

## Rapport de séance

### Objectif de cette séance :

Lors de cette semaine de projet, nous devons passer un oral afin de présenter notre projet et son avancement et continuer notre projet.

Le but de cette séance pour mon binôme et moi-même était de créer notre application Bluetooth qui permettrait de gérer nos différents composants.

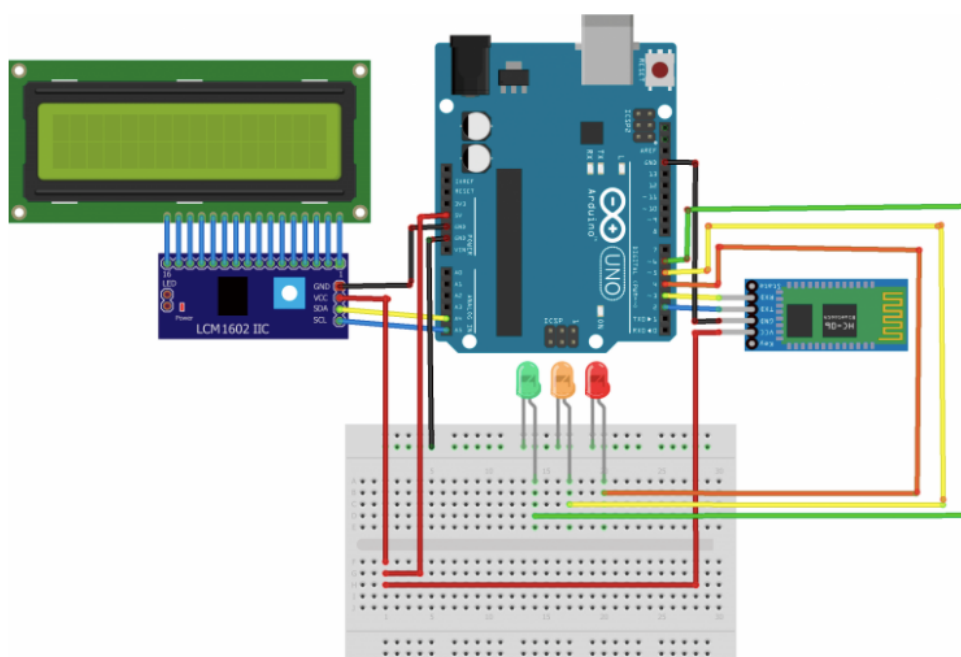
Pour mon binôme, il s'agissait de gérer l'ouverture et la fermeture des portes par Bluetooth et pour ma part, il s'agissait d'allumer des LEDs par Bluetooth.

### Déroulement de la séance :

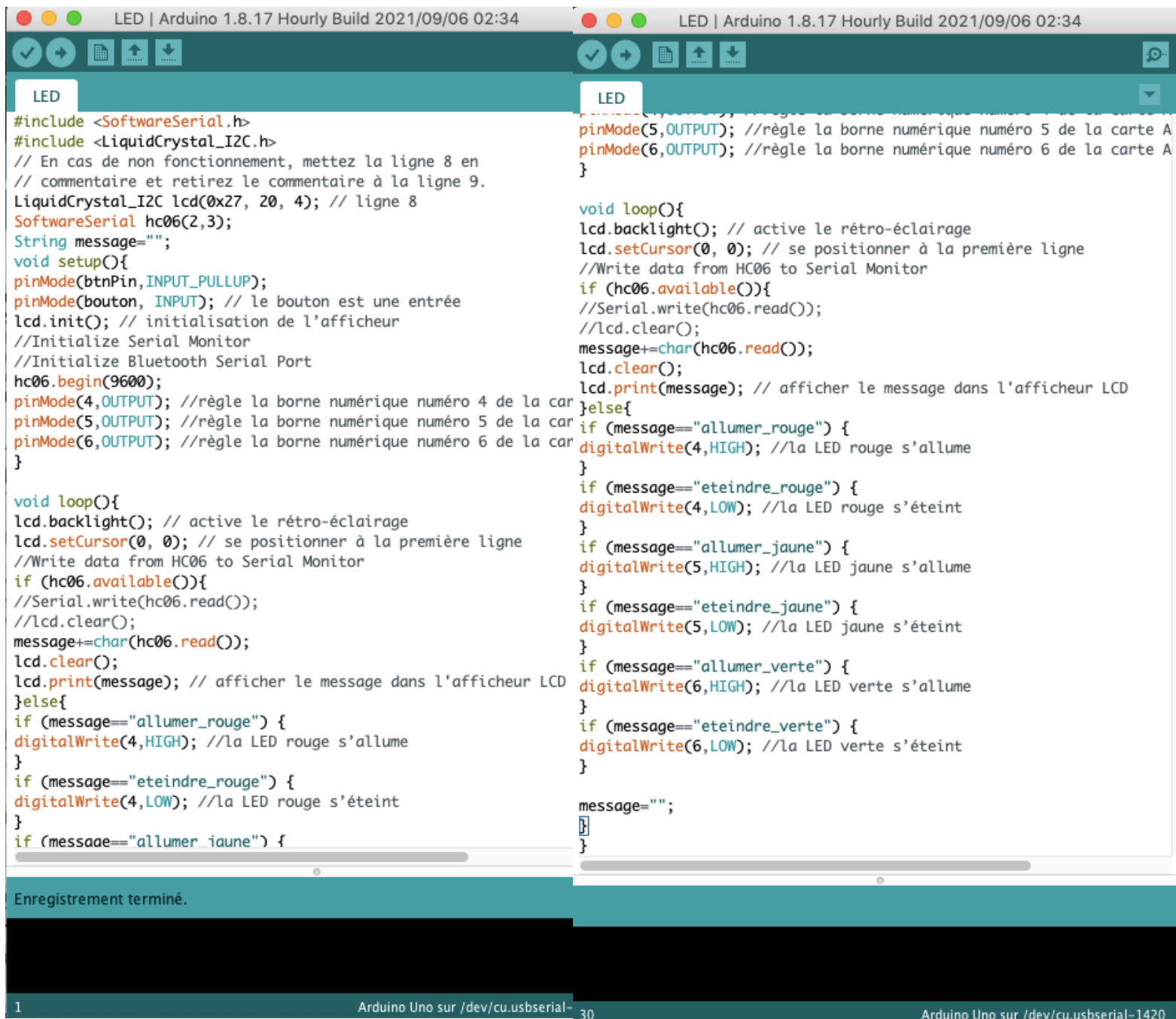
#### Création de l'application:

J'avais commencé l'application grâce au logiciel MIT App Inventor lors de la première séance. Cependant, je n'avais pas réussi à allumer une LED par Bluetooth. J'ai décidé de tout recommencer à zéro. Parmi les recherches que j'ai faites ce site : <https://www.robotique.tech/tutoriel/allumer-et-eteindre-des-leds-avec-smartphone-et-arduino/> expliquait comment allumer et éteindre les LEDS avec un smartphone.

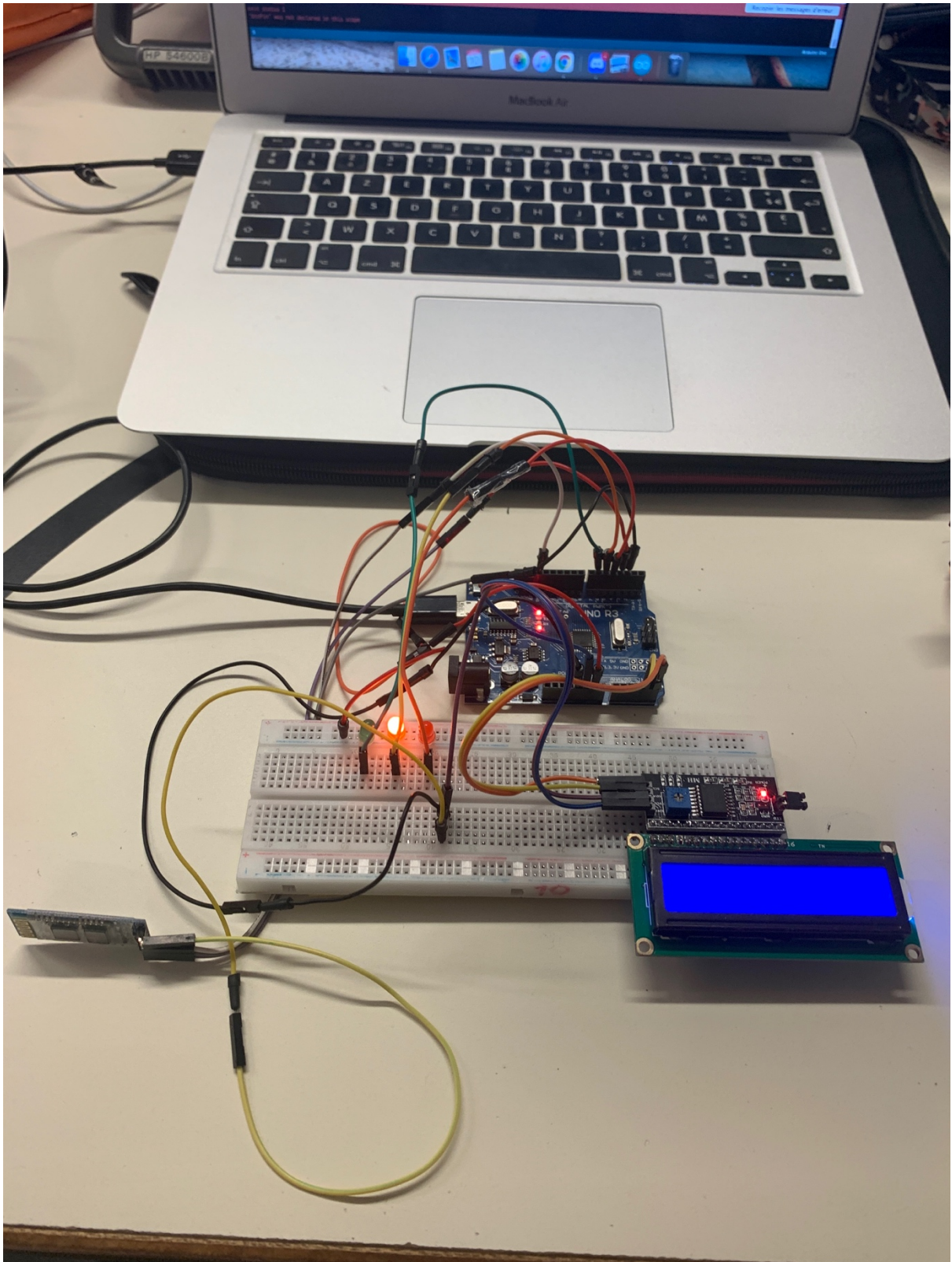
J'ai reproduit ce montage-ci :



Je m'étais inspiré de ce code mais je n'avais pas repris le même :



J'ai eu des soucis car aucune LEDs ni le tableau ne s'allumaient à part la LED orange. Voici le résultat :



Étant donné que mon ordinateur ne marche plus, je ne pourrai pas montrer le programme que j'ai effectué durant cette séance vis-à-vis de la création de l'application et de l'allumage des LEDs.

## Découverte du servomoteur:

Nous avons décidé d'ouvrir nos fenêtres en fonction de la température, autrement dit, lorsqu'il fait trop chaud dans la maison, les fenêtres s'ouvrent et lorsqu'il fait trop froid nos fenêtres se ferment.

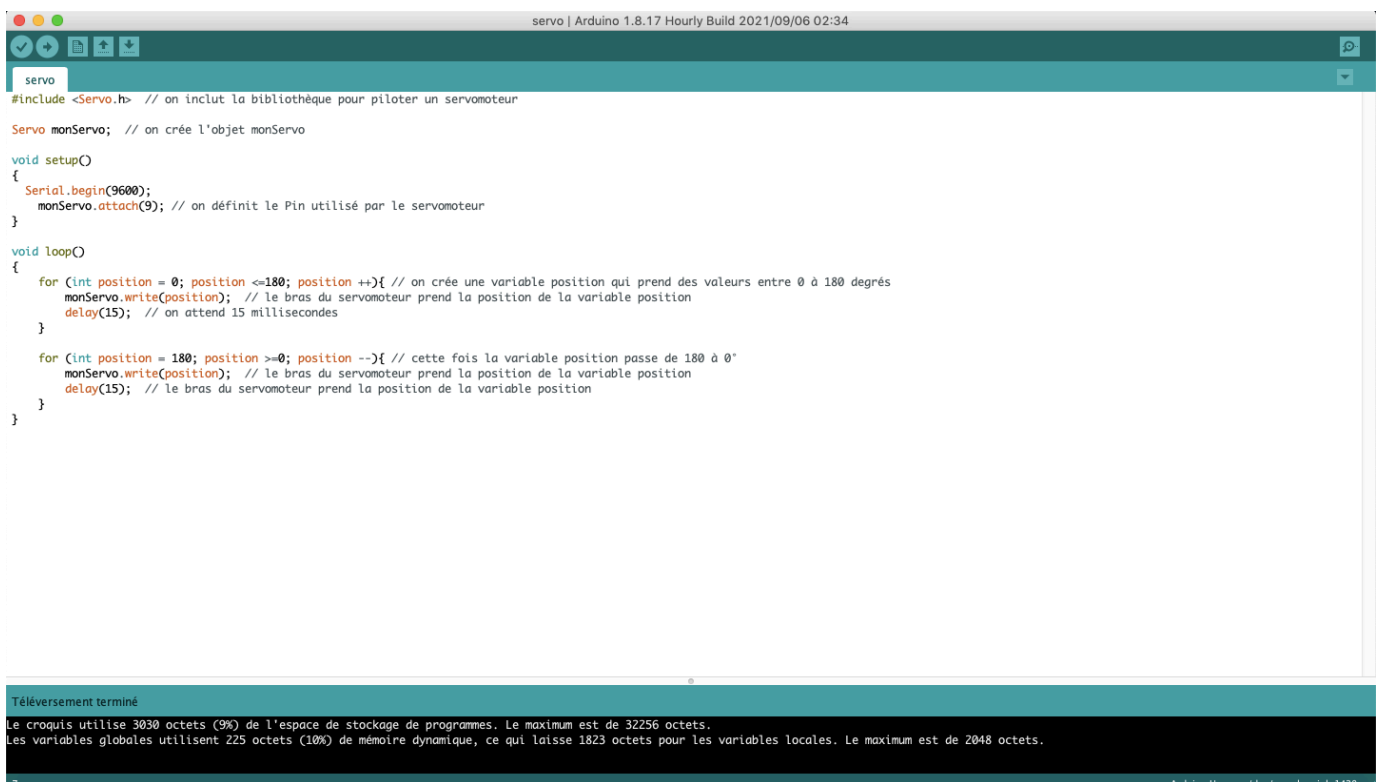
Afin d'apprendre à utiliser ce composant, j'ai commencé à faire des recherches sur le servomoteur TS90A.

Parmi les sites que j'ai trouvés, celui-ci m'a beaucoup aidé : <https://arduino.developpez.com/tutoriels/arduino-a-l-ecole/?page=projet-12-utiliser-un-servomoteur> .

Durant cette phase d'apprentissage, j'ai réussi à faire bouger le palonnier d'un servomoteur de manière progressive dans les deux sens.

Pour ce code, nous avons besoin de télécharger la bibliothèque Servo.

Voici le code qui permet de faire bouger le palonnier d'un servomoteur de manière progressive :



```
servo
#include <Servo.h> // on inclut la bibliothèque pour piloter un servomoteur

Servo monServo; // on crée l'objet monServo

void setup()
{
  Serial.begin(9600);
  monServo.attach(9); // on définit le Pin utilisé par le servomoteur
}

void loop()
{
  for (int position = 0; position <=180; position++){ // on crée une variable position qui prend des valeurs entre 0 à 180 degrés
    monServo.write(position); // le bras du servomoteur prend la position de la variable position
    delay(15); // on attend 15 millisecondes
  }

  for (int position = 180; position >=0; position--){ // cette fois la variable position passe de 180 à 0°
    monServo.write(position); // le bras du servomoteur prend la position de la variable position
    delay(15); // le bras du servomoteur prend la position de la variable position
  }
}
```

Téléversement terminé

Le croquis utilise 3030 octets (9%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.  
Les variables globales utilisent 225 octets (10%) de mémoire dynamique, ce qui laisse 1823 octets pour les variables locales. Le maximum est de 2048 octets.

Arduino Uno rev.3 (atmega328p)-1470

## Code de l'ouverture des fenêtres:

Pour le moment, nous n'avons pas encore défini à quelles températures nos fenêtres doivent se fermer et s'ouvrir. Par conséquent, j'ai affecté une rotation de 40 degrés pour chacune de nos fenêtres.

Voici le code pour l'ouverture des fenêtres :



```
servoFenetreOuverture | Arduino 1.8.17 Hourly Build 2021/09/06 02:34

servoFenetreOuverture
#include <Servo.h>

Servo monServo;

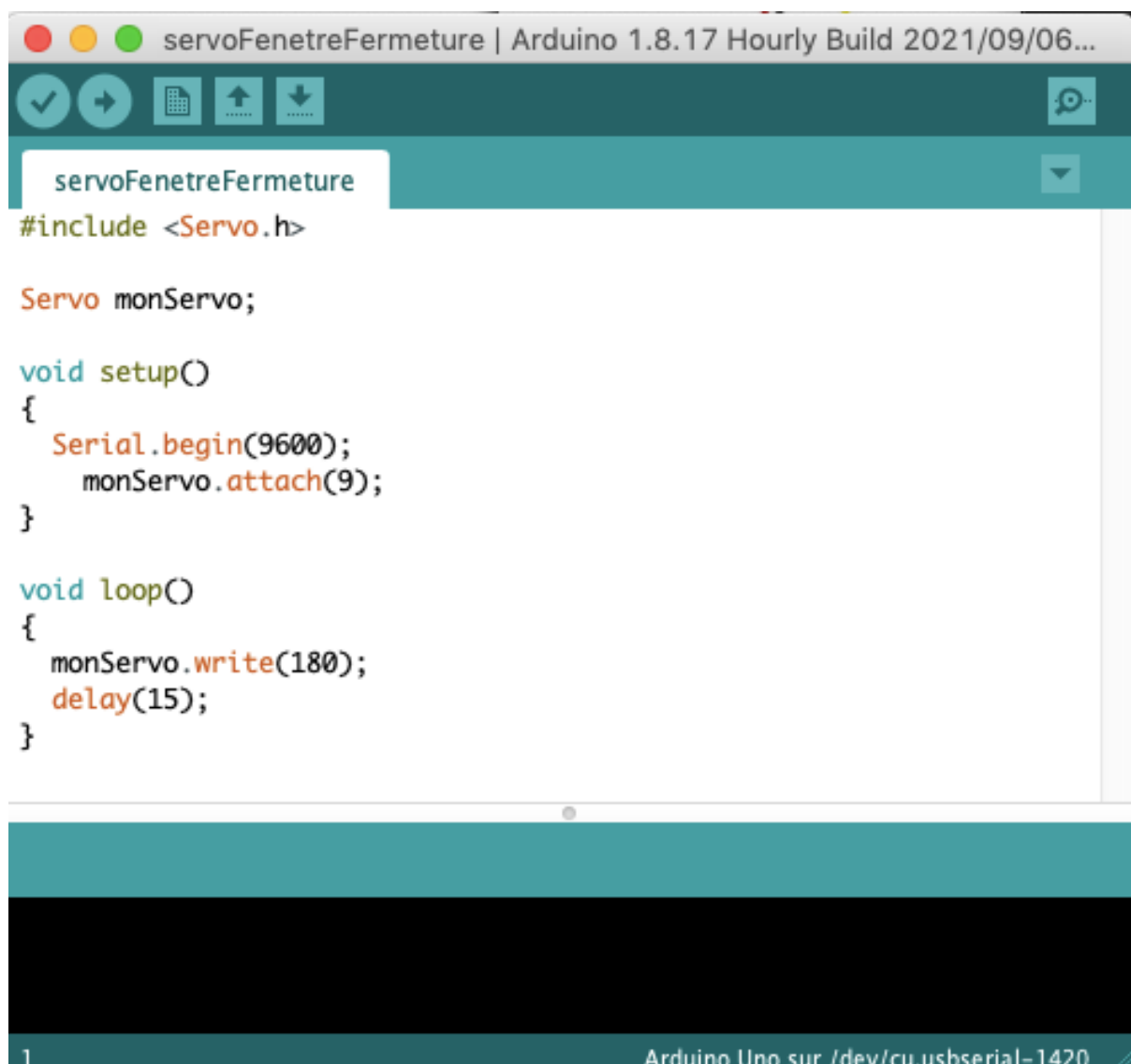
void setup()
{
  Serial.begin(9600);
  monServo.attach(9);
}

void loop()
{
  monServo.write(40);
  delay(15);
}
```

15 Arduino Uno sur /dev/cu.usbserial-1420



Voici le code pour la fermeture des fenêtres :

A screenshot of the Arduino IDE interface. The title bar at the top reads "servoFenetreFermeture | Arduino 1.8.17 Hourly Build 2021/09/06...". Below the title bar is a toolbar with icons for checking, running, uploading, and downloading. The main text area contains the following C++ code:

```
servoFenetreFermeture
#include <Servo.h>

Servo monServo;

void setup()
{
  Serial.begin(9600);
  monServo.attach(9);
}

void loop()
{
  monServo.write(180);
  delay(15);
}
```

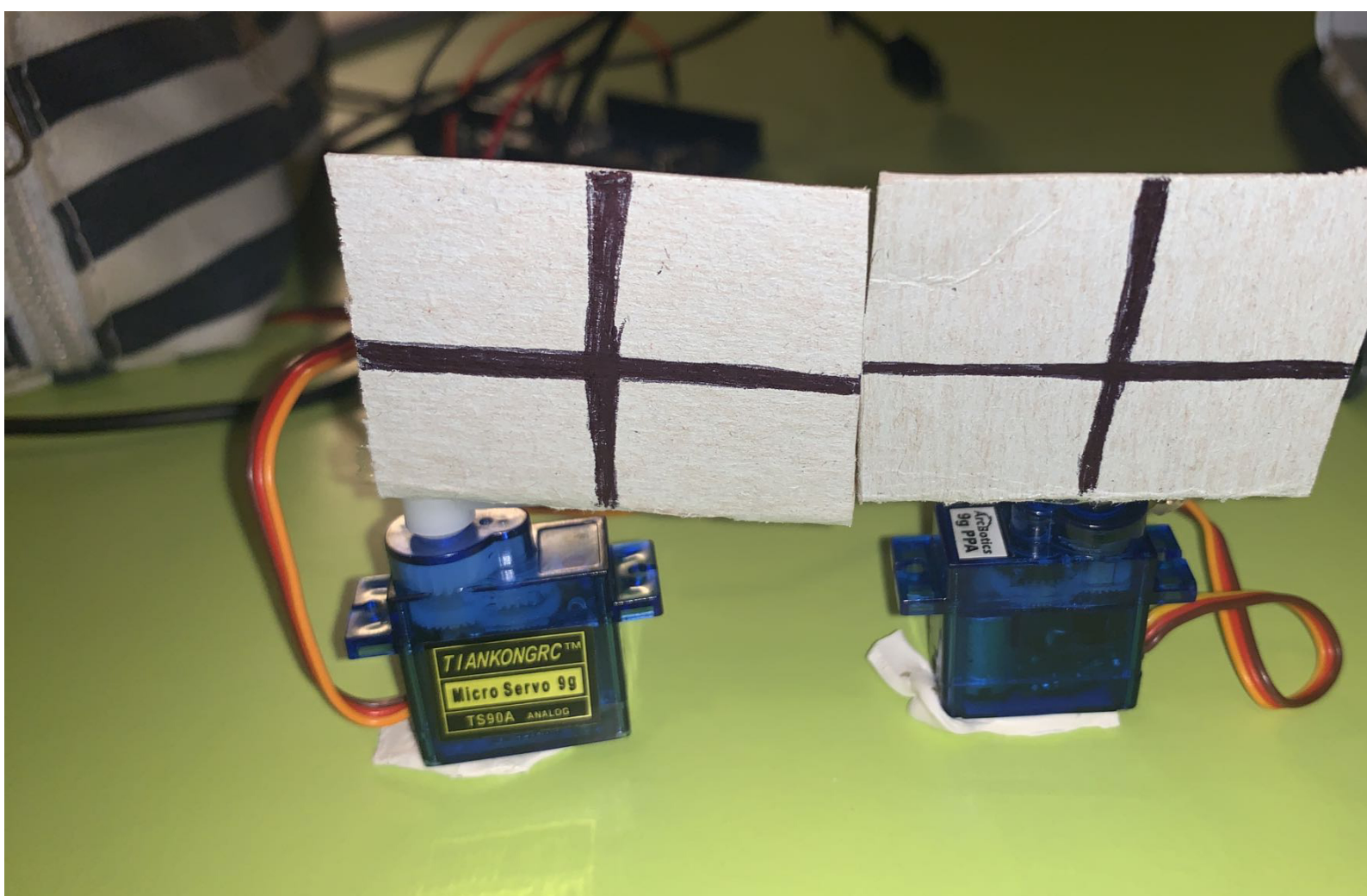
The bottom status bar shows "1" on the left and "Arduino Uno sur /dev/cu.usbserial-1420" on the right.

Lors de cette séance, mon binôme effectue le programme qui permet de détecter la température à l'intérieur de la maison. Ainsi, une fois qu'elle aura fini le thermomètre et que nous aurons décidé des valeurs seuils de température (pour ouvrir/fermer portes), je l'ajouterai au programme et rajouterai des conditions (if..).

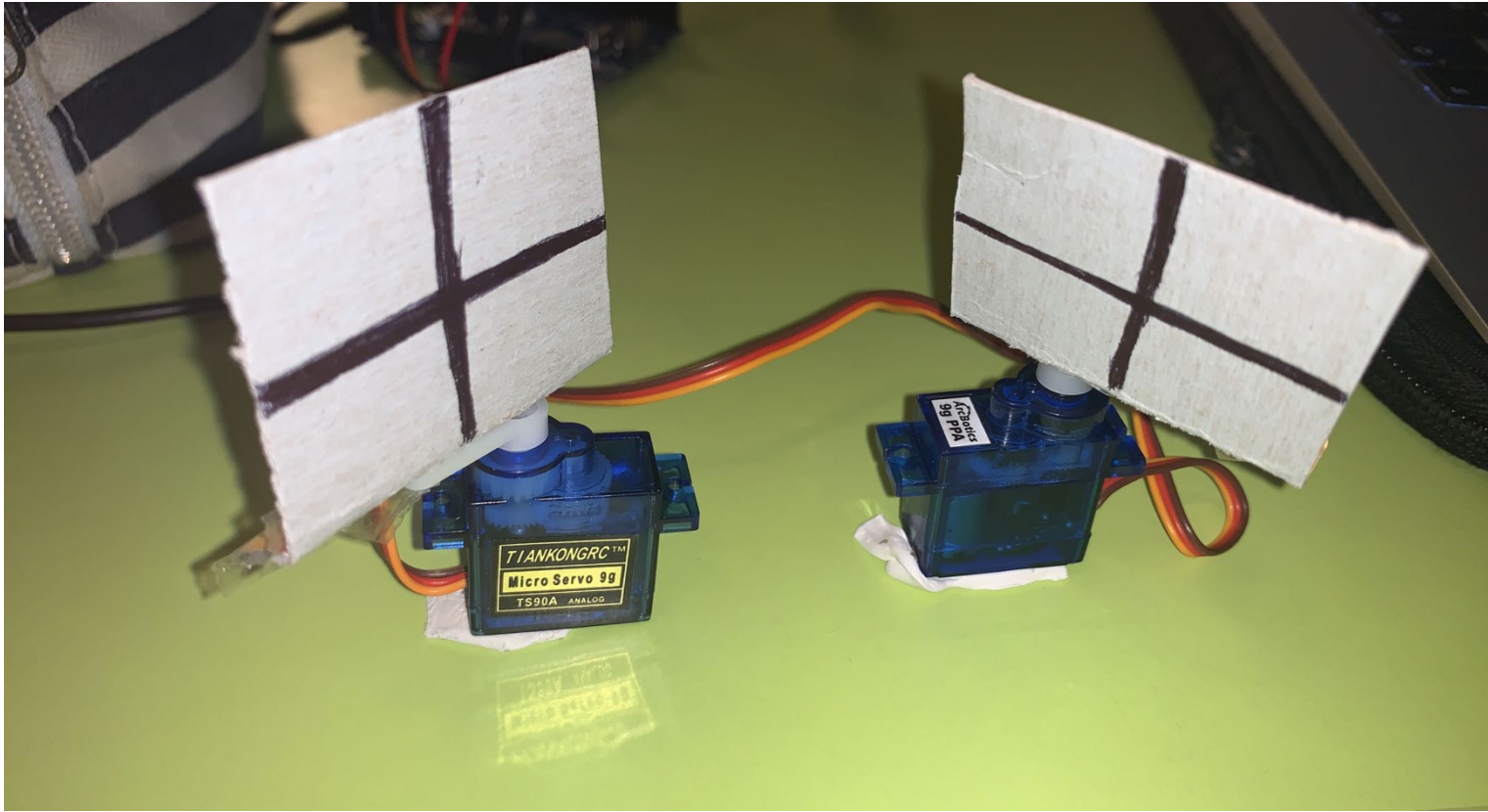
### Résultats:

J'ai réalisé chaque fenêtre en carton puis je les ai collés au palonnier du servomoteur afin de visualiser le rendu.

Voici le résultat lorsque les fenêtres sont fermées :



Voici le résultat lorsque les fenêtres sont ouvertes :



### Travail à la maison :

Durant cette semaine, étant donné que nous avons eu à passer un oral, j'ai continué chez moi à programmer l'horloge sur un tableau LCD.

En faisant quelques recherches, j'ai abouti à ce programme-ci :



horloge | Arduino 1.8.17 Hourly Build 2021/09/06 02:34

✓ → 📄 ⬆ ⬇

🗨

horloge ▾

```
#include <LiquidCrystal.h>
#include <Wire.h>
#include <ds3231.h>

struct ts t; //déclaration de la variable t
struct ts nt; //déclaration de la variable nt. Si j'utilise t partout

LiquidCrystal lcd(2,3,4,5,6,7);

void setup() {
  Serial.begin(9600);
  lcd.begin(16,2);
  lcd.setCursor(0,0); // (colonne,ligne) 0 est le premier indice
  Serial.println("Test");
  lcd.print("Il est :");

  Wire.begin();
  DS3231_init(DS3231_CONTROL_INTCN); //ligne modifiée

  //Réglage de l'heure
  /*nt.mday = 06;
  nt.mon = 01;
  nt.year = 2022;
  nt.hour = 22;
  nt.min = 10;
  nt.sec=0;
  DS3231_set(nt);
  */

  //L'heure est réglée
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Enregistrement terminé.

1 Arduino Uno sur /dev/cu.usbserial-1420

horloge | Arduino 1.8.17 Hourly Build 2021/09/06 02:34

✓ → 📄 ⬆ ⬇

🔍

horloge ▾

```
nt.min = 10;
nt.sec=0;
DS3231_set(nt);
*/

//L'heure est réglée

}

void loop() {
    // put your main code here, to run repeatedly:

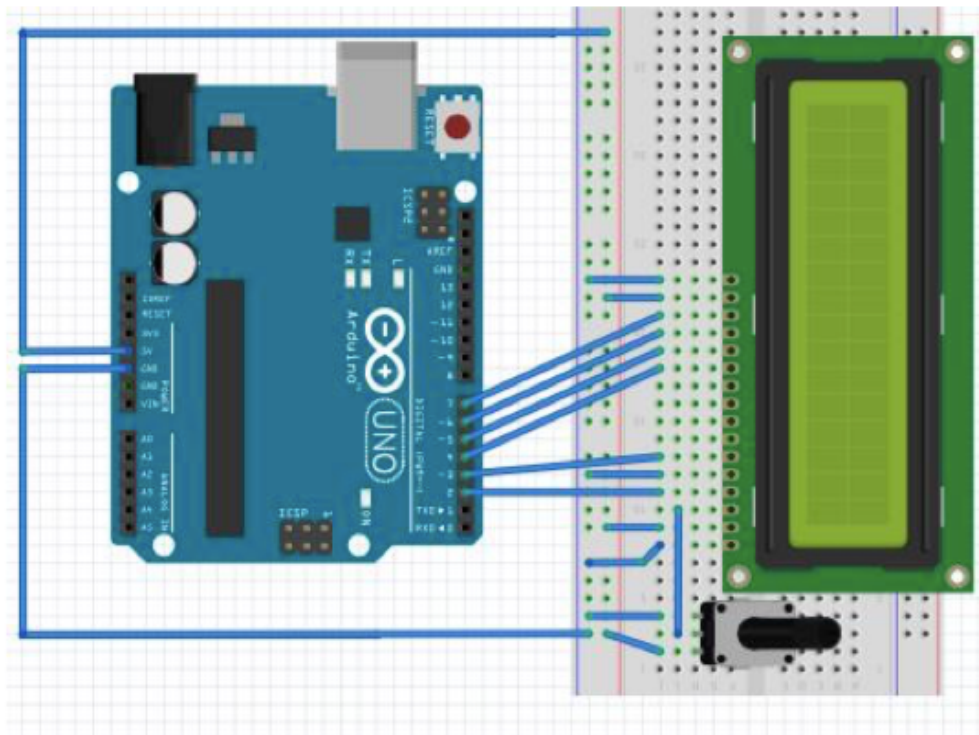
    DS3231_get(&t);
    Serial.println(t.year);
    Serial.print("date : ");
    Serial.print(t.mday);
    Serial.print("/");
    Serial.print(t.mon);
    Serial.print("/");
    Serial.print(t.year);
    Serial.print("\tHeure : ");
    Serial.print(t.hour);
    lcd.setCursor(0,1);
    lcd.print(t.hour);
    Serial.print(":");
    Serial.print(t.min);
    lcd.print("h ");
    lcd.print(t.min);
    Serial.print(".");
    Serial.println(t.sec);
    lcd.print("min ");
    lcd.print(t.sec);
    lcd.print("sec");

    delay(1000);
}
```

Enregistrement terminé.

15 Arduino Uno sur /dev/cu.usbserial-1420

Pour le câblage de l'écran LCD, j'ai utilisé le diaporama de Mr Masson :



Voici le résultat :

