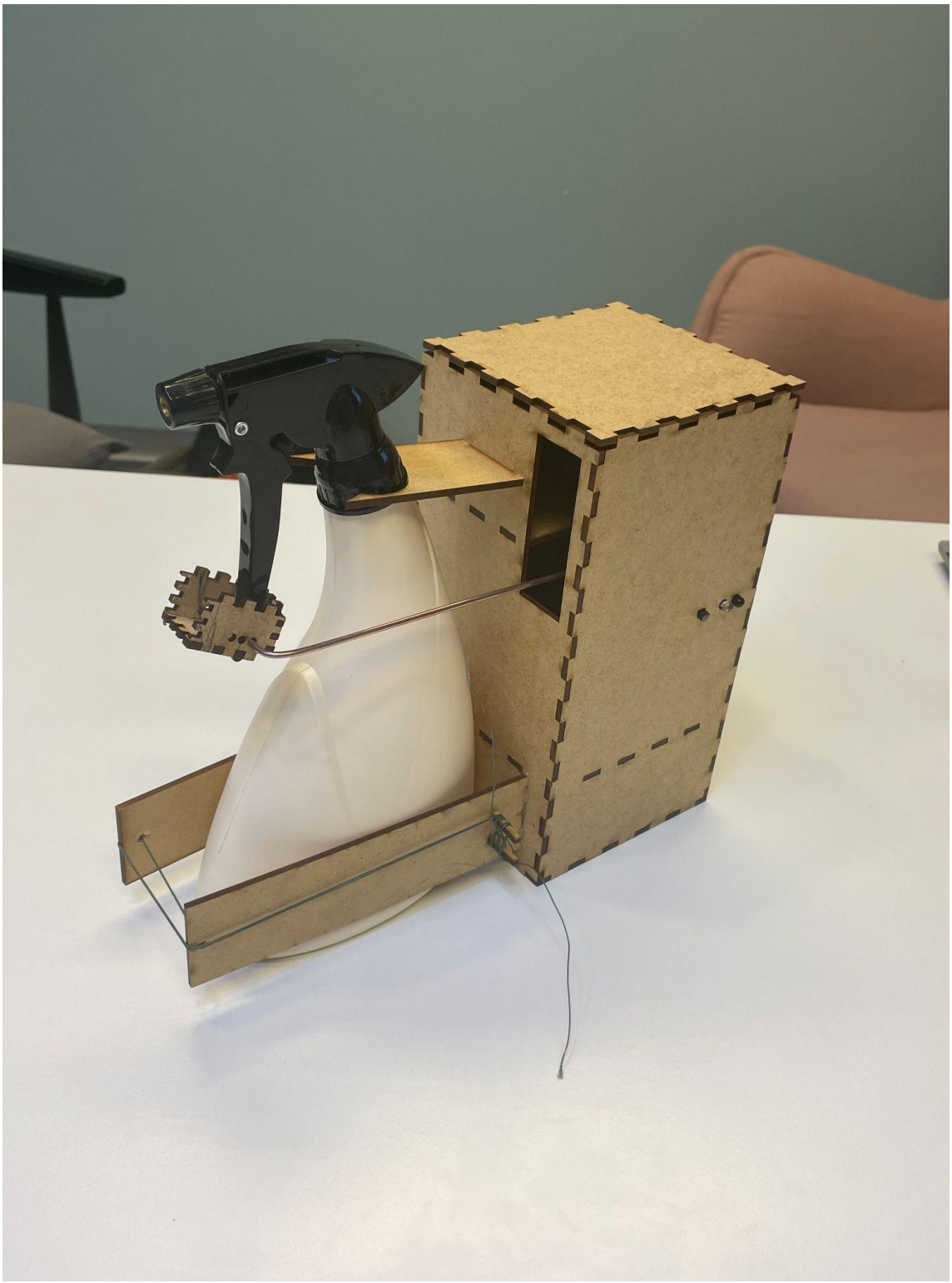
Principles of Digital Fabrication

Final Report – Group 20

The Spray Refresher

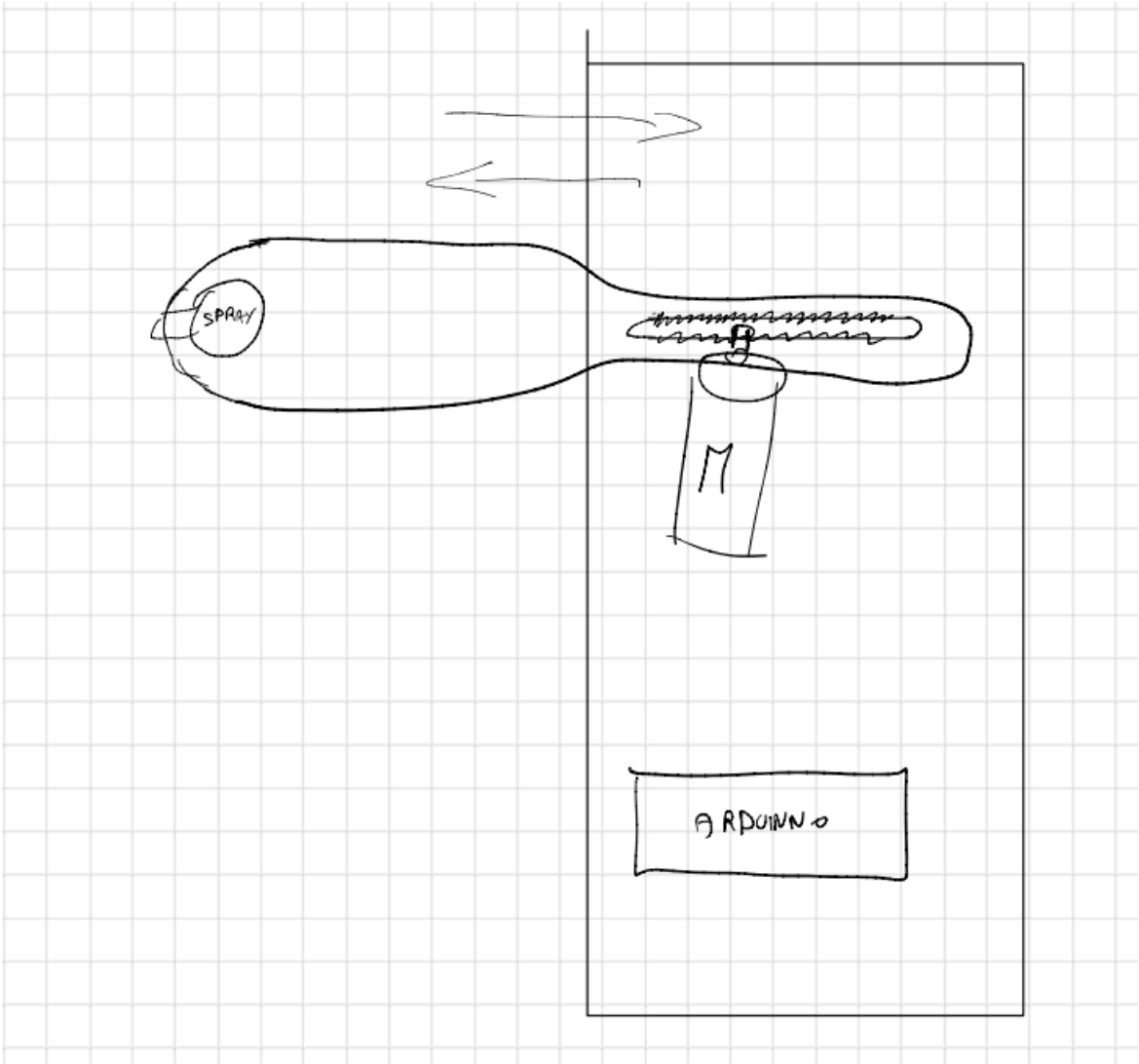


Introduction

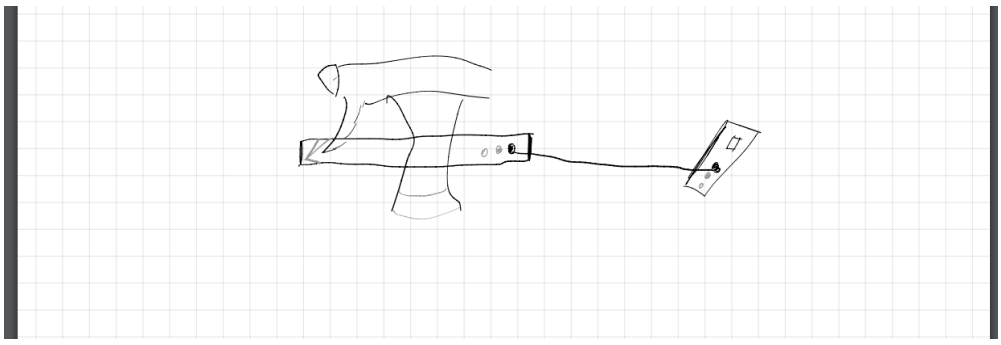
In this report, we are going to explain the process of developing our prototype and the difficulties we faced. Our idea was to develop a refresher gadget, which would have the following functionality: when the environment is warm (temperature over 25 degrees) and the user moves their hand in front of the device, a spray that revitalizes the client with fresh water is activated. Our goal was to make the project as sustainable as possible by avoiding the use of plastic and by designing from the beginning our prototype in such a way that works with every spray bottle.

First decisions

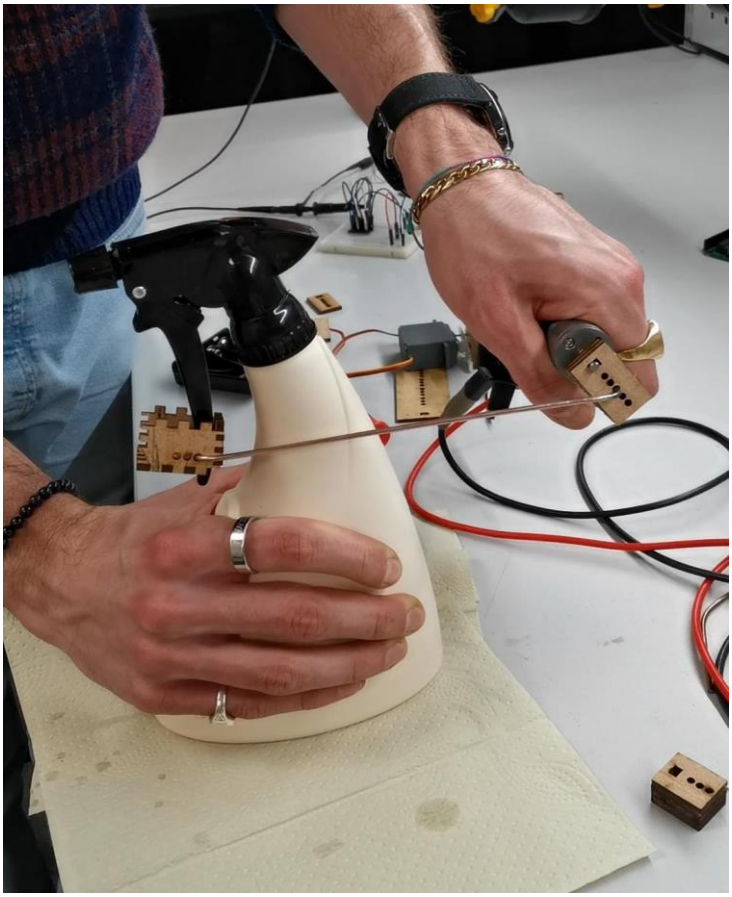
First of all, we started to sketch some ideas about the functionality of the gadget. We had two ideas about the spray mechanisms: the first one was to have a unique piece of material, one end of which would surround the spray bottle and on the other end the motor would move due to the friction, you can see it in more detail in the following image.



The second idea was to divide the mechanism into three pieces: the spray finger, a wire, and the motor mechanism. The spray finger is wedged under the bottle handle and the motor mechanism is also wedged in the motor's shaft, the wire connects the two, the spray finger and the motor mechanism. In the design process of this mechanism, we were inspired by a YouTube video (s://www.youtube.com/watch?v=WIimV5lZlQ4), in which a similar device is showcased. You can see the first sketch of the idea:



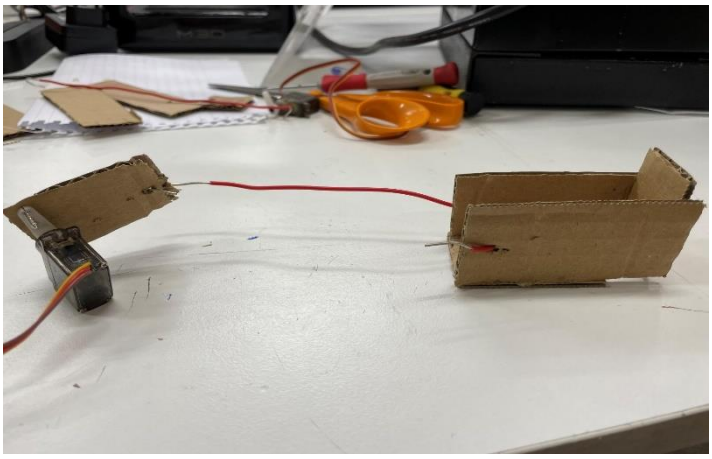
After a discussion, we decided to implement the second solution because the bottle handle would have more support with using the spray finger. Furthermore, in the second idea, the rotational movement of the motor mechanism causes a recurring linear movement of the wire between two extreme positions (periodic movement), which makes possible the spraying action. In the following picture you can see the result of our final design idea:



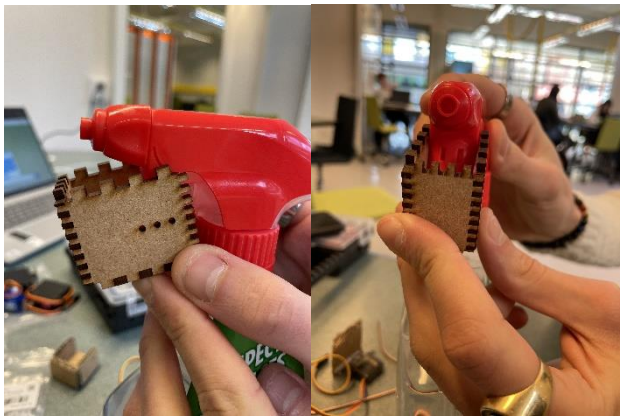
The Development

The finger mechanism

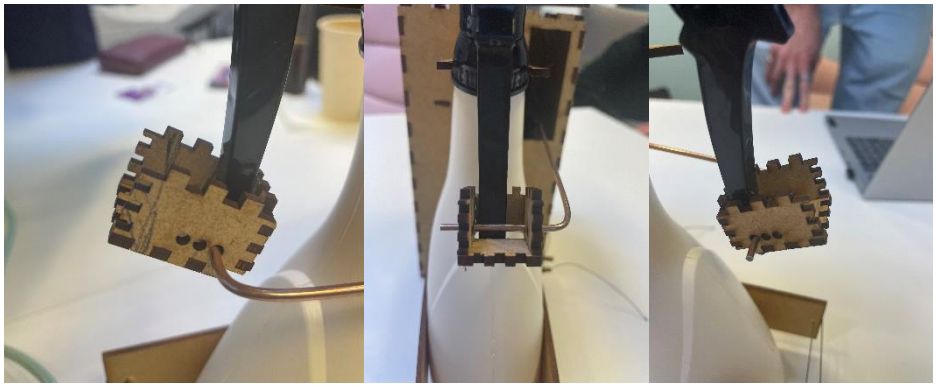
The development of our idea, as you can see in the photo, started with cutting cardboard boxes and pieces to simulate the finger mechanism.



We cut some plywood prototypes of the finger, and placed them in the following way:

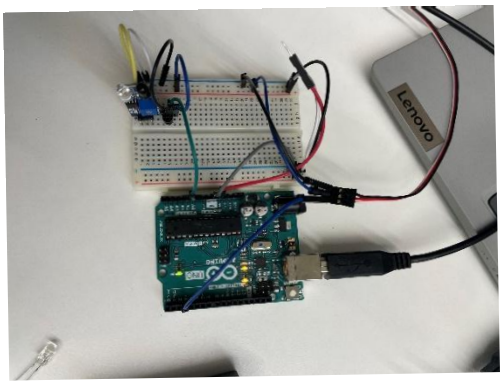


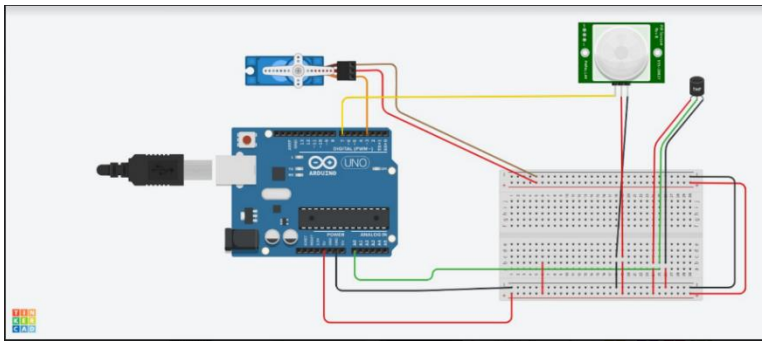
Then, we realized that the finger had to be put in the opposite orientation for two reasons: firstly, we did not want the finger bumping on the bottle, so there should be as less as possible material on the back of the handle; secondly, after a tutoring session with professor Georgi, we realized that if the wire was ahead of the spray handle, touching it, then the motor would require less effort to move the spray handle back. So, in the end, we designed our final finger prototype, which is the following:



The Motor

When we first started to think about our project and the finger mechanism, we decided to use a servo motor because it was easy to control the angle of movement. The servo would do the exact amount of movement that we needed for the finger mechanism to work (180 degrees). Here is the first Tinkercad circuit of our project:

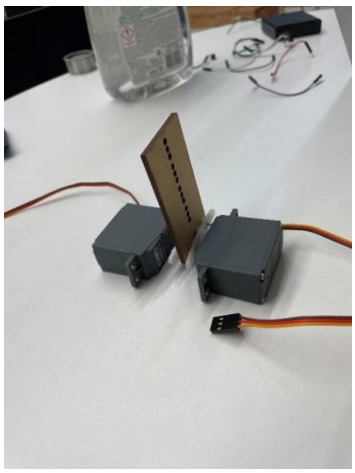


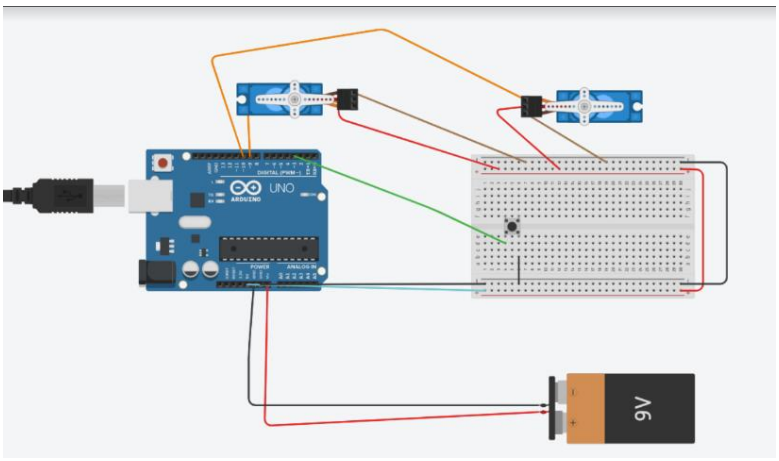


Regarding the Arduino code, we used the Servo.h library to control the servo. In the loop function, we first check if the motion sensor detects a movement and if yes, we check the temperature of the environment by converting the sensor's output into a temperature value. We first check the motion sensor so that if there is no movement, the microprocessor does not have to convert the temperature sensor's voltage uselessly.

```
#include <Servo.h>
Servo mainservo; //Servo


//Define temperature sensor pin to A0
#define tempPin A0
//Define motion sensor pin to 7
#define motionPin 7
//Define servo pin to 3
#define servoPin 3

int pos = 0; //Servo position in the beginning


void setup() {
  pinMode(tempPin, INPUT);
  pinMode(motionPin, INPUT);
  Serial.begin(400);
}

void loop(){

  if(digitalRead(motionPin) == HIGH){
    int test = 20;
     float temperature_value = analogRead(tempPin);     //Reads temperature from the pin (1 celcius correspond to 2millivolt)
     float temp = (temperature_value * 5.0) / 1024;    //Calculation to voltage
     float tempC = (temp - 0.5) * 100;      //Calculation to make temperature celcius
     Serial.println(tempC);
    if(tempC > test){ //If temperature is greater than given parameter and motion sensor does not detect movement

      mainservo.attach(servoPin);
      delay(200);
      mainservo.write(180);
      delay(2000);
      mainservo.write(0);
      delay(2000);
    }
  }else{
      mainservo.detach();
    }
}
```

When we started to test the mechanism, we noticed that the servo motor did not have enough torque to pull the spray handle back. We needed a solution to that problem, so we tried if the mechanism worked with two servos. We did the Tinkercad circuit to test how the two servos could move simultaneously and be synchronized.

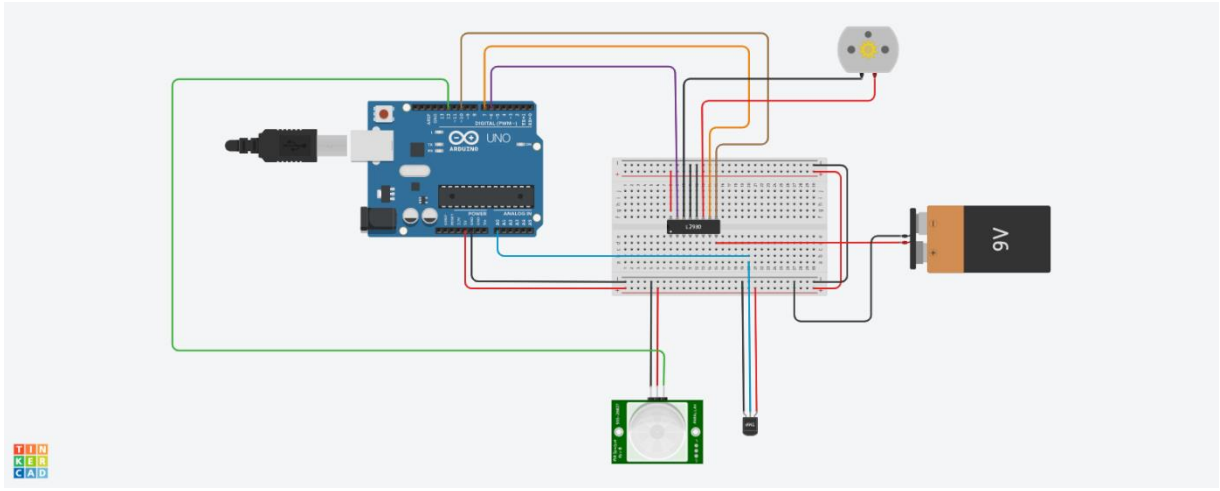




Two servos were not the solution to our problem, as their movements were not easily synchronized. After multiple tests and failures, we decided not to use servo motors at all, because they did not have enough torque to pull the spray bottle trigger.

We switched to the DC motor, which was given to us in the component box, as it was faster in the movement, and we thought that more speed could solve the problem. The difference with DC motor was that the movement could not be controlled by certain degrees like the servo one. We tested the DC motor by just connecting the pins to the battery, without Arduino. After setting up the finger mechanism, we realized that the torque of this motor was not enough to push the bottle handle back.

For that reason, we asked for a motor with more torque and we redesigned the Tinkercad circuit. Since the new motor could be supplied with a maximum of 12V we needed to use a driver (H-bridge) able to control it with Arduino. We also added to the circuit a switch in order to be able to turn on and off the whole device. In the following, you can see our final Tinkercad circuit:



Next, we needed to adjust the amount of time the DC motor would spin. After testing a little bit, we let the motor spin for two seconds because we noticed that corresponds to two sprays. In the following picture, you can find the Arduino code of this new circuit. In this case, we are using the analogWrite and digitalWrite functions to control the motor through the driver.

```
//Defining temperature sensor to A0
#define tempPin A0

//Defining motor control pins
//input_3 and input_4 decide motor direction
#define input_3 7
#define input_4 6
//Enabling the motor and PWM
#define enable 10
//defining motion sensor pin to 12
#define motionPin 12


void setup()
{
  pinMode(input_3, OUTPUT);
  pinMode(input_4, OUTPUT);
  pinMode(enable, OUTPUT);
}

void loop()
{
  if(digitalRead(motionPin) == LOW){  //If motion sensor detects motion
    float temperature_value = analogRead(tempPin);     //Reads temperature from the pin (1 celcius correspond to 2millivolt)
    float temp = (temperature_value * 5.0) / 1024;    //Calculation to voltage
    float tempC = (temp - 0.5) * 100;      //Calculation to make temperature celcius
    Serial.println(tempC);
    float test = 20;

    if(tempC > test){  // If temperature is greater than given parameter
      analogWrite(enable, 255);  // Spins motor with given speed up to 255
      digitalWrite(input_3, HIGH);
      digitalWrite(input_4, LOW); //If you switch input_4 HIGH and input_3 LOW the motor will spin the other way around

      delay(2000); //Spin for 2000 ms --> 2 seconds
      analogWrite(enable,0);

     }
  }else{ //else do nothing
    analogWrite(enable, 0); //Speed is 0 so motor does not spin
  }

}
```

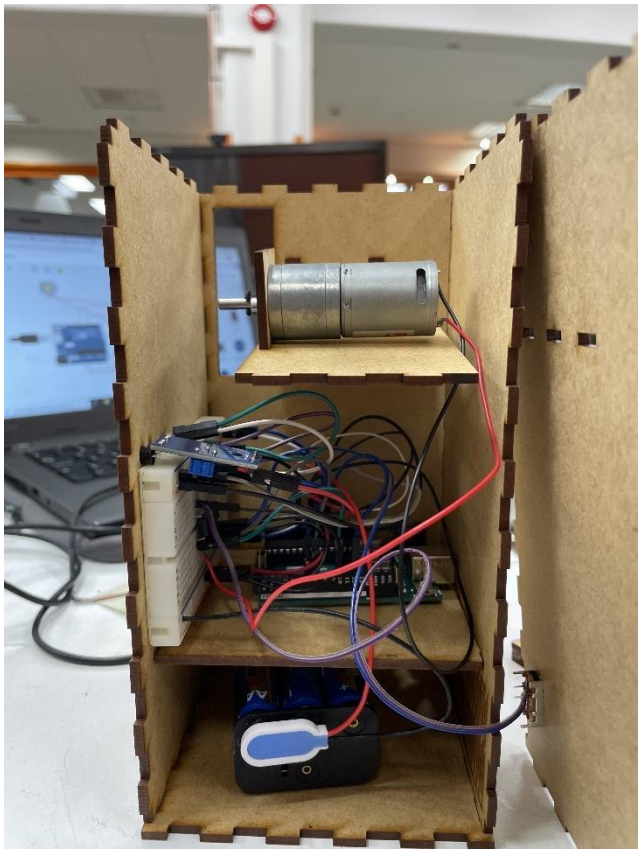
The Batteries

With the new motor, the 6V batteries weren't giving enough power, so the spray finger could not move. For that reason, we used the power supplier in the Electronics Workbench of FabLab, so we could measure how much power our motor needed. After doing some measurements, we concluded that our mechanism needed 9V power. We got 6 batteries of 1.5V, so we had a total of 9V power supplying our mechanism. After connecting the whole circuit with the Arduino and all the components, we had a lot of loss of power within the circuit so at the end of the circuit, to the motor, went only 6-7V of power. For that reason, we used new batteries which had their maximum capacity available.

After the battery problems were solved, we tested the new motor with the circuit, and it was able to spray, finally!

The Box of Components

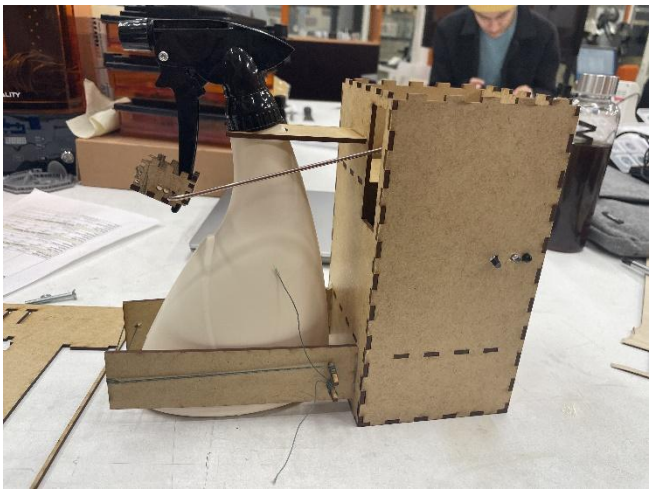
In the final stage of our development, we made the box of components. All the components have been built with the laser cutting machine and then assembled. In the box, all the electronic components are in three compartments: the battery on the bottom, the circuit on the middle shelf, and the motor on the top shelf. We also designed a door to access the batteries to be able to change them.



After assembling part of the box, we tested the whole gadget. We checked the temperature sensor by heating it up with a lighter and then the motion sensor by simply moving our hand in front of it. Thankfully, the gadget was working properly!

Bottle Holders

After having some trouble with the bottle moving, while the device was working, we elaborated more on our initial design by adding some pieces of plywood able to hold the bottle still. In the first stages of the development, our mechanism worked just by holding the bottle with our hands. So now, we attached to the upper part of the box of components a wood bottle handler, which stopped the linear movement of the bottle, and two wood stripes on the bottom part of the bottle connected through a string, which kept the bottle from moving sideways.



Finally, our gadget is composed of the following components:

- Spray bottle
- Box of components
- Bottle holders
- Spray finger
- Wire
- Motor mechanism (wood on shaft)
- Arduino Uno
- TMP36 sensor
- HW-201 IR motion sensor
- L293D driver
- 12V DC motor
- Six Battery Case
- Six 1.5v Batteries
- Switch slide SPDT
- Wires
- BreadBoard

Sustainability and Reusability

During the whole process of our prototype development, we had in mind the sustainability principles and requirements. First of all, we preferred using wood material for designing our finger and our box of components instead of plastic. Furthermore, we avoided form gluing the wood pieces together both for the design of the finger and the box. In this way, all the wood components of our prototype can be reused or recycled. Regarding the electronic parts of our prototype, we tested their functionality first as a simulation in the Tinkercad circuits, which resulted in being able not to burn or destroy any of the given equipment. All the electronic components of our prototype can be reused. Moreover, in order to have a gadget usable with every available spray bottle, we implemented the holding parts which can be resized.

What To Do Differently

After many discussions with our professors, we understood some changes that we could have done during our development process in order to make it easier. As we were trying to simulate the internal mechanical movement of the spray bottle mechanism, we could have investigated and analysed the internal structure of the mechanism: how much force the bottle handle needs to be pushed back, and how much torque it needs. It would be a good idea to simulate the spray handle movement with a spring and measure its extension during the movement of the spray handle (Hooke's law).

Feedback

All of us have found the course interesting, we enjoyed the fact that with few machines we were able to design everything that we needed for the gadget. The idea to put into practice the theory notions explained in the first week has been challenging as we had no electrical and mechanical engineering background. We think that the tutoring sessions with the course staff were helpful: the continuous feedback from them encouraged us to continue and resolve any struggles or doubts that we had during the development process. Finally, the trial-and-error iterative approach has been challenging but we have learnt how hard the project development process can be if not organised and managed correctly.