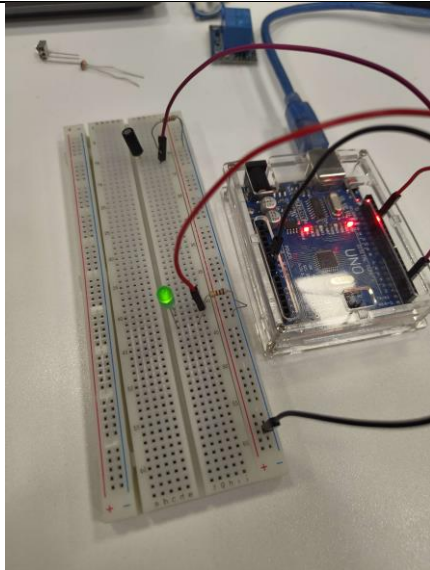


Dutrannois Esteban

Dubois Théo

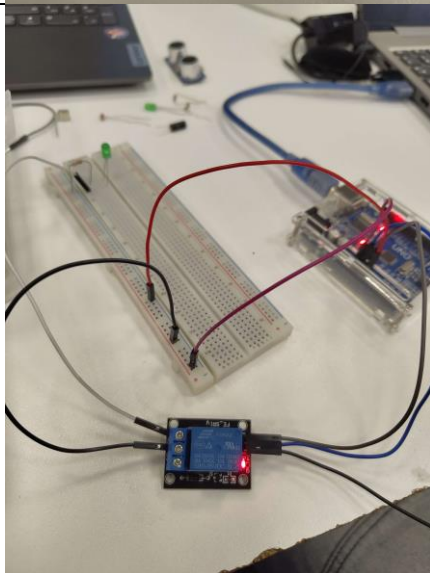
Labo 4 : Sensors



```
#define SHOCK_PIN 2

void setup() {
  pinMode(LED_BUILTIN, OUTPUT); // on-board LED, usually pin 13
  pinMode(SHOCK_PIN, INPUT);    // shock sensor pin set to input
}

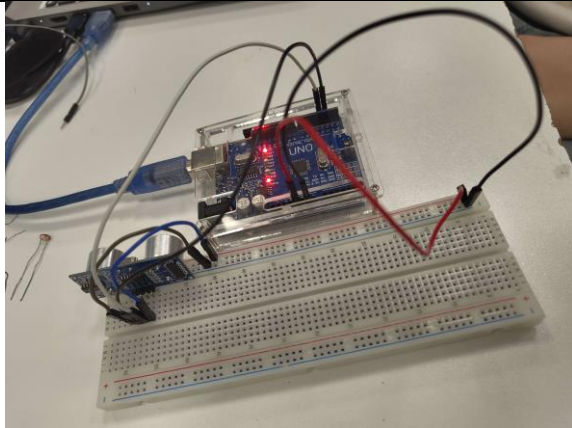
void loop() {
  if (digitalRead(SHOCK_PIN)) { // shock detected?
    // shock detected with pull-down resistor
    digitalWrite(LED_BUILTIN, HIGH); // switch LED on
    delay(2000);                     // leave LED on for period
  }
  else {
    // shock not detected with pull-down resistor
    digitalWrite(LED_BUILTIN, LOW);  // switch LED off
  }
}
```



```
int relayPin = 2; // broche utilisée pour contrôler le relais

void setup() {
  pinMode(relayPin, OUTPUT); // définit la broche du relais en sortie
}

void loop() {
  digitalWrite(relayPin, HIGH); // allume le relais
  delay(1000); // attend 1 seconde
  digitalWrite(relayPin, LOW); // éteint le relais
  delay(1000); // attend 1 seconde
}
```

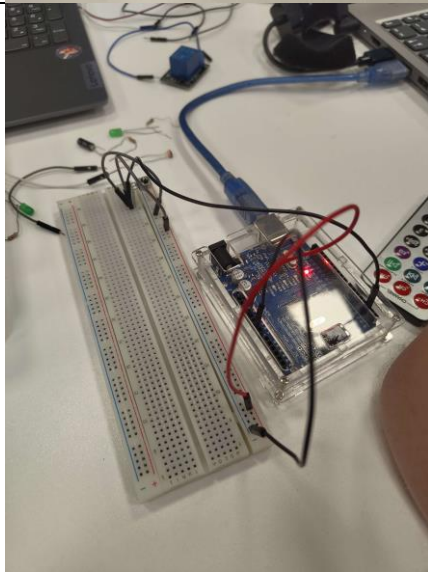


```
// Définition des numéros de port
const int trigPin = 3; // Trigger (émission)
const int echoPin = 2; // Echo (réception)

// Variables utiles
long duree; // Durée de l'écho
int distance;

void setup() {
  pinMode(trigPin, OUTPUT); // Configuration du port du Trigger comme une SORTIE
  pinMode(echoPin, INPUT); // Configuration du port de l'Echo comme une ENTREE
  Serial.begin(9600); // Démarrage de la communication série à 9600 bits/s
}

void loop() {
  // Emission d'un signal de durée 10 microsecondes
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Ecoute de l'écho
  duree = pulseIn(echoPin, HIGH);
  // Calcul de la distance
  distance = duree * 0.034 / 2;
  // Affichage de la distance dans le Moniteur Série
  Serial.print("Distance en cm : ");
  Serial.println(distance);
  delay(500); // Délai d'attente pour éviter d'afficher trop de résultats à la seconde
}
```



ereceiver.ino PinDeMissionsArduino.fr

```
#include <IRremote.h>

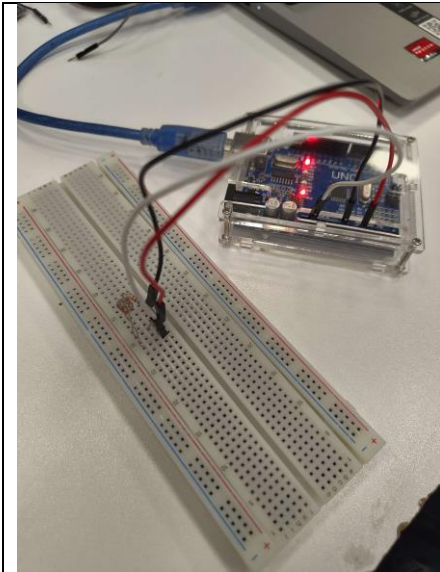
const char recepteurIR = 2;

IRrecv monIR(recepteurIR);

decode_results message;

void setup()
{
  Serial.begin(9600);
  monIR.enableIRIn();
}

void loop()
{
  if(monIR.decode(&message) == "FFFFFFFF")
  {
    Serial.println("Noise");
  }
  if (monIR.decode(&message))
  {
    Serial.println(message.value, HEX);
    delay(500);
    monIR.resume();
  }
  delay(1);
}
```



```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  // reads the input on analog pin A0 (value between 0 and 1023)
  int analogValue = analogRead(A0);

  Serial.print("Analog reading: ");
  Serial.print(analogValue);  // the raw analog reading

  // We'll have a few thresholds, qualitatively determined
  if (analogValue < 10) {
    Serial.println(" - Dark");
  } else if (analogValue < 200) {
    Serial.println(" - Dim");
  } else if (analogValue < 500) {
    Serial.println(" - Light");
  } else if (analogValue < 800) {
    Serial.println(" - Bright");
  } else {
    Serial.println(" - Very bright");
  }

  delay(500);
}
/*
```