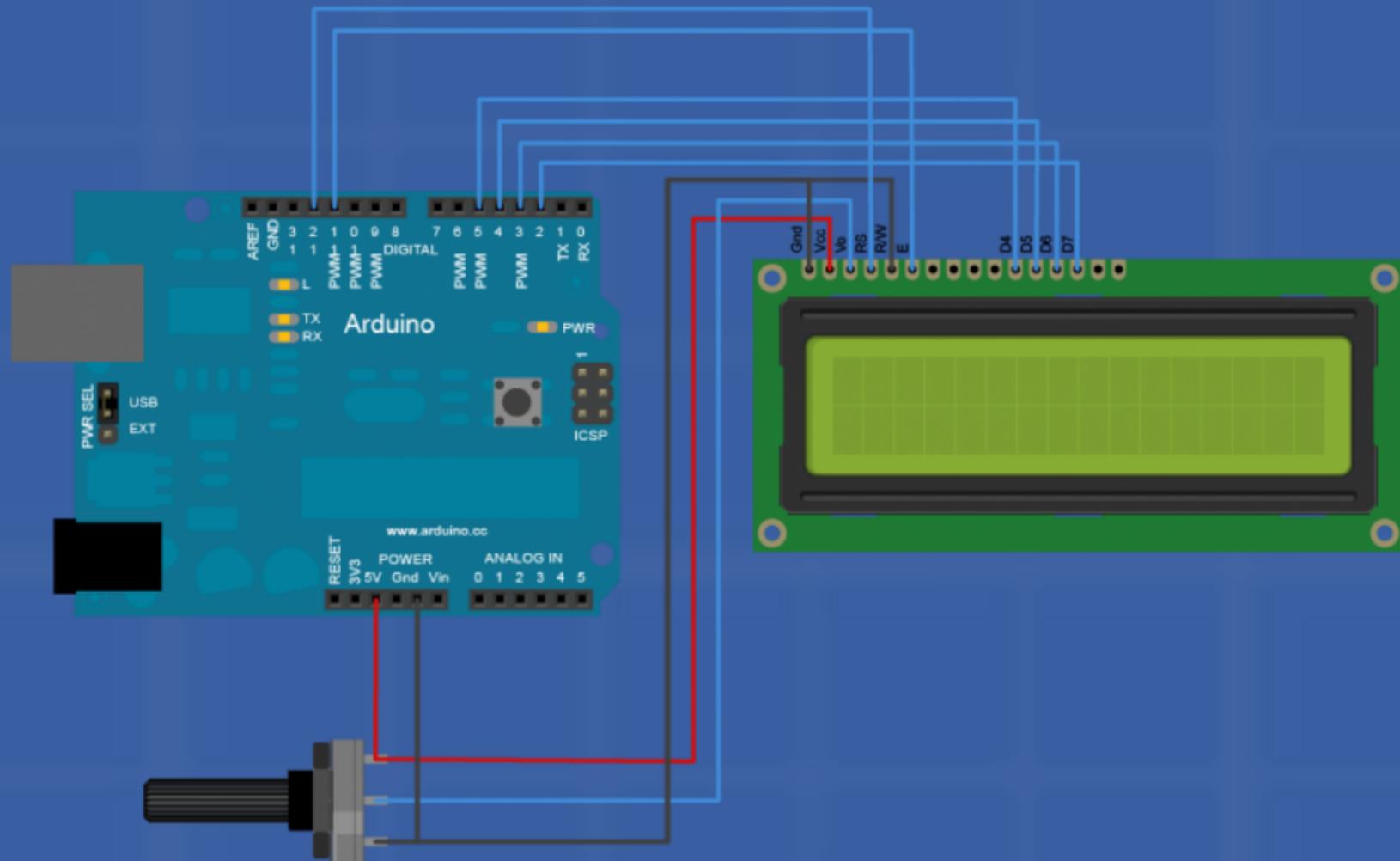
Programmation



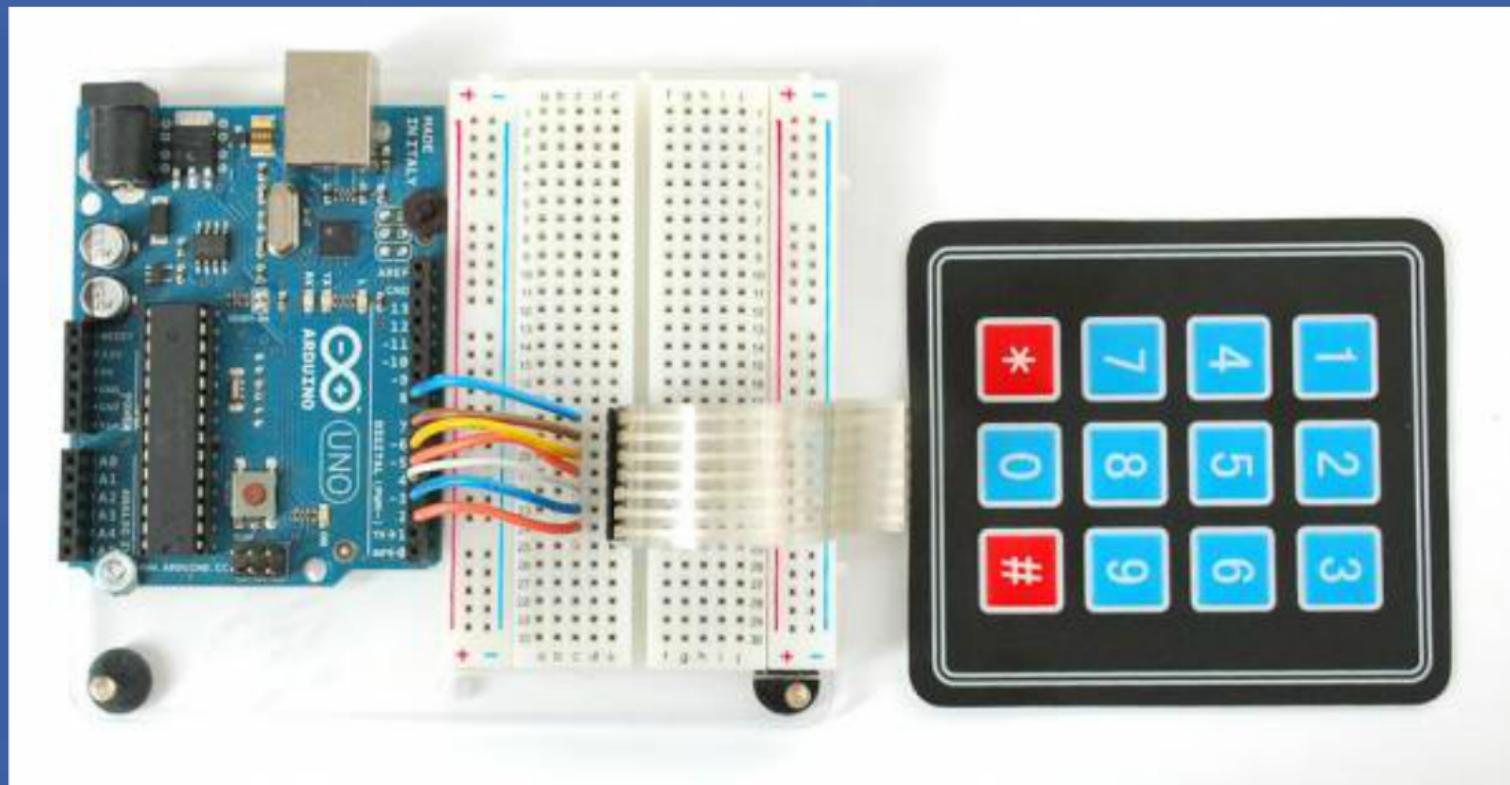
Moteur



Ecran LCD



Pavé numérique



CODE FINAL

Combination des codes
de l'écran LCD, le
pavé numérique et le
moteur

```
private void setup() {  
    // Set the background color to white:  
    background(255);  
  
    // Set the LCD pin numbers:  
    // These might need to be changed to match your LCD module!  
    int redPin = 2; // Pin 2  
    int greenPin = 3; // Pin 3  
    int bluePin = 4; // Pin 4  
  
    // Set the digital pins for the keypad:  
    int keypadRow1 = 5; // Pin 5  
    int keypadRow2 = 6; // Pin 6  
    int keypadRow3 = 7; // Pin 7  
    int keypadRow4 = 8; // Pin 8  
  
    int keypadCol1 = 9; // Pin 9  
    int keypadCol2 = 10; // Pin 10  
    int keypadCol3 = 11; // Pin 11  
    int keypadCol4 = 12; // Pin 12  
  
    // Set the motor pins:  
    int motorA = 13; // Pin 13  
    int motorB = 14; // Pin 14  
    int enable = 15; // Pin 15  
}
```

```
private void loop() {  
    // Set the background color to white:  
    background(255);  
  
    // Set the LCD pin numbers:  
    // These might need to be changed to match your LCD module!  
    int redPin = 2; // Pin 2  
    int greenPin = 3; // Pin 3  
    int bluePin = 4; // Pin 4  
  
    // Set the digital pins for the keypad:  
    int keypadRow1 = 5; // Pin 5  
    int keypadRow2 = 6; // Pin 6  
    int keypadRow3 = 7; // Pin 7  
    int keypadRow4 = 8; // Pin 8  
  
    int keypadCol1 = 9; // Pin 9  
    int keypadCol2 = 10; // Pin 10  
    int keypadCol3 = 11; // Pin 11  
    int keypadCol4 = 12; // Pin 12  
  
    // Set the motor pins:  
    int motorA = 13; // Pin 13  
    int motorB = 14; // Pin 14  
    int enable = 15; // Pin 15  
  
    // Set the keypad matrix:  
    int keypadMatrix[4][4] = {{  
        keypadRow1, keypadRow2, keypadRow3, keypadRow4},  
        {keypadCol1, keypadCol2, keypadCol3, keypadCol4}};  
  
    // Set the keypad labels:  
    String keypadLabels[16] = {"1", "2", "3", "A",  
        "4", "5", "6", "B",  
        "7", "8", "9", "C",  
        "*", "0", "#", "D"};  
  
    // Set the LCD text:  
    String lcdText = "LCD Test";  
  
    // Set the motor speed:  
    int motorSpeed = 100;  
  
    // Set the keypad state:  
    int keypadState = 0;  
  
    // Set the motor direction:  
    int motorDirection = 1;  
  
    // Set the keypad row:  
    int keypadRow = 0;  
  
    // Set the keypad column:  
    int keypadColumn = 0;  
  
    // Set the keypad row:  
    int keypadRow = 0;  
  
    // Set the keypad column:  
    int keypadColumn = 0;  
  
    // Set the keypad row:  
    int keypadRow = 0;  
  
    // Set the keypad column:  
    int keypadColumn = 0;  
  
    // Set the keypad row:  
    int keypadRow = 0;  
  
    // Set the keypad column:  
    int keypadColumn = 0;
```

```
private void setup() {  
    // Set the background color to white:  
    background(255);  
  
    // Set the LCD pin numbers:  
    // These might need to be changed to match your LCD module!  
    int redPin = 2; // Pin 2  
    int greenPin = 3; // Pin 3  
    int bluePin = 4; // Pin 4  
  
    // Set the digital pins for the keypad:  
    int keypadRow1 = 5; // Pin 5  
    int keypadRow2 = 6; // Pin 6  
    int keypadRow3 = 7; // Pin 7  
    int keypadRow4 = 8; // Pin 8  
  
    int keypadCol1 = 9; // Pin 9  
    int keypadCol2 = 10; // Pin 10  
    int keypadCol3 = 11; // Pin 11  
    int keypadCol4 = 12; // Pin 12  
  
    // Set the motor pins:  
    int motorA = 13; // Pin 13  
    int motorB = 14; // Pin 14  
    int enable = 15; // Pin 15  
}
```

```
private void loop() {  
    // Set the background color to white:  
    background(255);  
  
    // Set the LCD pin numbers:  
    // These might need to be changed to match your LCD module!  
    int redPin = 2; // Pin 2  
    int greenPin = 3; // Pin 3  
    int bluePin = 4; // Pin 4  
  
    // Set the digital pins for the keypad:  
    int keypadRow1 = 5; // Pin 5  
    int keypadRow2 = 6; // Pin 6  
    int keypadRow3 = 7; // Pin 7  
    int keypadRow4 = 8; // Pin 8  
  
    int keypadCol1 = 9; // Pin 9  
    int keypadCol2 = 10; // Pin 10  
    int keypadCol3 = 11; // Pin 11  
    int keypadCol4 = 12; // Pin 12  
  
    // Set the motor pins:  
    int motorA = 13; // Pin 13  
    int motorB = 14; // Pin 14  
    int enable = 15; // Pin 15  
  
    // Set the keypad matrix:  
    int keypadMatrix[4][4] = {{  
        keypadRow1, keypadRow2, keypadRow3, keypadRow4},  
        {keypadCol1, keypadCol2, keypadCol3, keypadCol4}};  
  
    // Set the keypad labels:  
    String keypadLabels[16] = {"1", "2", "3", "A",  
        "4", "5", "6", "B",  
        "7", "8", "9", "C",  
        "*", "0", "#", "D"};  
  
    // Set the LCD text:  
    String lcdText = "LCD Test";  
  
    // Set the motor speed:  
    int motorSpeed = 100;  
  
    // Set the keypad state:  
    int keypadState = 0;  
  
    // Set the motor direction:  
    int motorDirection = 1;  
  
    // Set the keypad row:  
    int keypadRow = 0;  
  
    // Set the keypad column:  
    int keypadColumn = 0;  
  
    // Set the keypad row:  
    int keypadRow = 0;  
  
    // Set the keypad column:  
    int keypadColumn = 0;  
  
    // Set the keypad row:  
    int keypadRow = 0;  
  
    // Set the keypad column:  
    int keypadColumn = 0;
```

```
private void setup() {  
    // Set the background color to white:  
    background(255);  
  
    // Set the LCD pin numbers:  
    // These might need to be changed to match your LCD module!  
    int redPin = 2; // Pin 2  
    int greenPin = 3; // Pin 3  
    int bluePin = 4; // Pin 4  
  
    // Set the digital pins for the keypad:  
    int keypadRow1 = 5; // Pin 5  
    int keypadRow2 = 6; // Pin 6  
    int keypadRow3 = 7; // Pin 7  
    int keypadRow4 = 8; // Pin 8  
  
    int keypadCol1 = 9; // Pin 9  
    int keypadCol2 = 10; // Pin 10  
    int keypadCol3 = 11; // Pin 11  
    int keypadCol4 = 12; // Pin 12  
  
    // Set the motor pins:  
    int motorA = 13; // Pin 13  
    int motorB = 14; // Pin 14  
    int enable = 15; // Pin 15  
}
```

```

#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_terminé = 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_terminé == 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();
}
}
}

```

```

lcd.setCursor(15 - numLength, 1);
lcd.print(num2);
final = true;
}
}
else if ((calcul_terminé == false) && (valOnePresent == false) && (key != NO_KEY) && (key == '/') || key == '*'
key == '-' || key == '+')){
if (valOnePresent == false){
valOnePresent = true;
op = key;
lcd.setCursor(15,0); //operator on right corner
lcd.print(op);
}
}
if ((calcul_terminé == false) && (final == true) && (key != NO_KEY) && (key == '=')) {
if (op == '+'){
ans = num1.toInt()*425 ;
//1ml = 425steps
temp = num2.toInt();
}
lcd.clear();
lcd.setCursor(15,0);
lcd.autoscroll();
lcd.print(ans);
lcd.noAutoscroll();
calcul_terminé = true;
valOnePresent = false;
final = false;
}
}

```

```
}

if (calcul_terminé) {
    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(temp);
        digitalWrite(5,LOW); // Output low
        delay(temp);
    }
    delay(1000); // pause one second
    direction = 0;
    calcul_termine = false;
    lcd.clear();
    num1 = "";
    num2 = "";
    ans = 0;
    op = '';
    digitalWrite(6,HIGH); // Set Enable low
}

```

```

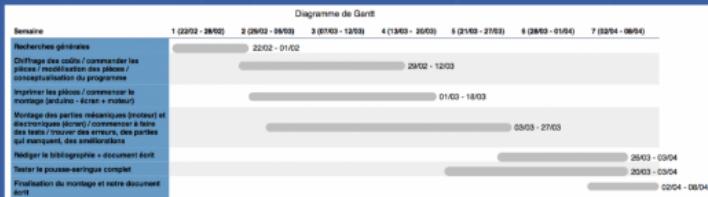
else if (direction == 2){
    lcd.clear();
    lcd.print("Direction -ve");
    digitalWrite(6,LOW); // Set Enable low
    digitalWrite(4,LOW); // Set Dir low
    for(int x = 0; x < ans; x++) // Loop ans times
    {
        digitalWrite(5,HIGH); // Output high
        delay(temp);
        digitalWrite(5,LOW); // Output low
        delay(temp);
    }
    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

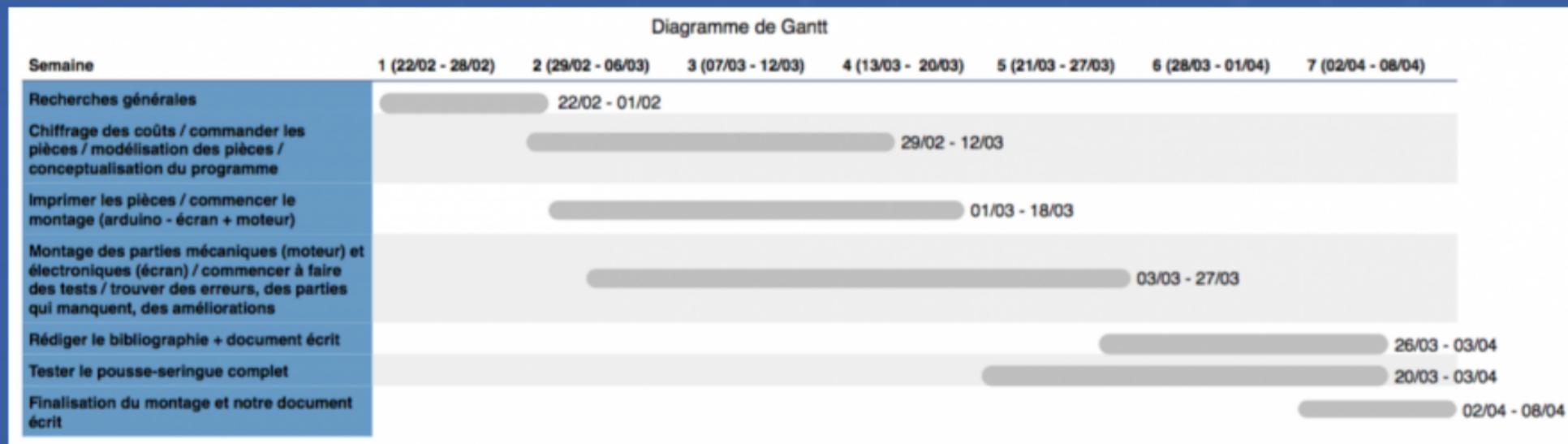


Difficultés

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



Avancement



Difficultés

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



Conclusion

