

Rayar Mathis rayarmathis@gmail.com

Bourguelle Alexis alexis@bourguelle.net

Helard Axel lexawolf03@gmail.com

Minier William william.minier2007@gmail.com

Pont Mobile



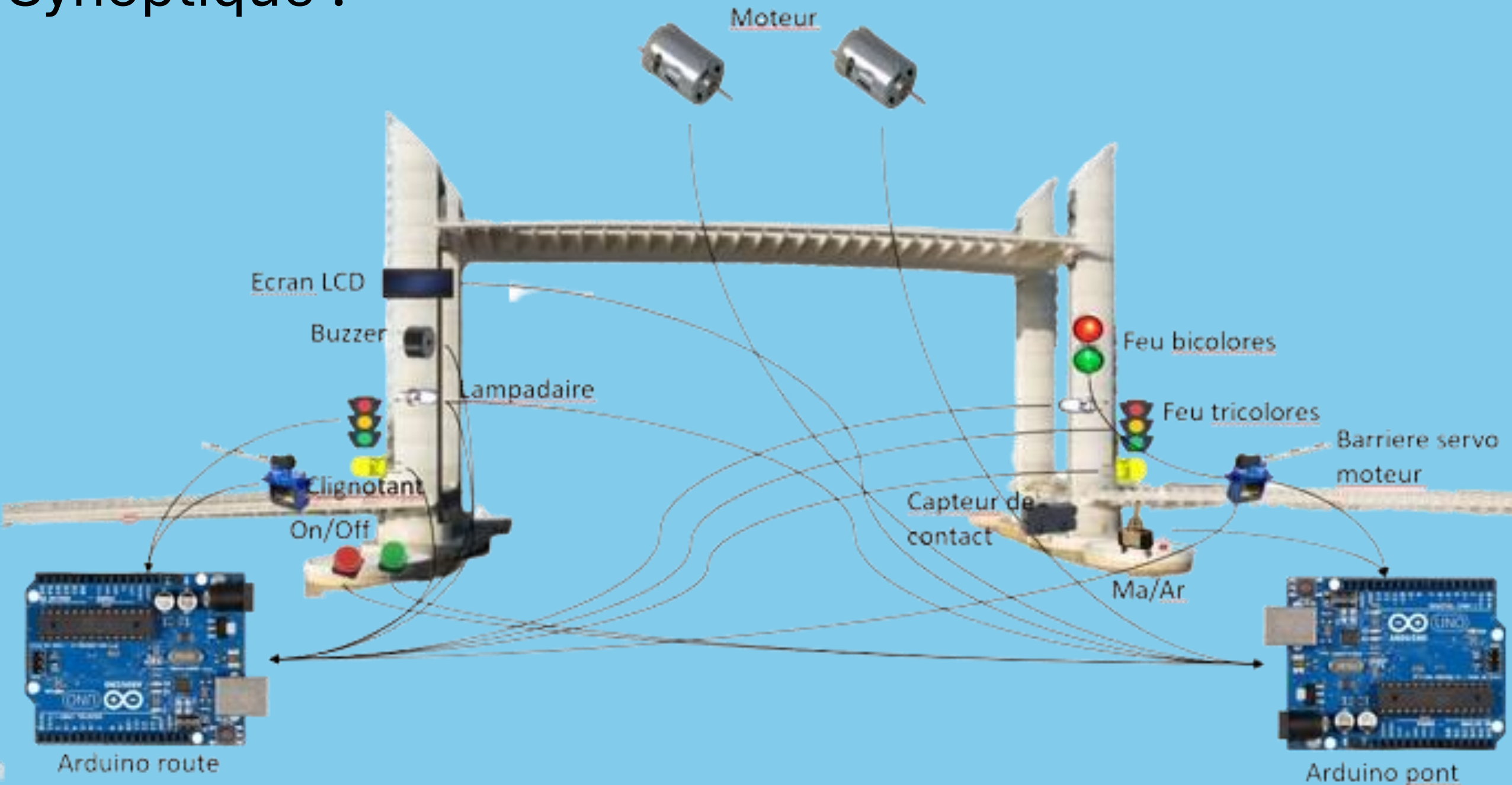
Problématique :

Nous envisageons de concevoir un pont mobile pour permettre le passage des bateaux entrant et sortant d'un port maritime. Lorsqu'aucun bateau n'est présent, le maintien de la circulation routière assure une fluidité dans le port et ses environs. Un poste de commande situé à proximité du pont permet d'ouvrir et de fermer le pont. Un opérateur contrôle et surveille le fonctionnement du pont lorsqu'un bateau souhaite passer. Selon la position du pont, des feux bicolores doivent informer les navigateurs de la possibilité de passage. Des feux tricolores doivent signaler aux automobilistes les interruptions et reprises de la circulation lorsque nécessaire. Des barrières de sécurité doivent bloquer la circulation automobile avant l'ouverture du pont et jusqu'à ce que celle-ci puisse reprendre. Un afficheur doit informer les automobilistes de la position du pont afin qu'ils puissent choisir un itinéraire alternatif si nécessaire. Des feux clignotants doivent alerter les usagers pendant toute la durée de fonctionnement du pont, et un signal sonore doit les prévenir lorsque le pont est en mouvement.

Répartition des tâches :

| | |
|---------|----------------------------|
| Mathis | → Gestion conception route |
| Alexis | → Gestion codage route |
| Axel | → Gestion codage pont |
| William | → Gestion conception pont |

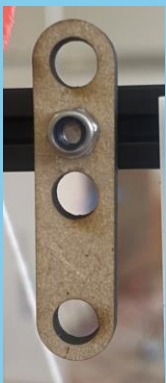
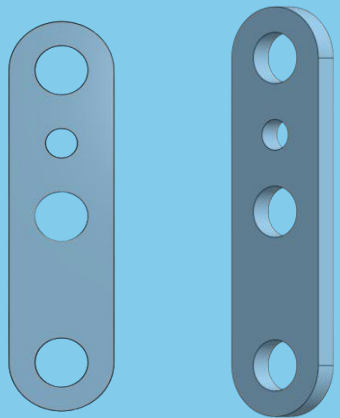
Synoptique :



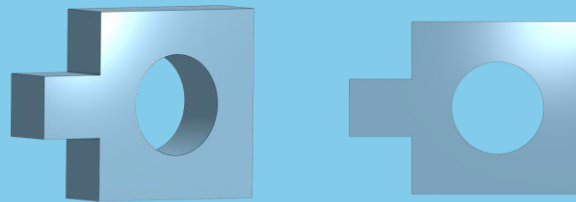
Travail individuel : Mathis

Croquis/Onshape :

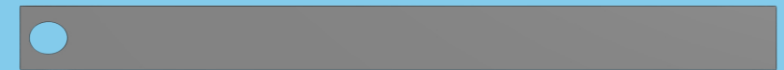
Feux tricolores:



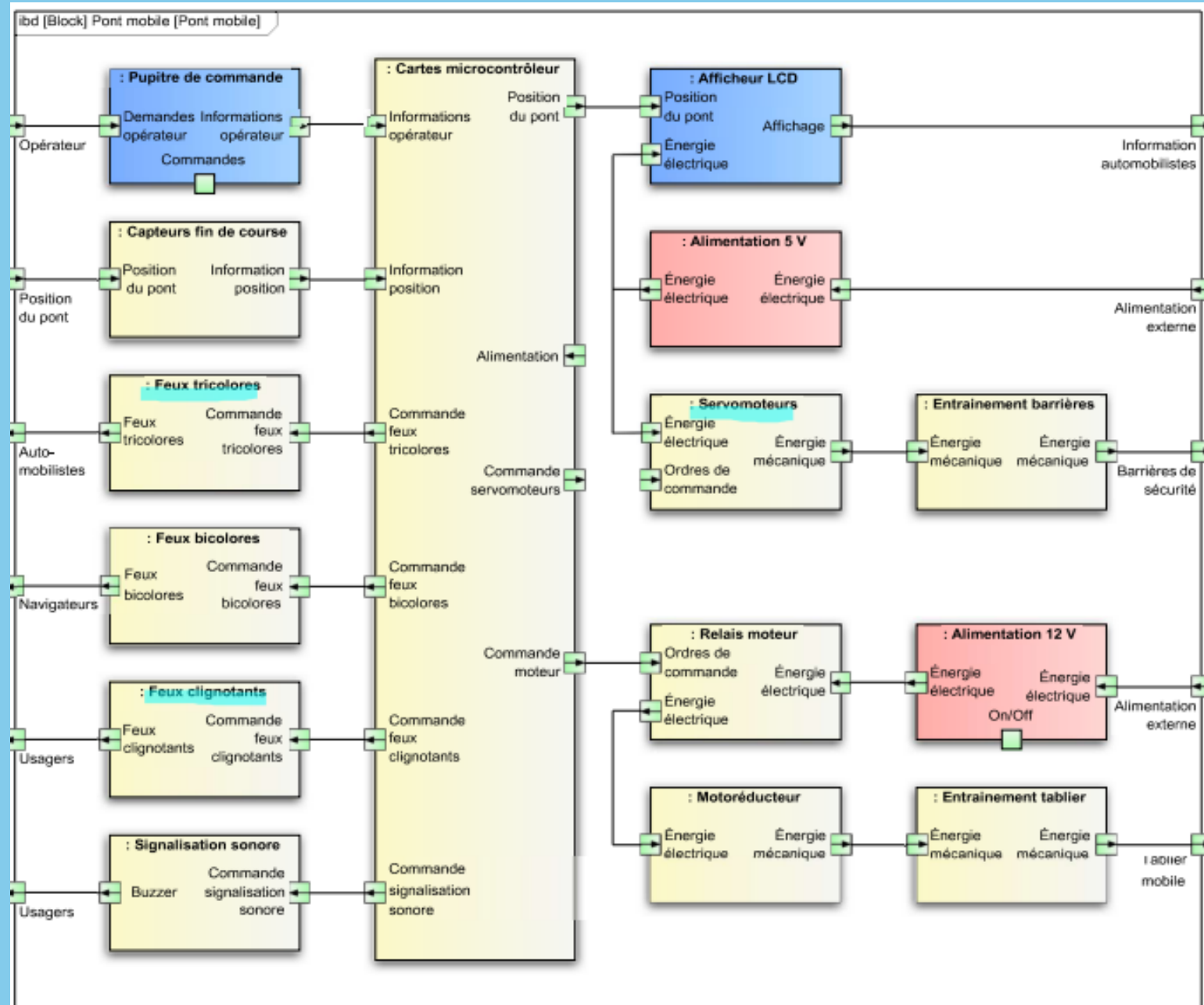
Lampadaire:



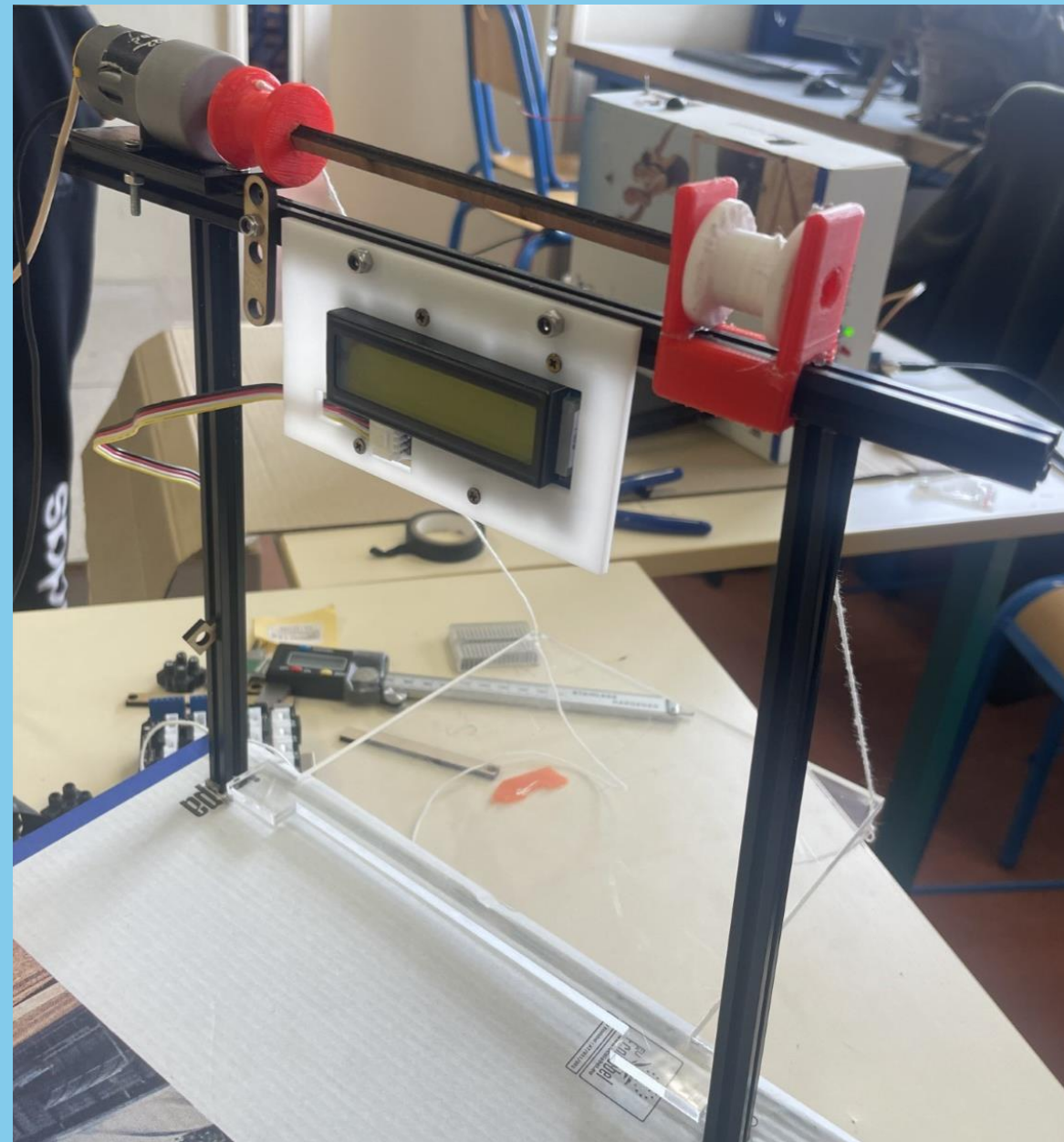
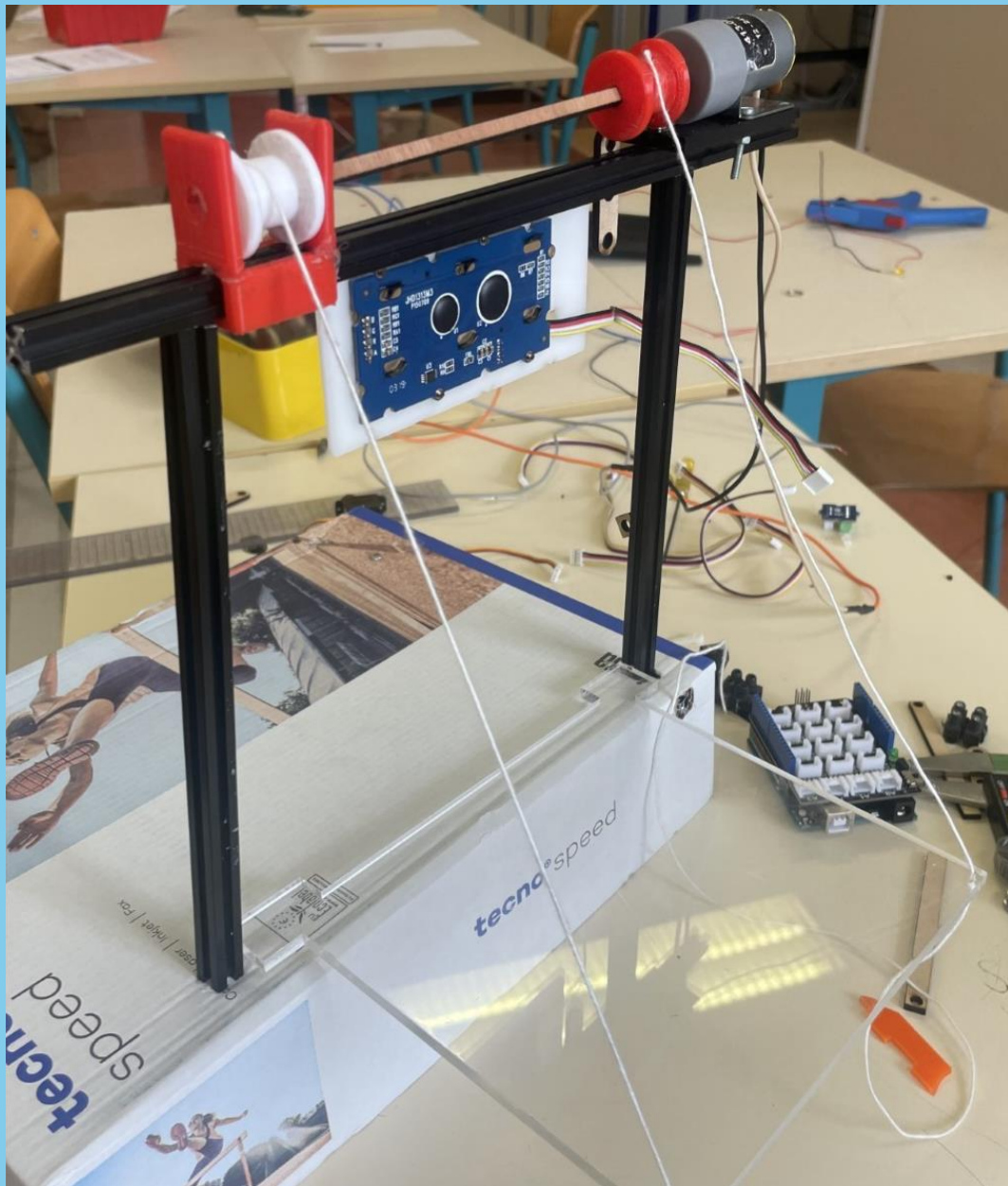
Barrière:



Chaine d'énergie et/ou d'information :



Réalisation :



Bilan technique/Conclusion :

Nous devions réaliser un pont mobile en 36 heures. Bien que la partie conception route qui étais sous ma responsabilité a été réalisé entièrement, nous n'avons, malheureusement, pas pu aboutir à bien ce projet. Nous n'avons pas su gérer notre temps, bien répartir les tâches entre nous et nous n'étions pas coordonnés. J'ai malgré tout appris de nouvelles choses et apprécié travailler sur ce projet. Il y a quand même de bonnes grosses bases qui ont été réalisées.

Travail individuel : Alexis

Algorithme :

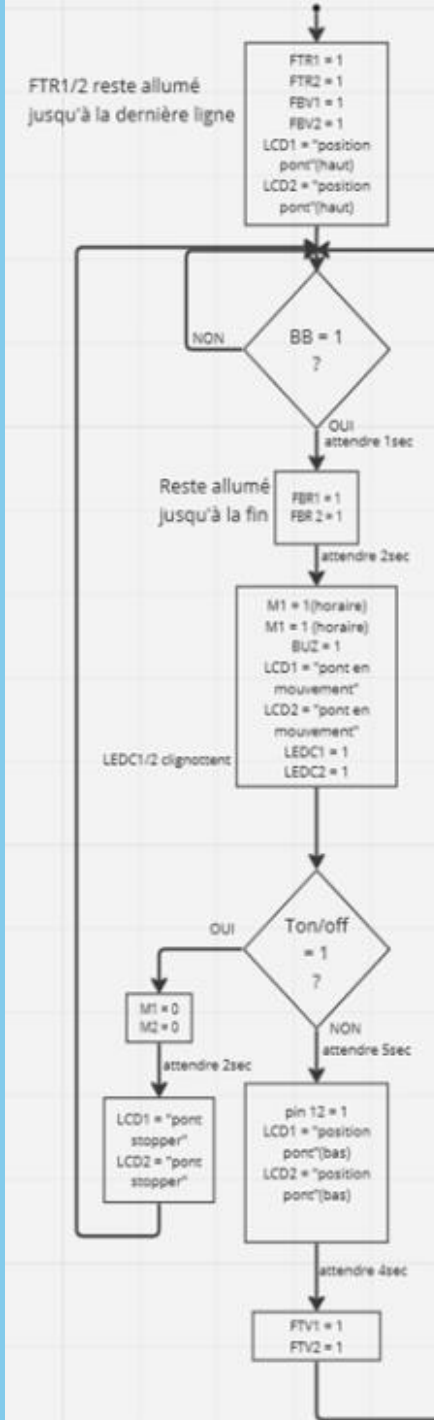
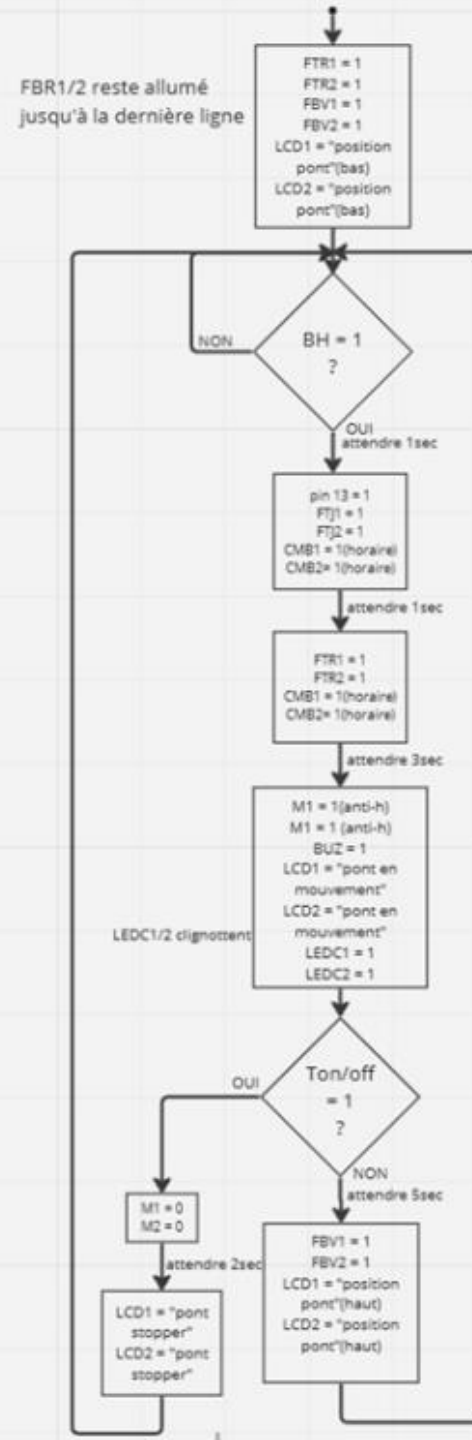
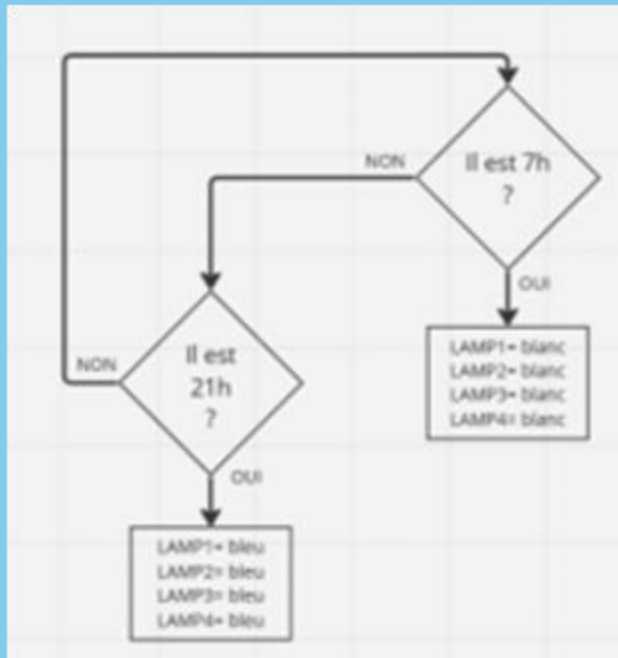
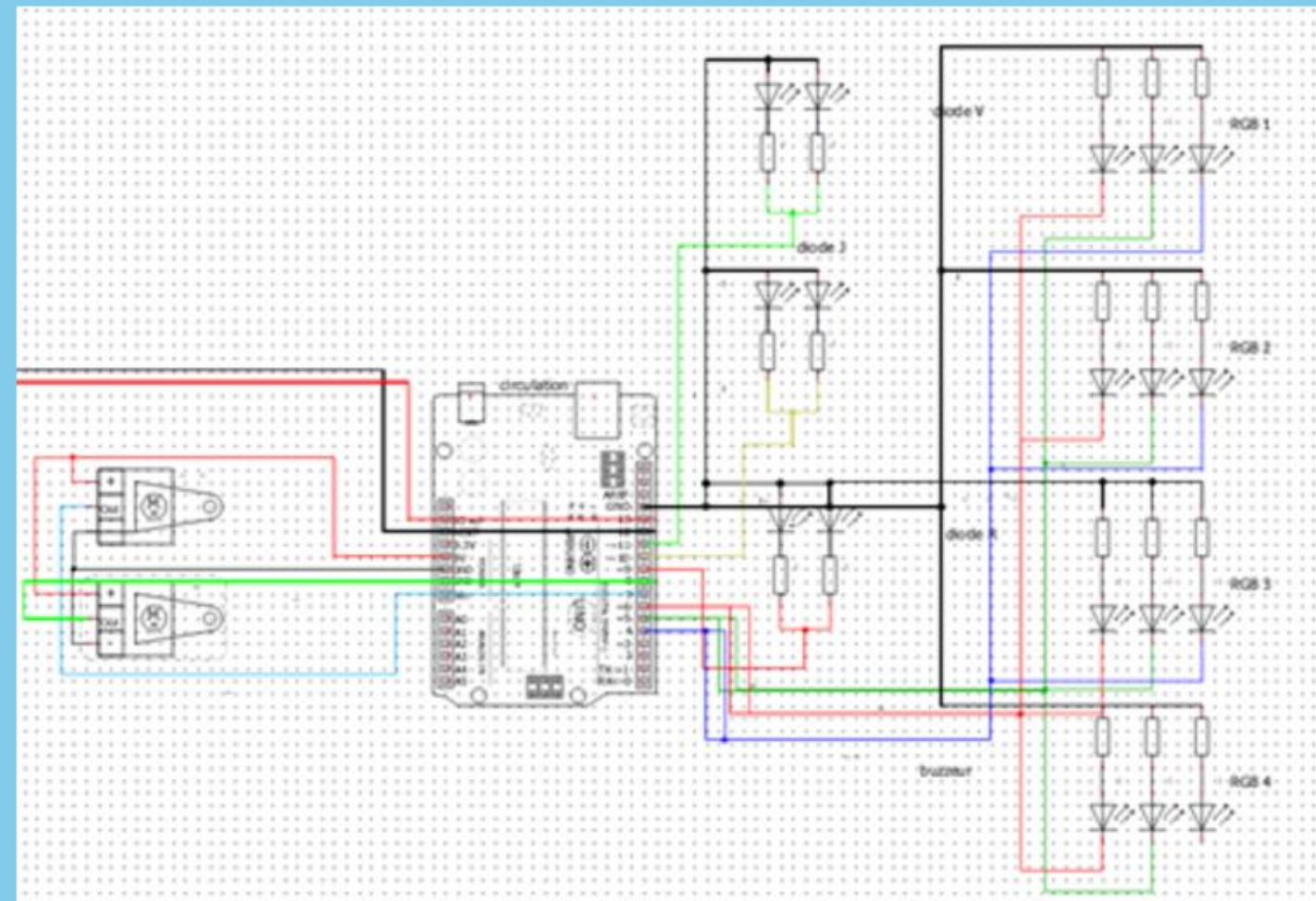
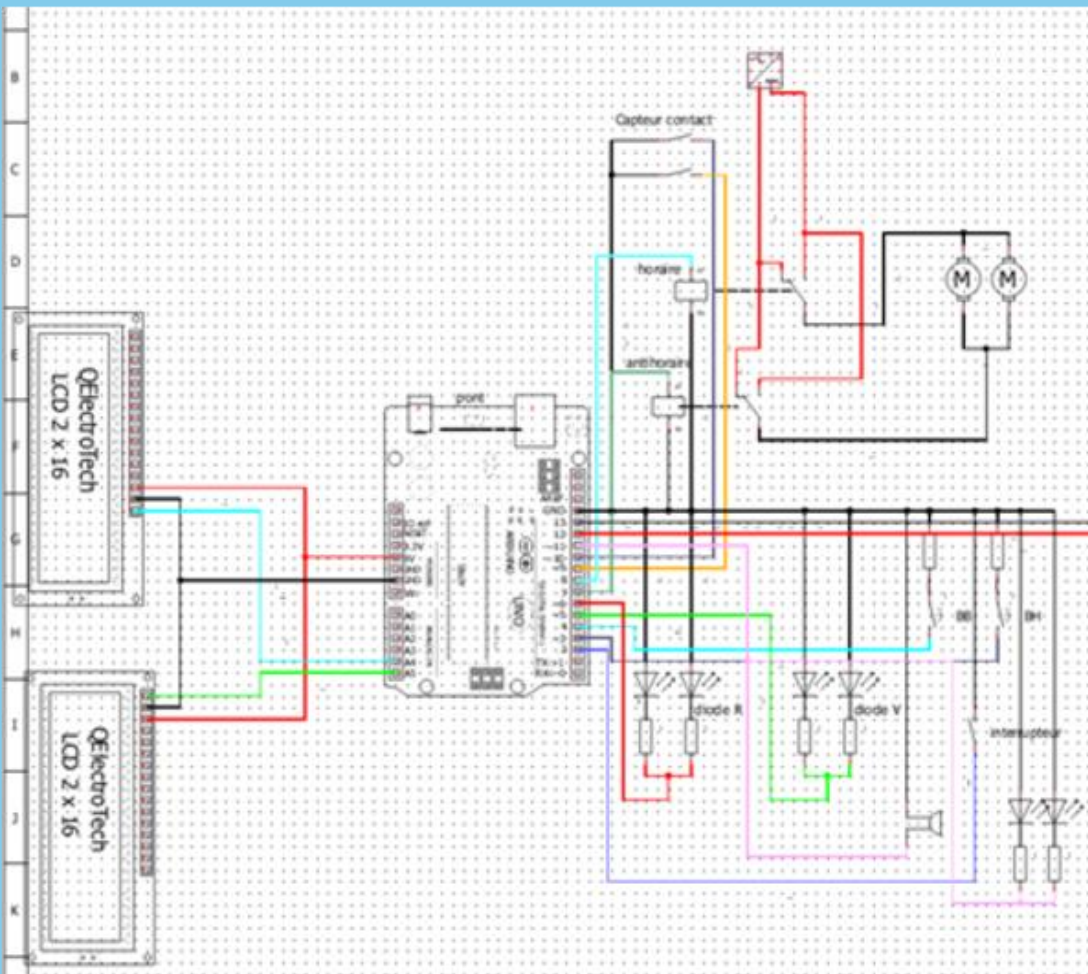


Schéma QElectrotech:



Code Arduino :

```
#include <Servo.h>
#include <RTClib.h>
#include <Wire.h>

#define FTR 9 //feux tricolores rouges
#define FTJ 10 //feux tricolores jaunes
#define FTV 11 //feux tricolores verts
#define pin13 13
#define pin12 12

RTC_DS3231 rtc;

//barrière automatique
Servo CMB1; // Servo moteur barrière automatique 1
Servo CMB2; // Servo moteur barrière automatique 2

//RGB
const int redpin = 6;
const int greenpin = 5;
const int bluepin = 4;

void setColor(int redValue, int greenValue, int blueValue) {
  analogWrite(redpin, redValue);
  analogWrite(greenpin, greenValue);
  analogWrite(bluepin, blueValue);
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(FTR, OUTPUT);
  pinMode(FTJ, OUTPUT);
  pinMode(FTV, OUTPUT);
  pinMode(redpin, OUTPUT);
  pinMode(greenpin, OUTPUT);
  pinMode(bluepin, OUTPUT);
  pinMode(pin13, INPUT);
  pinMode(pin12, INPUT);
  setColor(0, 0, 0);
  CMB1.attach(7); //cerveau moteur barrière auto
  CMB2.attach(8); //cerveau moteur barrière auto

  if (!rtc.begin()) {
    Serial.println("RTC non trouvé !");
    while (1);
  }
```

```
if (rtc.lostPower()) {
  Serial.println("RTC a perdu de l'alimentation, configurer l'heure !");
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}

void loop() {
  // put your main code here, to run repeatedly:
  DateTime now = rtc.now();
  digitalWrite(FTV, HIGH);
  digitalWrite(FTJ, LOW);
  digitalWrite(FTR, LOW);
  CMB1.write(0);
  CMB2.write(0);
  if (now.hour() == 7) {
    setColor(255, 255, 255); // Blanc
  } else if (now.hour() == 21) {
    setColor(0, 0, 255); // Bleu
  } else {
    setColor(0, 0, 0); // Éteint
  }
  if (pin13 == 1) {
    delay(2000);
    digitalWrite(FTJ, HIGH);
    CMB1.write(90);
    CMB2.write(90);
    delay(1000);
    digitalWrite(FTR, HIGH);
  }
  else {
    digitalWrite(FTJ, LOW);
    digitalWrite(FTR, LOW);
    CMB1.write(0);
    CMB2.write(0);
  }
  if (pin12 == 1) {
    delay(2000);
    CMB1.write(0);
    CMB2.write(0);
    delay(2000);
    digitalWrite(FTV, HIGH);
  }
  else {
    digitalWrite(FTV, LOW);
    CMB1.write(90);
    CMB2.write(90);
  }
}
```

Table d'adressage :

| Nom | Type | Adresse |
|----------|--------------------|------------------|
| FTR1 | LED rouge | pin 9 |
| FTR2 | LED rouge | pin 9 |
| FBR1 | LED rouge | pin 6 |
| FBR2 | LED rouge | pin 6 |
| FTJ1 | LED jaune | pin 10 |
| FTJ2 | LED jaune | pin 10 |
| LCD1 | écran LCD | A5 |
| LCD2 | écran LCD | A4 |
| FTV1 | LED verte | pin 11 |
| FTV2 | LED verte | pin 11 |
| FBV1 | LED verte | pin 5 |
| FBV2 | LED verte | pin 5 |
| BH | Bouton poussoir | pin 3 |
| BB | Bouton poussoir | pin 4 |
| LAMP1 | LED RGB | R=6 G=5 B=4 |
| LAMP2 | LED RGB | R=6 G=5 B=4 |
| LAMP3 | LED RGB | R=6 G=5 B=4 |
| LAMP4 | LED RGB | R=6 G=5 B=4 |
| T ON/OFF | Interrupteur | pin 2 |
| BUZ | Buzeur | pin 11 |
| M1 | Moteur | pin 7 sur relais |
| M2 | Moteur | pin 8 sur relais |
| CMB1 | Serveau moteur | pin 7 sur relais |
| CMB2 | Serveau moteur | pin 8 sur relais |
| CCH | Capteur de contact | pin 9 |
| CCB | Capteur de contact | pin 10 |
| LEDC1 | LED jaune | pin 9 |
| LEDC2 | LED jaune | pin 9 |

Pont : Arduino pont

Route : Arduino route

Bilan technique/Conclusion :

Ce projet a posé quelques problèmes techniques qui m'as obligé à revoir plusieurs fois le schéma électrique comme la gestion des composants qui se révélaient bien mieux placer sur une des deux Arduino plutôt qu'une autre. J'ai aussi dû apprendre en même temps comment programmer certains composants comme les cerveaux moteurs.

D'après moi, ce projet a posé des problèmes au vus du grand nombre de composants et que nous avons surestimer le temps qu'allais nous prendre certaines partit du projet mais nous avons tout de même pu surmonter et résolus ces problèmes.

Travail individuel : Axel

Table d'adressage :

| <u>nom</u> | <u>type</u> | Adresse (pin) |
|------------|-----------------------------|------------------------|
| FBR1 | LED rouge | 6 |
| FBR2 | <u>Led</u> rouge | 6 |
| LCD1 | Écran <u>lcd</u> RGB | Protocole I2C |
| LCD2 | Écran <u>lcd</u> <u>rgb</u> | Protocole I2C |
| FBV1 | <u>Led</u> verte | 5 |
| FBV2 | <u>Led</u> verte | 5 |
| BH | <u>bouton</u> | 3 |
| BB | <u>bouton</u> | 4 |
| TON/OFF | Interrupteur | 2 |
| M1 | <u>moteur</u> | 7 et 8 (relais double) |
| M2 | <u>moteur</u> | 7 et 8 (relais double) |
| CC1H | Capteur de contact | 9 |
| CC2H | Capteur de contact | 11 |
| BUZ | <u>buzzeur</u> | 10 |
| JAUNE | <u>Leds</u> jaunes | 10 |

Code Arduino :

```
code : #include
<Wire.h> #include
<rgb_lcd.h> rgb_lcd
lcdRgb;

// Définition des broches const int
LUMIERE_VERTE = 6; const int
LUMIERE_ROUGE = 5; const int BUZZER =
10; const int CAPTEUR_FERMER = 3; const
int CAPTEUR_OUVERT = 11; const int
CAPTEUR_MOUVEMENT = 9; const int
CAPTEUR_SECONDAIRE = 2; const int
LUMIERE_SECONDAIRE = 7; const int
CAPTEUR_TERTIAIRE = 4; const int
LUMIERE_TERTIAIRE = 8; const int
LUMIERE_FINALE = 12;
```

```
void vert() { lcdRgb.setRGB(0,
255, 0); lcdRgb.setCursor(0,
0); lcdRgb.print("pont
ouvert"); lcdRgb.setCursor(0,
1);
lcdRgb.print("veuillez circulez");
}

void rouge() {
lcdRgb.clear();
lcdRgb.setCursor(0, 0);
lcdRgb.print("pont ferme");
lcdRgb.setRGB(255, 0, 0);
}

void Orange() { lcdRgb.clear();
lcdRgb.setCursor(0, 0);
lcdRgb.print("pont en mouvement");
lcdRgb.setRGB(255, 128, 0);
}

void bip() { digitalWrite(BUZZER,
HIGH); delay(500);
digitalWrite(BUZZER, LOW);
delay(500);
}

void test() {
for (int count = 0; count < 3; count++) {
lcdRgb.setCursor(0, 0);
lcdRgb.print("test"); lcdRgb.setRGB(0, 0,
255); digitalWrite(LUMIERE_VERTE,
HIGH);
digitalWrite(LUMIERE_ROUGE, HIGH);
delay(500);
digitalWrite(LUMIERE_ROUGE, LOW);
digitalWrite(LUMIERE_VERTE, LOW);
delay(500);
}
}

void setup() {
pinMode(LUMIERE_ROUGE, OUTPUT);
pinMode(LUMIERE_VERTE, OUTPUT);
```

```

lcdRgb.begin(16, 2);
pinMode(CAPTEUR_FERMER, INPUT);
pinMode(CAPTEUR_OUVERT, INPUT); pinMode(LUMIERE_FINALE,
OUTPUT); pinMode(CAPTEUR_MOUVEMENT, INPUT);
pinMode(CAPTEUR_SECONDAIRE, INPUT);
pinMode(LUMIERE_SECONDAIRE, OUTPUT);
pinMode(CAPTEUR_TERTIAIRE, INPUT);
pinMode(LUMIERE_TERTIAIRE, OUTPUT); pinMode(BUZZER,
OUTPUT);
test();
digitalWrite(LUMIERE_ROUGE, LOW);
digitalWrite(LUMIERE_VERT, LOW);
lcdRgb.clear();
}

void loop() {
digitalWrite(LUMIERE_VERT, HIGH); vert();

if (digitalRead(CAPTEUR_FERMER) == HIGH && digitalRead(CAPTEUR_OUVERT) == LOW) {
digitalWrite(LUMIERE_FINALE, HIGH);
Orange();
while (digitalRead(CAPTEUR_MOUVEMENT) == LOW) { bip();
}
delay(4000);
if (digitalRead(CAPTEUR_SECONDAIRE) == HIGH) {
digitalWrite(LUMIERE_SECONDAIRE, HIGH);
}
}

if (digitalRead(CAPTEUR_MOUVEMENT) == HIGH) {
digitalWrite(LUMIERE_SECONDAIRE, LOW);
digitalWrite(LUMIERE_ROUGE, HIGH);
digitalWrite(LUMIERE_VERT, LOW); rouge();
}

if (digitalRead(CAPTEUR_TERTIAIRE) == HIGH && digitalRead(CAPTEUR_MOUVEMENT) == HIGH) {
Orange();
digitalWrite(LUMIERE_VERT, HIGH);
digitalWrite(LUMIERE_ROUGE, LOW); while
(digitalRead(CAPTEUR_OUVERT) == LOW) { bip();
}
if (digitalRead(CAPTEUR_SECONDAIRE) == HIGH) {
digitalWrite(LUMIERE_TERTIAIRE, HIGH);
}
}
}

```

```

}

if (digitalRead(CAPTEUR_OUVERT) == HIGH) {
digitalWrite(LUMIERE_FINALE, HIGH);
digitalWrite(LUMIERE_TERTIAIRE, LOW);
}
}

```

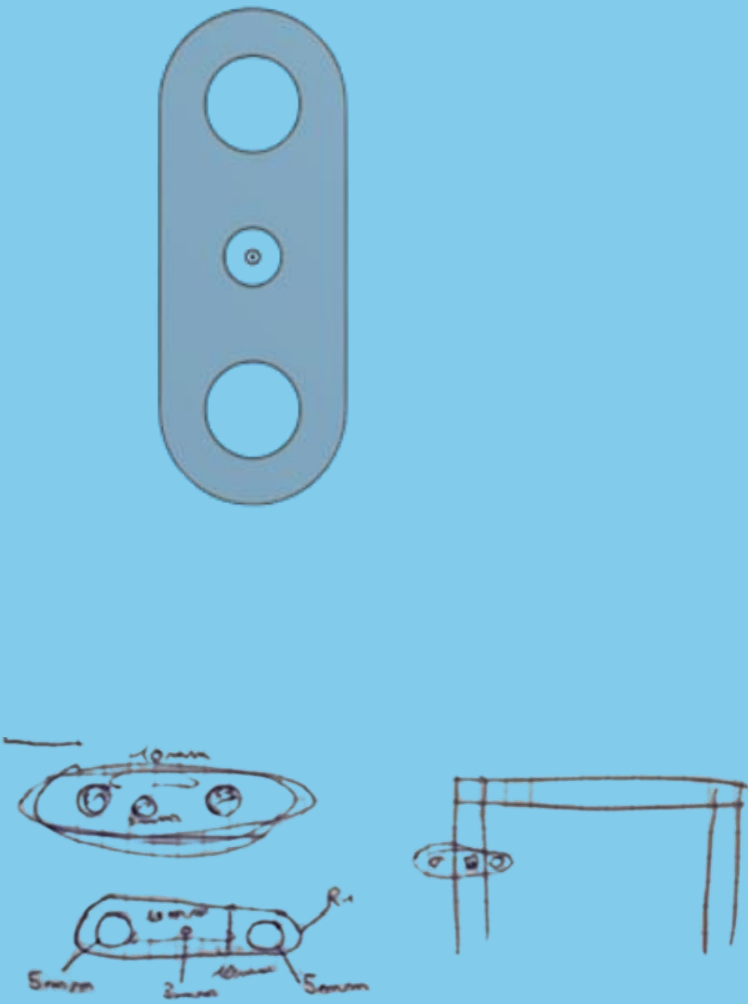
Bilan technique/Conclusion :

Je suis content et inquiet, même si notre projet ne fonctionne pas, je suis fier de moi et de ce que mes camarades ont fait, car ce travail est notre œuvre à tous les 4.

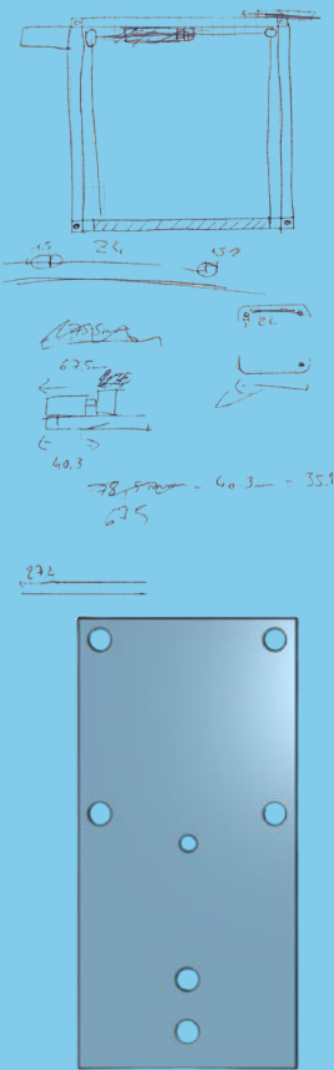
Travail individuel : William

Croquis/Onshape :

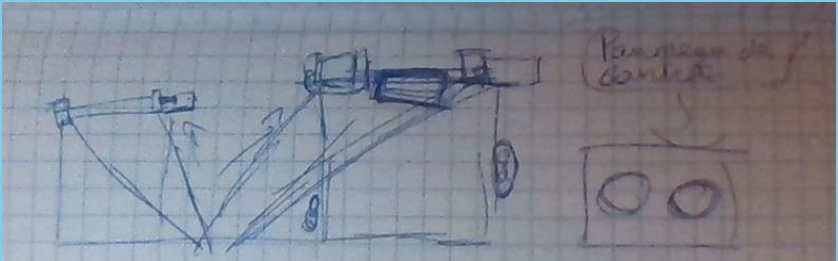
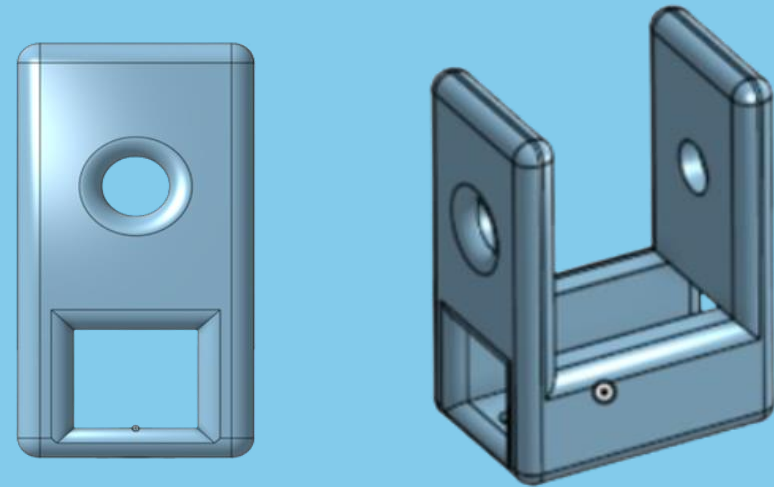
Feux bicolores :



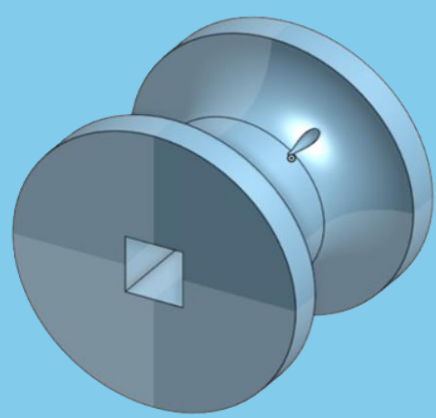
Support moteur :



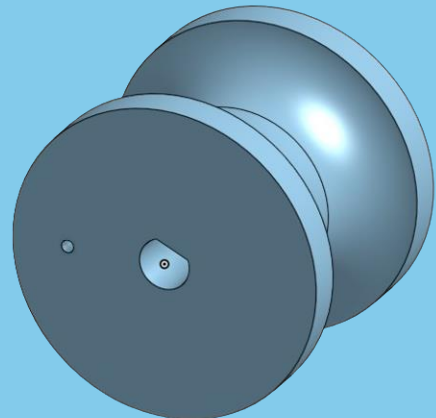
Support poulie :



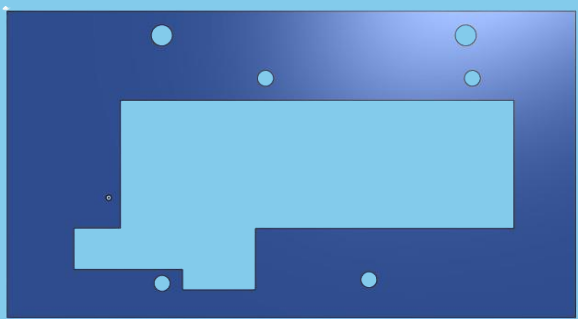
Poulie libre :



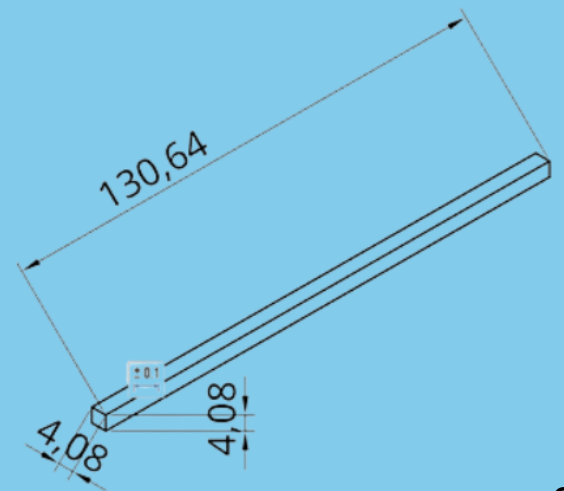
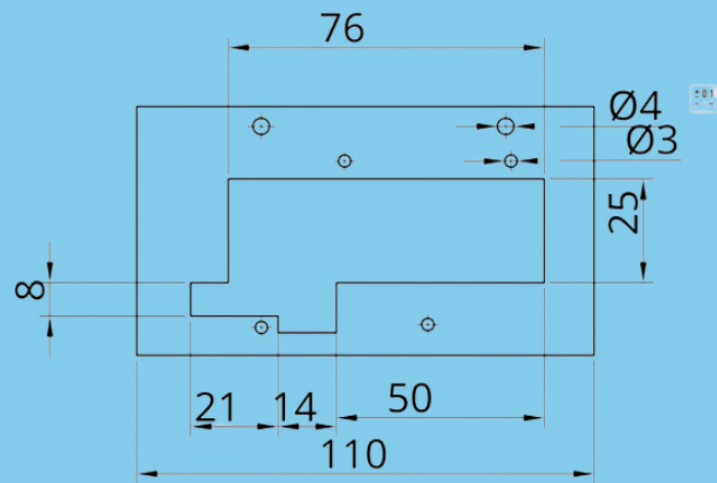
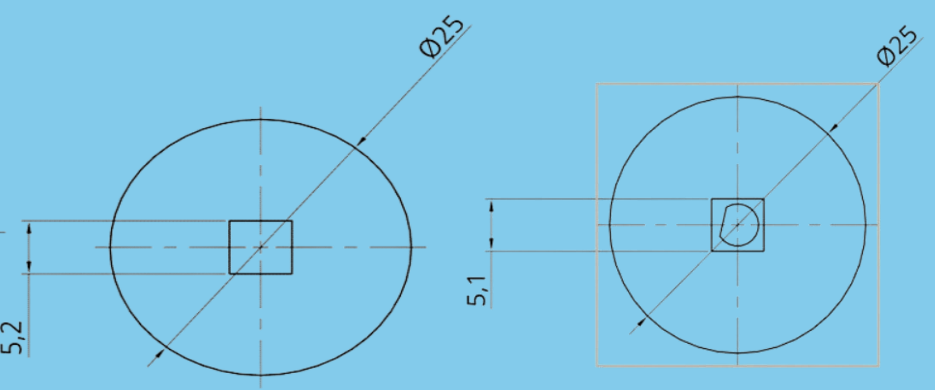
Poulie motrice :



Support écran :

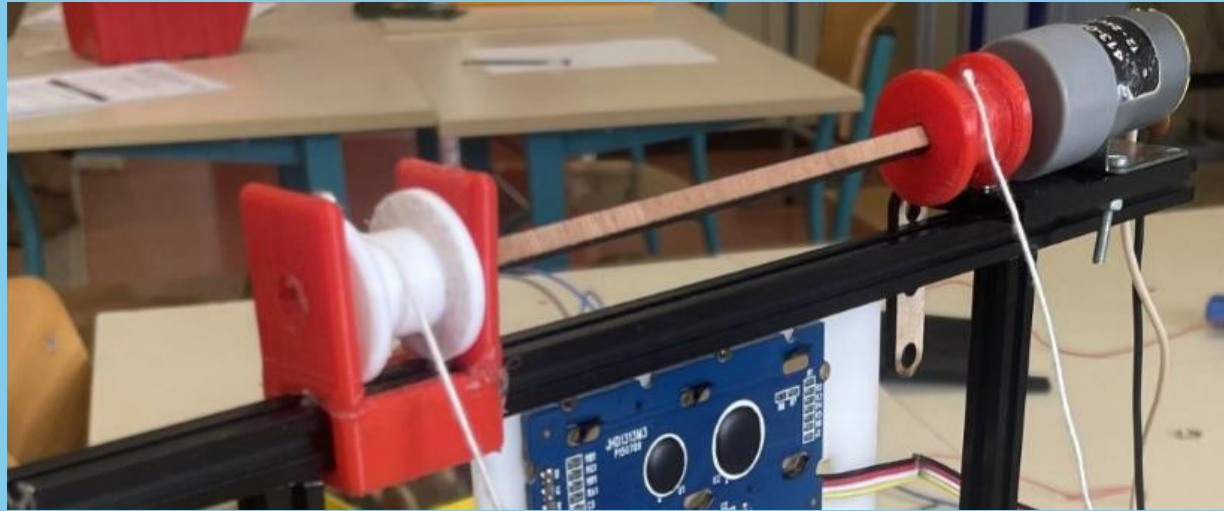


Transmission du moteur à la poulie :



Réalisation :

William



Bilan technique/Conclusion :

Ce projet nous a permis de progresser dans notre travail en équipe. Chacun a accompli les tâches qui lui étaient attribuées, et nous avons obtenu un excellent résultat avec un pont fonctionnel lors des tests du 30 mai. Malgré quelques obstacles, nous les avons surmontés ensemble.

En conclusion, ce projet de pont mobile nous a apporté de précieuses connaissances pour nos futures études. J'ai principalement travaillé sur Onshape, ce qui a facilité la modélisation rapide des pièces. Mes camarades ont également fourni un excellent travail. Grâce à notre effort collectif, nous avons atteint un résultat final de grande qualité.

Présentation de Mathis, Alexis, Axel & William

Merci de nous avoir écouté