TP MOSH - BENAZZOUZ - HEBRAS

TP1A : My first program : Blink

Un script nous est fourni :

```
int ledPin =  13;    // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup()   {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH);   // set the LED on
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // set the LED off
  delay(1000);                  // wait for a second
}
</div></pre>
```

Il nous permet de faire clignoter la LED pin 13, cependant on utilise un delay qui est très contraignant dans un code plus complexe, car il bloque l'exécution, cad que rien d’autre ne s’effectue pendant ce temps la

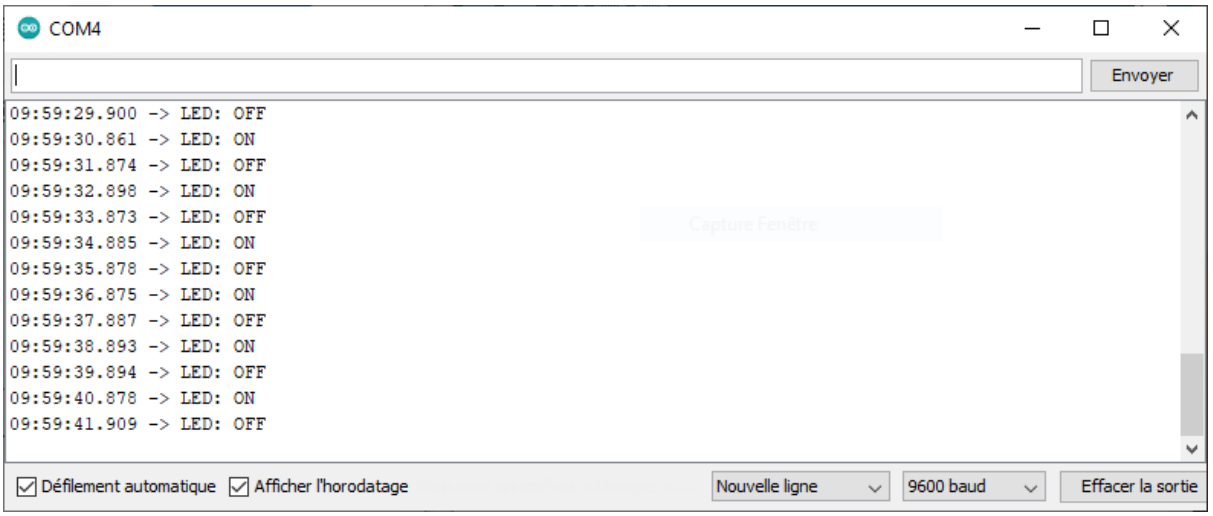
TP1B: Blink without delay

une autre méthode pour faire le blinking est en mesurant le temps passé entre deux changements d’état, pour cela on initialise deux variables currentMillis et previousMillis et on calcule la différence entre les deux. le currentMillis est déterminé grâce à la fonction “Millis()” qui renvoie le temps écoulé depuis que la carte a commencé à fonctionner.

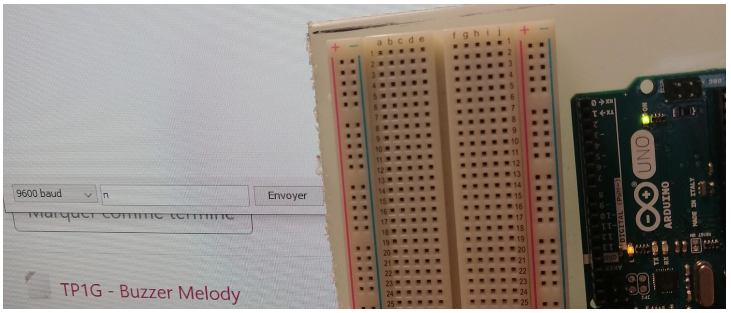
```
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }
```

TP1C : Serial Monitor (Arduino => PC)

On observe :

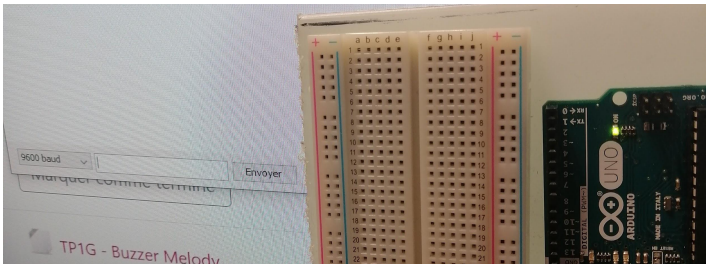


TP1D : Serial Monitor (PC => Arduino)



étape 1 : la led est allumée, on tape “n”

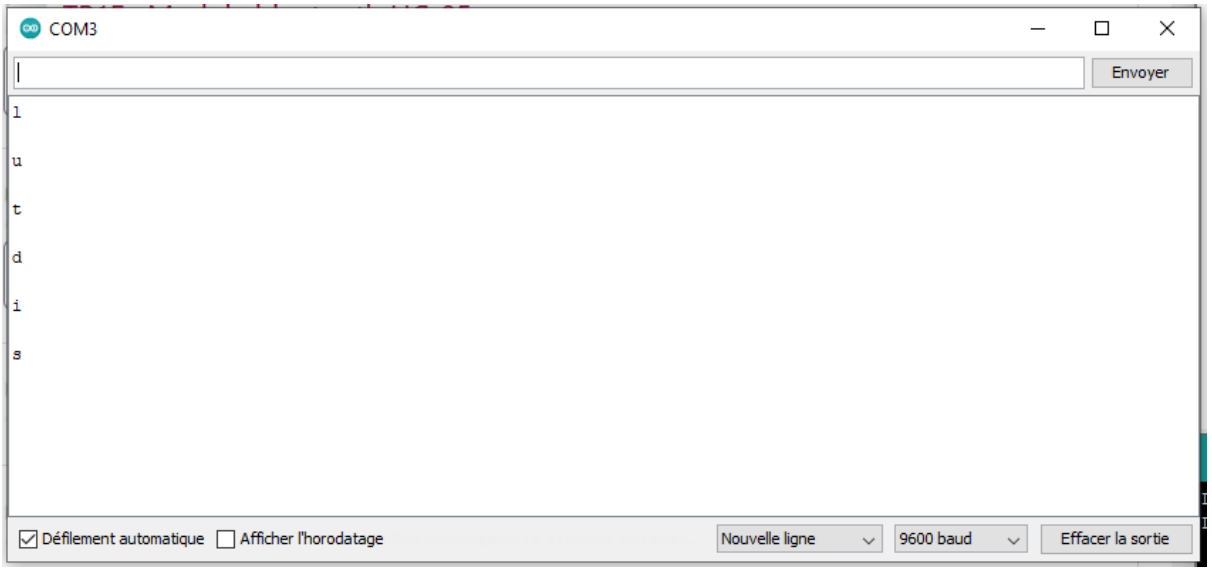
étape 2 : on tape “entrer”

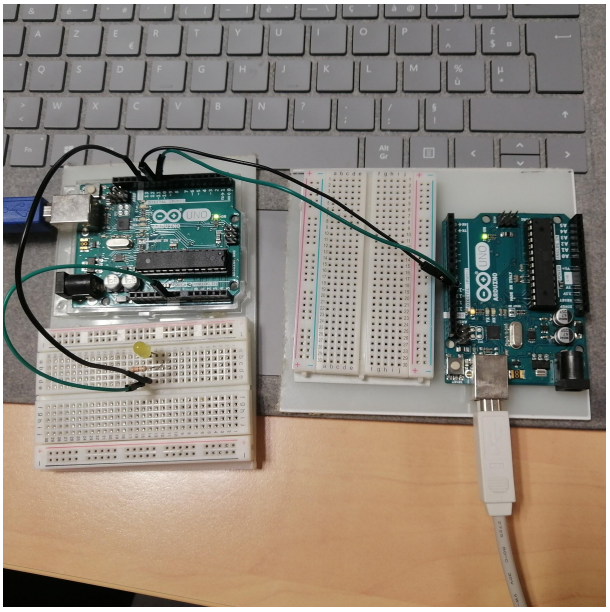


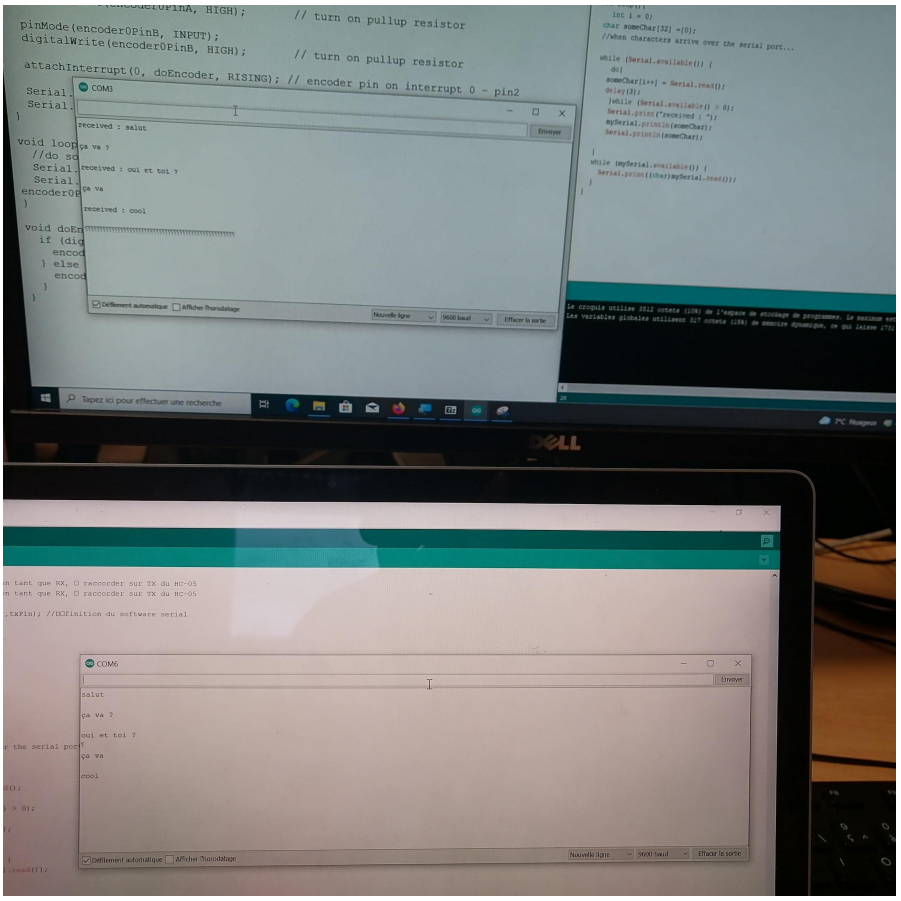
étape 3 : la led s’éteint

Remarque: les led RX et TX s’allument respectivement à l’envoi du PC vers la carte et de la carte vers le PC.

TP1E : Serial Monitor

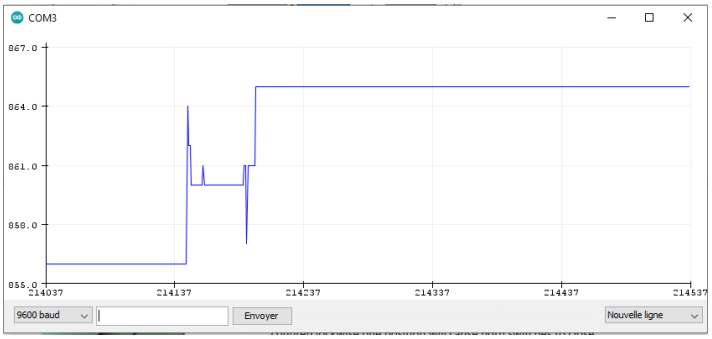





pinMode(encoder0PinB, INPUT);
digitalWrite(encoder0PinB, HIGH);
// turn on pullup resistor
attachInterrupt(0, doEncoder, RISING); // encoder pin on interrupt 0 - pin2
COM3
Envoyer
received : salut
ça va ?
received : oui et toi ?
ça va
received : cool
Défilement automatique
Afficher l'horodatage
Nouvelle ligne
9600 baud
Effacer la sortie
Tapez ici pour effectuer une recherche
COM6
salut
ça va ?
oui et toi ?
ça va
cool

TP1F: Encodeur rotatoire

En faisant tourner l'encodeur, on observe la variation de notre variable. Cependant, des rebonds ajoutent du bruit au signal. Pour les enlever, nous pourrions ajouter un filtre passe-bas hardware.



TP1G: Buzzer melody

Here we send a periodic signal to a buzzer, by varying the frequency we make different notes.
note: it resembles a PWM with a 50% ratio but with a varying frequency, whereas in a PWM the ratio is varying and the frequency is constant.

TP1H : Temperature Sensor

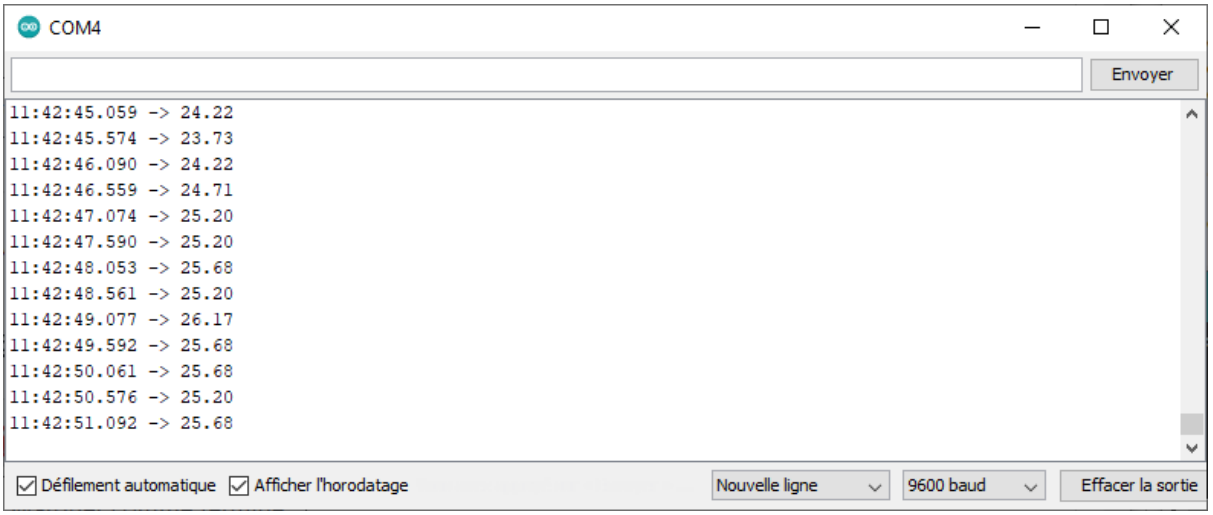
The getVoltage function returns a float, that is then stored in the temp variable

```
int tempPin = 0;
void setup(){
  Serial.begin(9600);
}

void loop(){
  float temp = getVoltage(tempPin);
  temp = (temp-0.5)*100;
  Serial.println(temp);
```

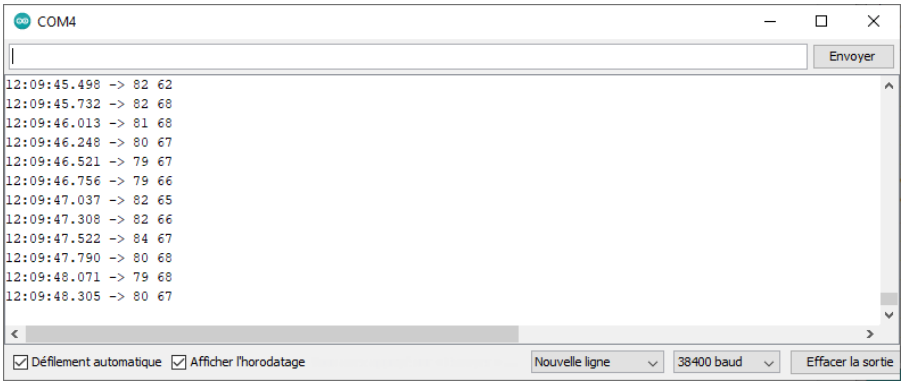
```
  delay(500);
}
float getVoltage(int pin){
  return(analogRead(pin)*.004882814);
}
```

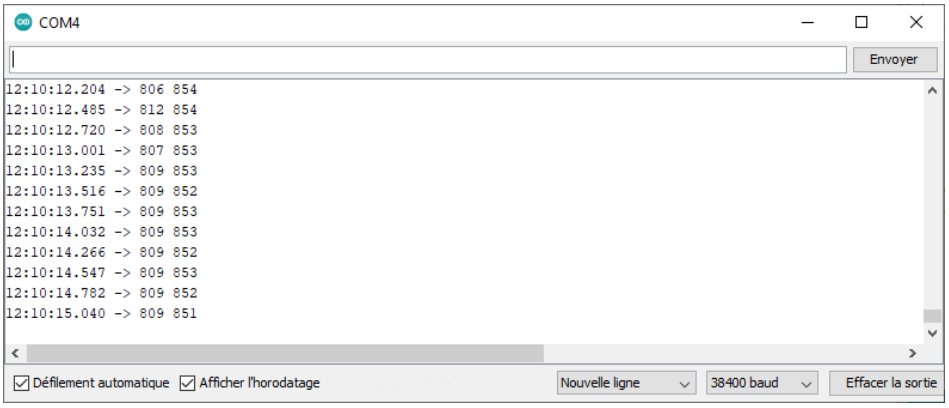
en posant le doigt sur le capteur, on observe l'évolution sur le moniteur série



TP2A : Touchpad DS

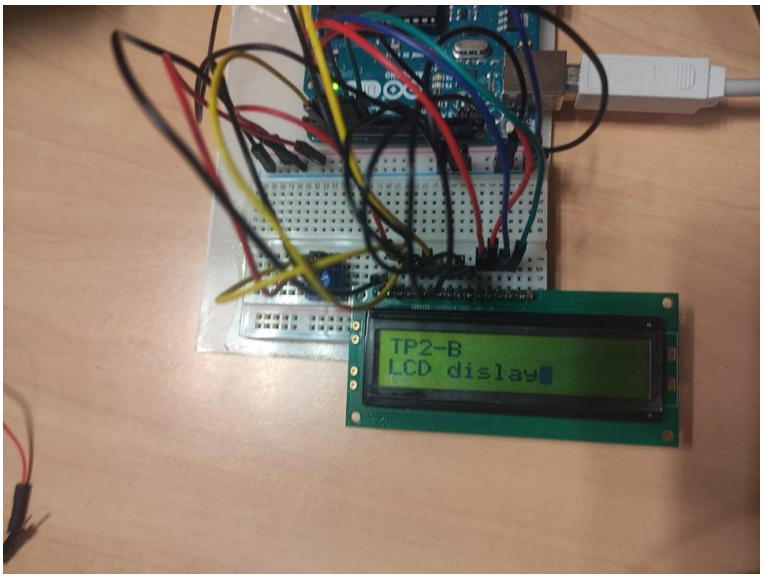
Here we code the reading of a touchpad, this is done by reading two tension divider, the two values are read on the analog ports and are scaled between 0 and 1000 (from 0 to 5V). in parctice those values are never reach, but we see very well the variation from a corner to an other.
for instance, this is the value when nothing is touched, and when we touch the bottom left corner:





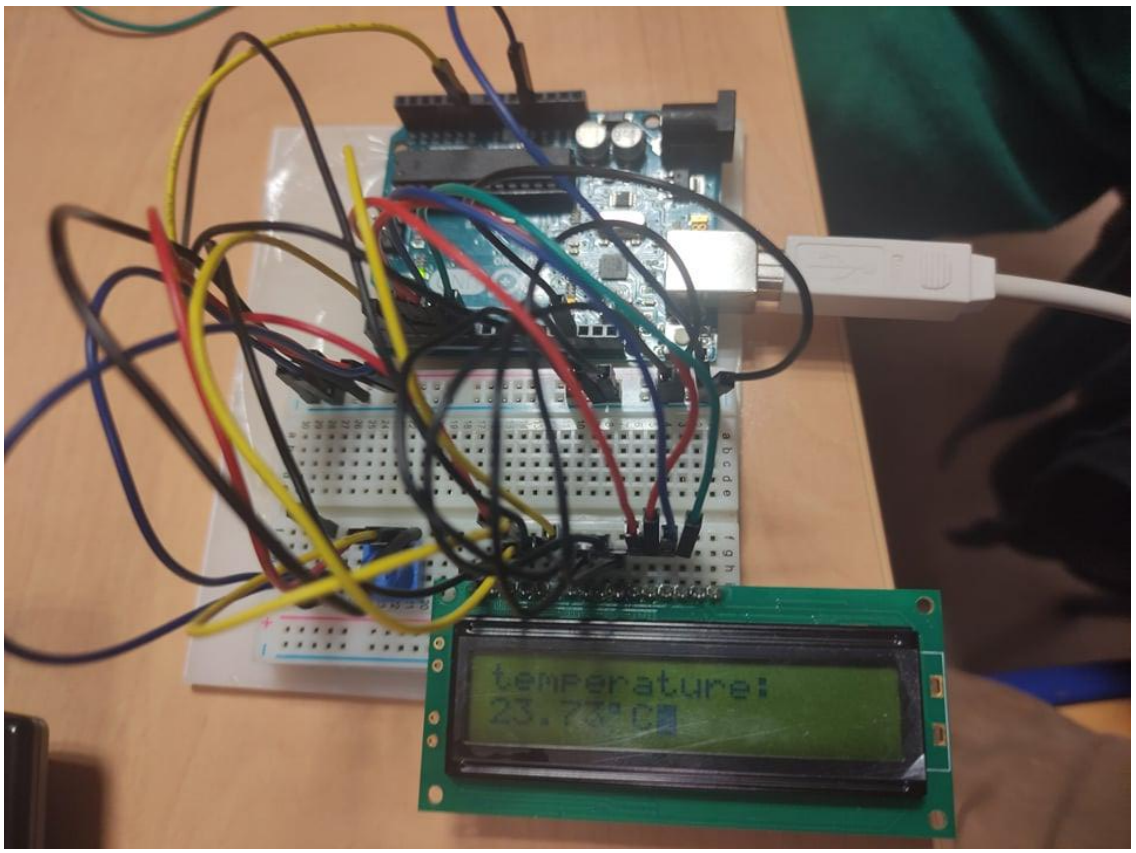
note, when we don't touch the screen a remaining tension is detected, we could correct that by fixing a threshold above the remaining value.

TP2B,C,D : LCD



La fonction blink permet de faire clignoter le curseur, utile si on souhaite que l'utilisateur puisse taper des choses (pour se s'en servir comme terminal par exemple)

TP2-B: lcd et température

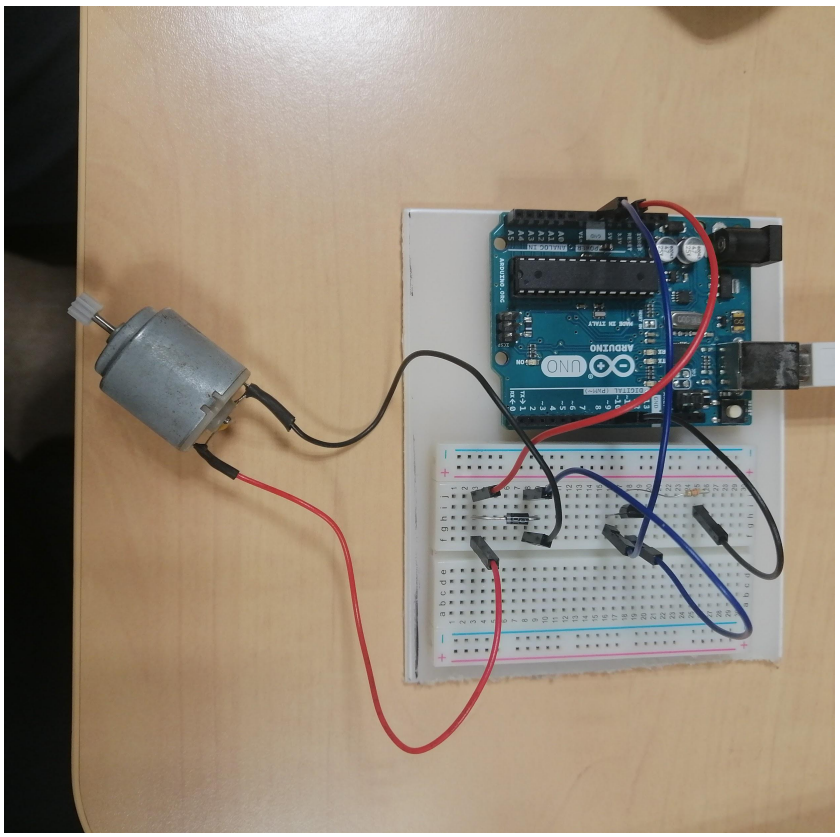


(NB : pour afficher le caractère “degré” : stocker dans une variable de type char la valeur en binaire du caractère et printer cette variable)

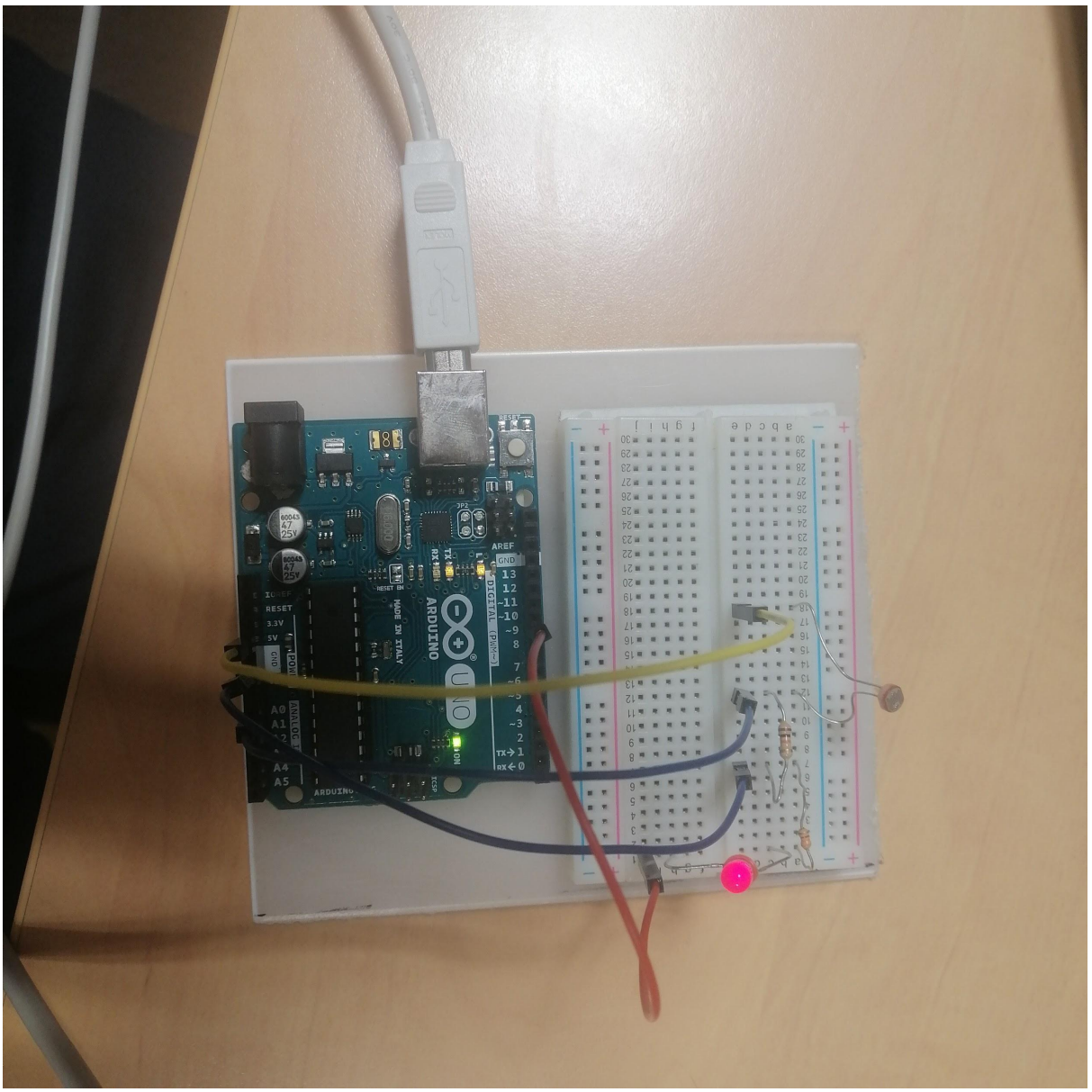
TP2-C: lcd et touchpad

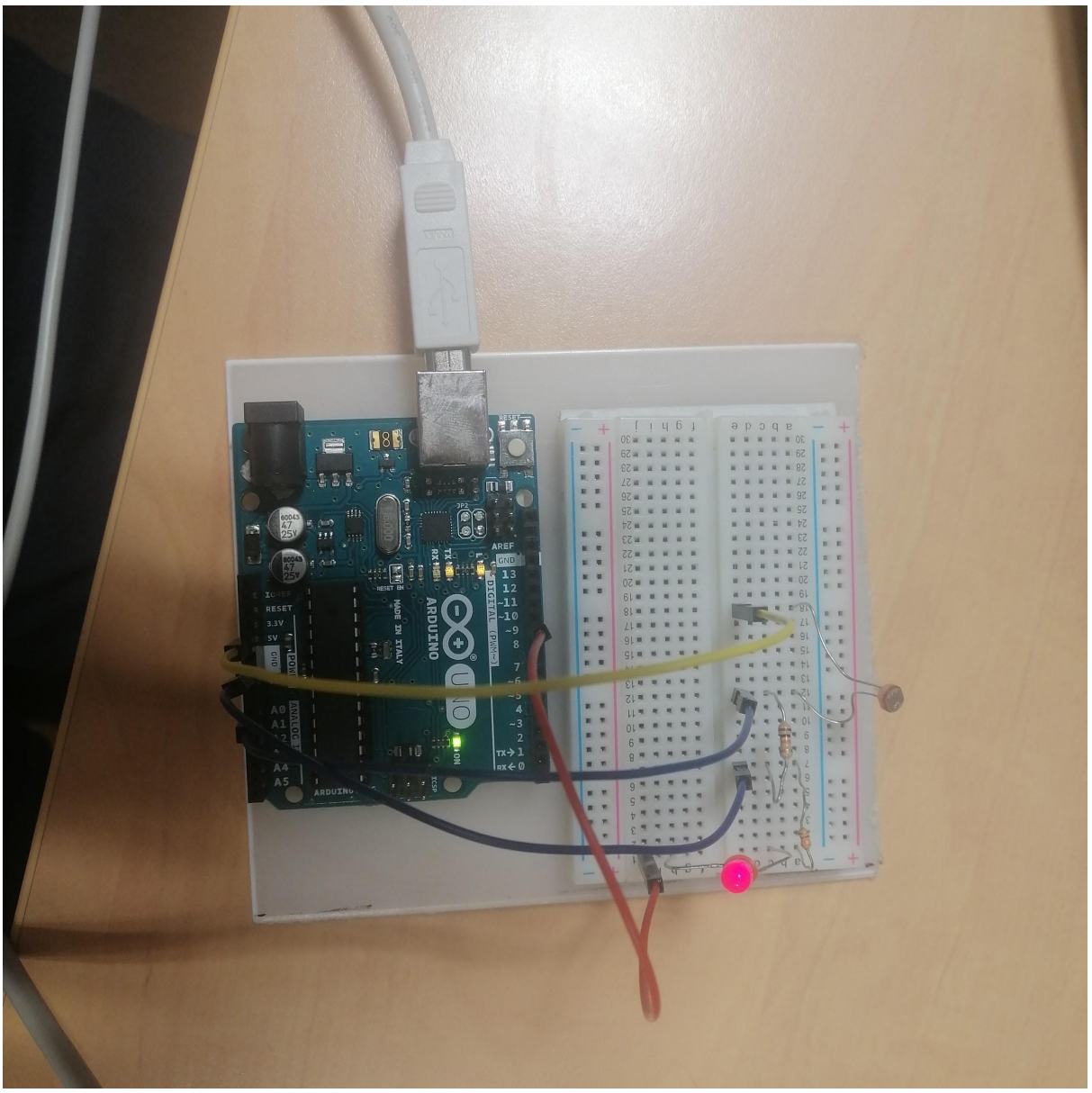
In this example we notice the use of the function lcd.scrollDisplayLeft, which allow us to display scrolling text, useful when the message is longer than what the lcd screen can display at once.

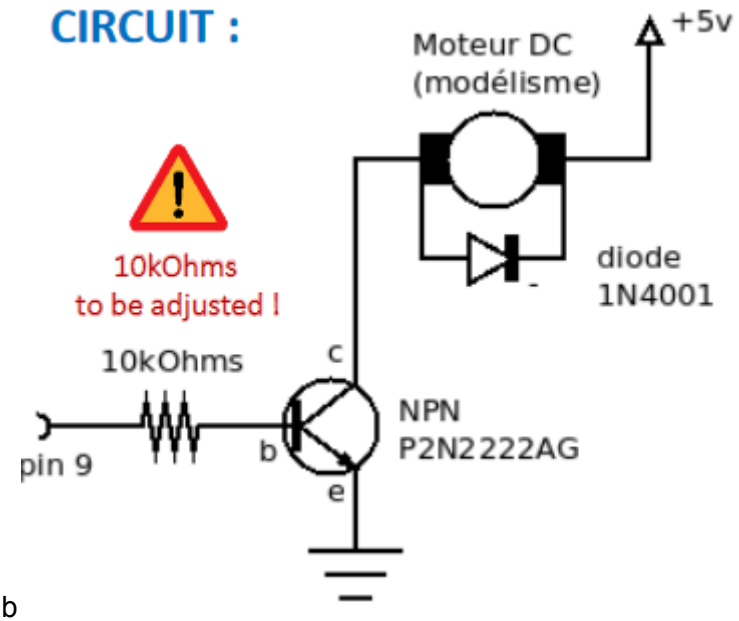
TP3A : Moteur DC



Lorsqu’aucun courant ne passe dans le moteur, pour que la tension n’explose pas, une diode est ajoutée au circuit.

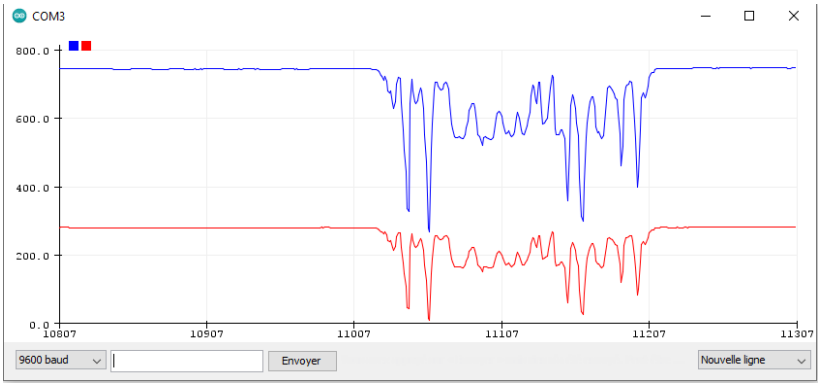

ARDUINO
UNO
MADE IN ITALY
DIGITAL (PWM~)
ANALOG IN
POWER
RESET
3.3V
5V
GND
VIN
AREF
ON
TX
RX

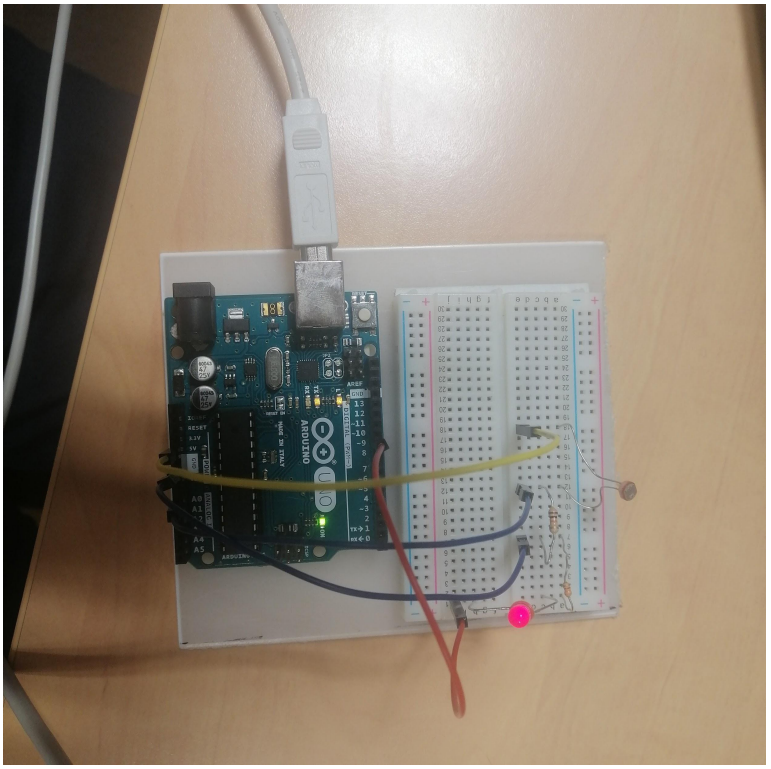



CIRCUIT :
Moteur DC
(modélisme)
+5v
10kOhms
to be adjusted !
diode
1N4001
10kOhms
c
NPN
P2N2222AG
pin 9
b
e

Bla kab

TP3B : Photorésistance

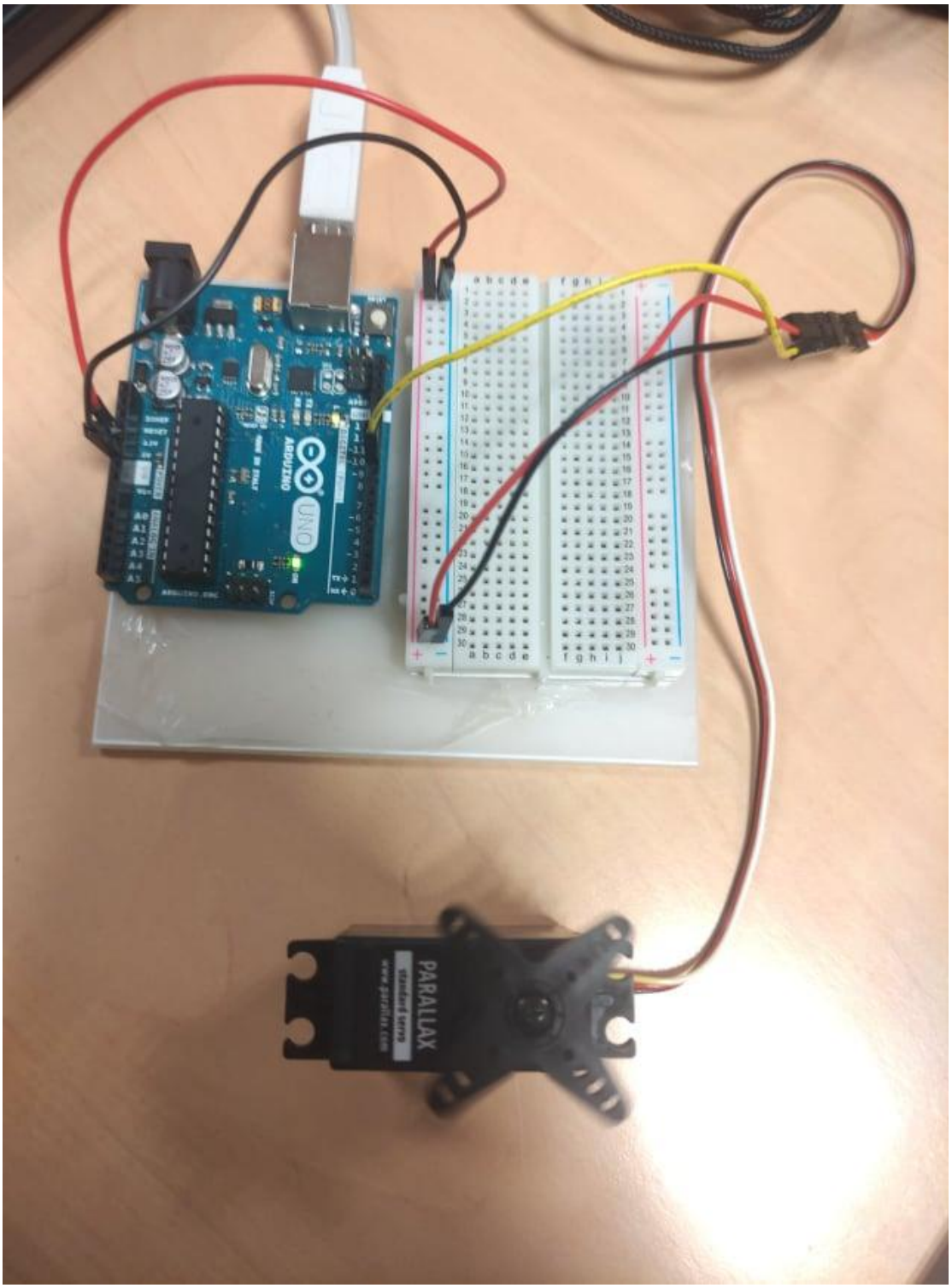




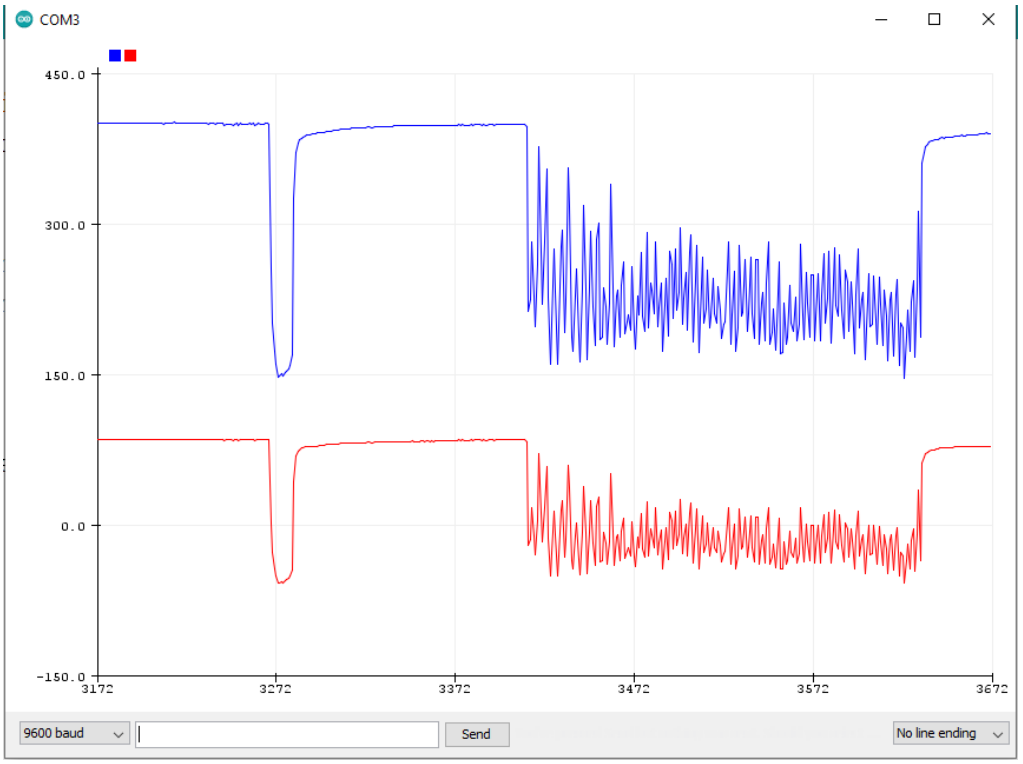
En approchant et en éloignant le doigt de la photorésistance, on peut moduler la tension reçue et la valeur de luminosité observée. Lorsque la luminosité mesurée baisse, la luminosité sortant de la diode baisse aussi.

TP3C : servo-motor

here we use the Servo library, which is super easy and intuitive to use to use.



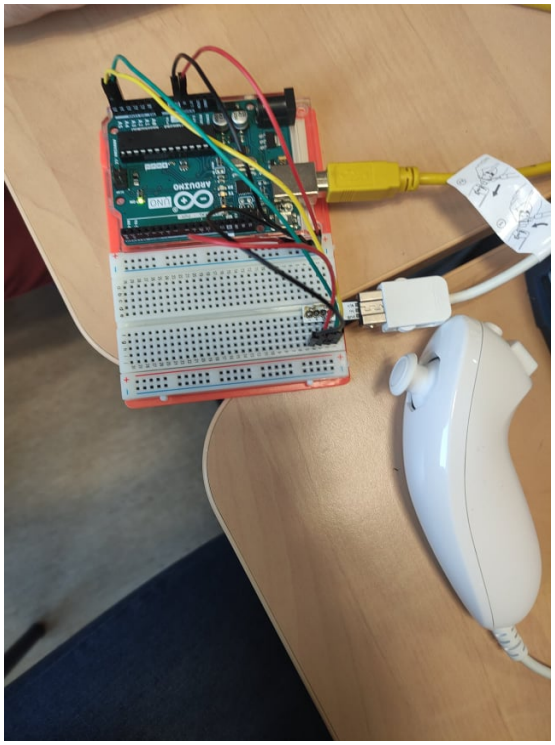
TP3D : FLEX sensor

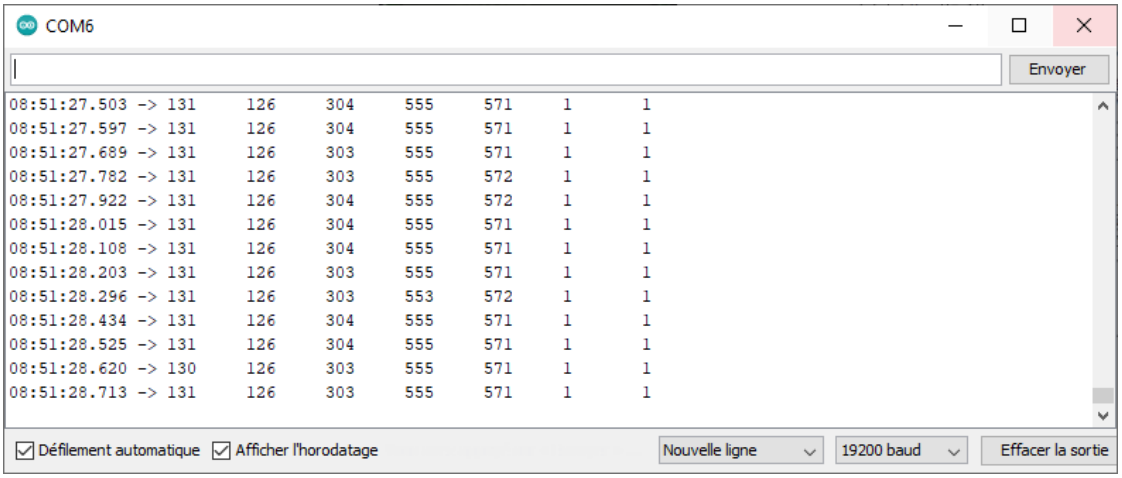


Un pont diviseur de tension permet de récupérer la tension aux bornes du capteur du flex sensor

TP3F: Interruptions

TP Nunchuck

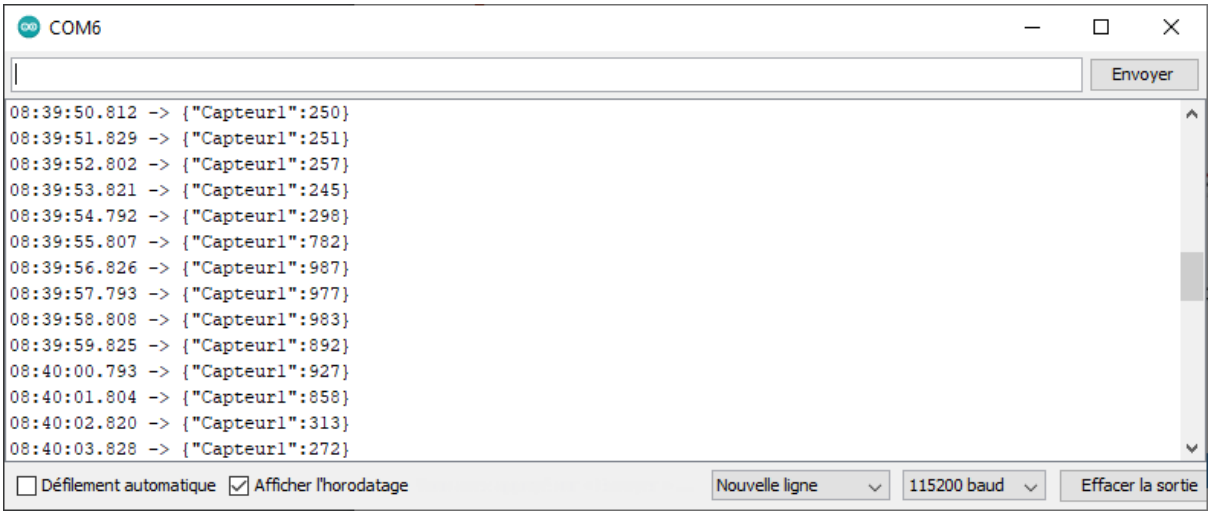




---

en raison de manque de temps, nous passons directement au TP7, node-red est un outils d'intérêt pour nous puisque nous l'utilisons dans d'autres projets

→ test capteur gaz



Nous constatons qu’effectivement la valeur mesurée par le capteur de gaz augmente quand on met le capteur en présence d’une vanne de briquet.