

SAMTEST



BUNKER Ilan, CASSEGRAIN Lisa, DANI Thibaut

# SOMMAIRE



- OBJECTIFS
- DIFFERENTS BLOCS
- PLAN D'EXECUTION
- MATERIEL UTILISE
- CE QU'ON AURAIT PU FAIRE

# OBJECTIFS DU PROJET

□ Rappelons le :

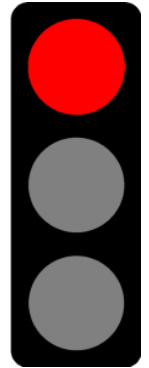
Conduite sous l'emprise d'alcool

= 1<sup>ère</sup> cause de mortalité sur les routes

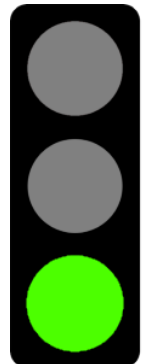


# OBJECTIFS DU PROJET

Si le bruit retentit + LED rouges :  
taux d'alcoolémie > seuil autorisé

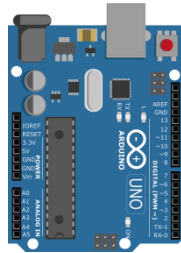


Si le bruit ne retentit pas + LED vertes :  
Test négatif



# MATERIEL UTILISE

Carte Arduino  
pro mini



Imprimante 3D  
Ultimaker 3



Tube à utiliser lors  
du soufflement

Boîte Samtest couvrant la  
carte et les  
branchements



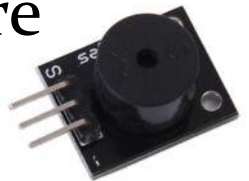
Fils, LED



Capteur MQ135 : Module  
de détection de gaz,  
qualité de l'air



Buzzer d'alarme (sonore  
continu)

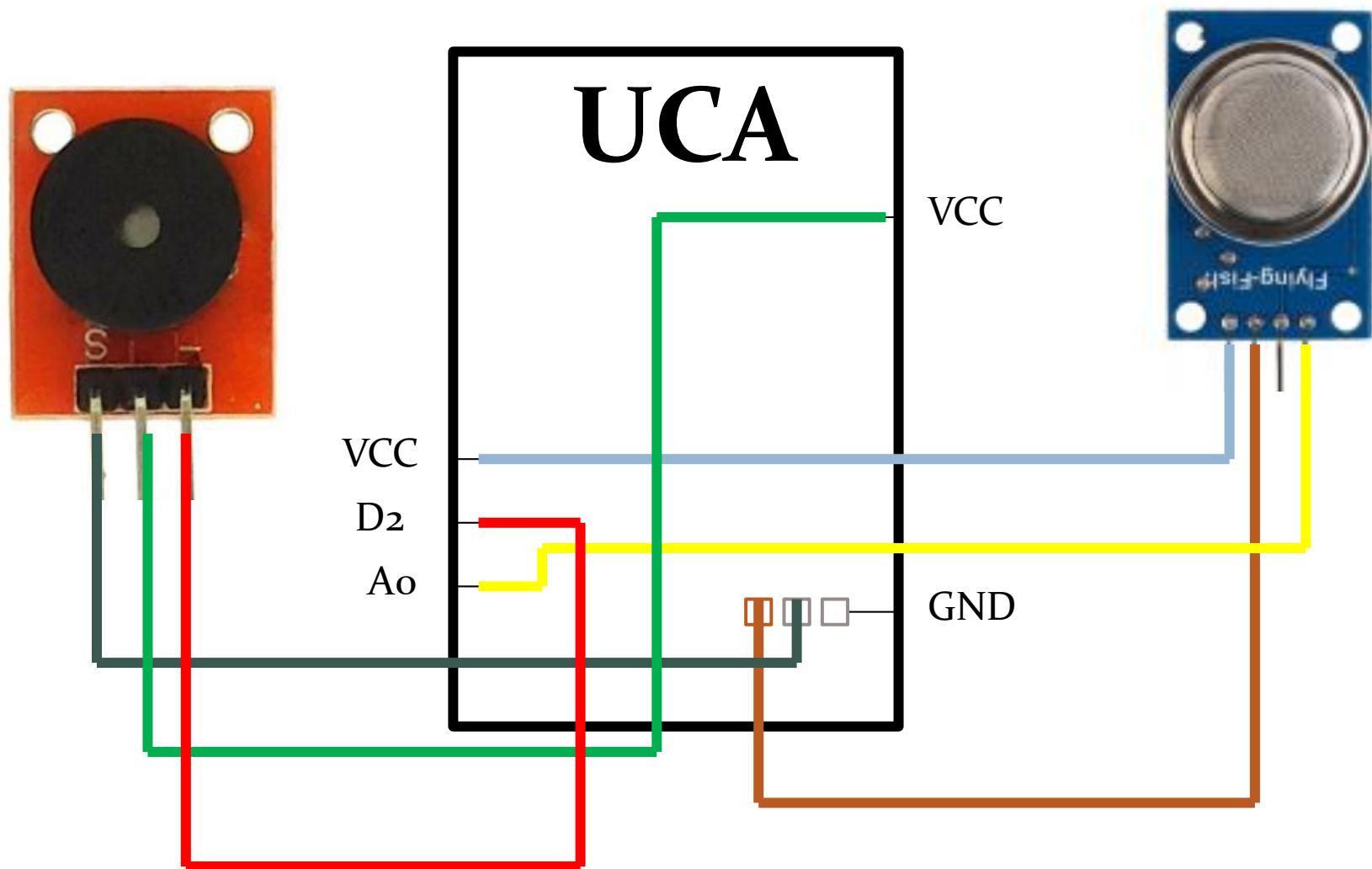


# DIFFERENTS BLOCS FONCTIONNELS DU PROJET

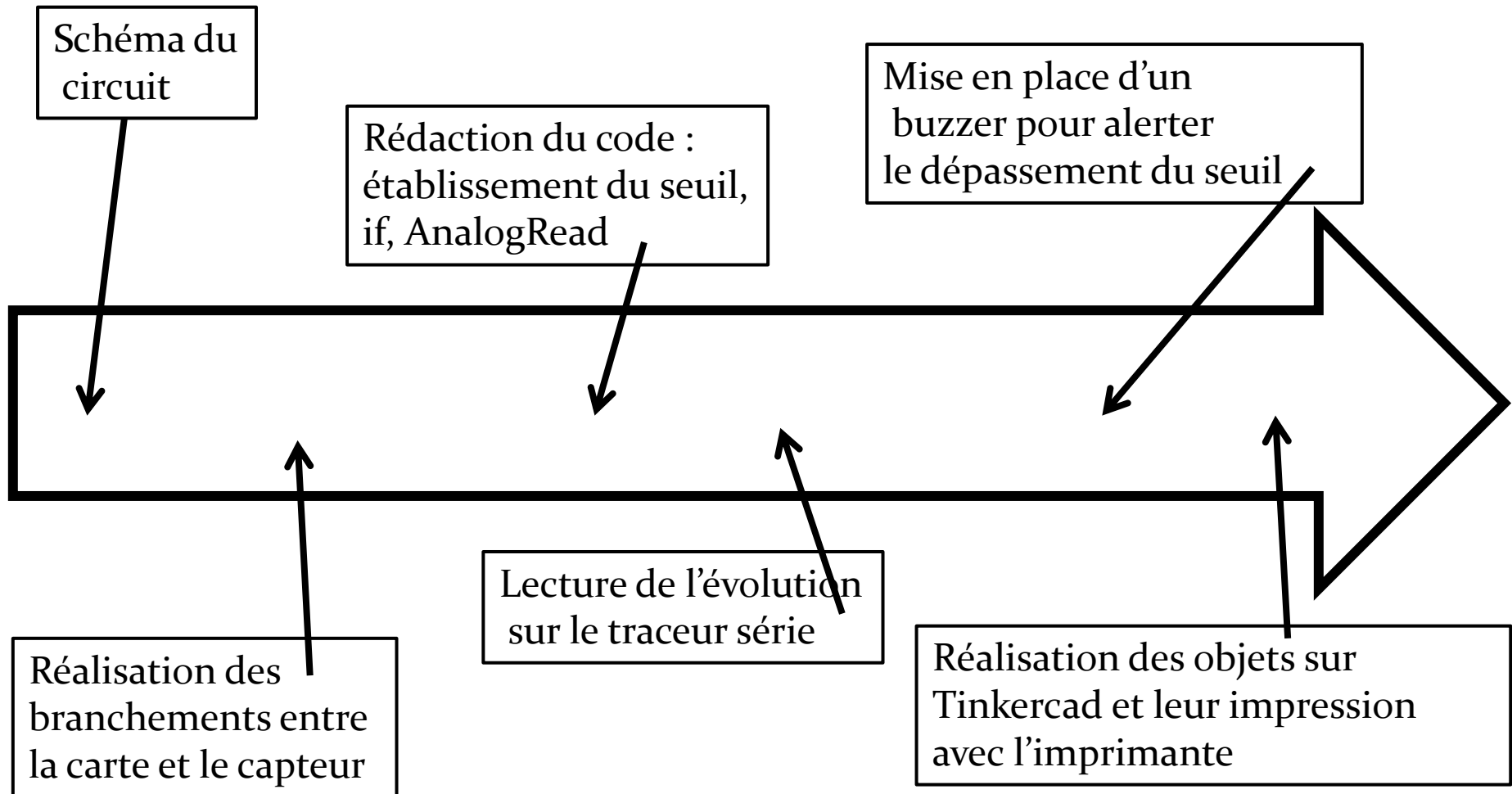
- Le capteur d'air est relié à la carte Arduino par trois branchements :
  - Un relié à l'entrée Analog In Ao
  - Un relié à l'alimentation 3.3V
  - Le dernier relié au GND de l'Arduino
- Le buzzer est relié à la carte Arduino par trois branchements également :
  - Un relié au GND
  - Un relié à l'alimentation 3.3V
  - Le dernier relié à la sortie Do



# DIFFERENTS BLOCS FONCTIONNELLS DU PROJET



# PLAN D'EXECUTION





# PLAN D'EXECUTION

## SAMTEST

```
#include "FastLED.h"
#include "LowPower.h"
#include <LoRa.h>

#define NUM_LEDS 9

// Data pin that led data will be written out over
#define DATA_PIN 4
// Clock pin only needed for SPI based chipsets when not using hardware SPI
// #define CLOCK_PIN 8

CRGB leds[NUM_LEDS];
int lum = 63;
int pin8 = 8;
int sensor = A0;
int sensorValue = 0;
int buzzer = 2; // Borne de sortie
int debut = 0;
int sensorReference = 0;
int valeur = 0;

void setup() {
  // sanity check delay - allows reprogramming if accidentally blowing power w/leds
  delay(2000);

  // Uncomment one of the following lines for your leds arrangement.

  FastLED.addLeds<WS2812, DATA_PIN, RGB>(leds, NUM_LEDS);
  pinMode(pin8, OUTPUT);
  Serial.begin(9600);
  pinMode(buzzer, OUTPUT); // Définir la sortie du buzzer
}

void loop() {
  int i; // Définir 1 variable pour faire des boucles
  if (debut == 0) {
    sensorReference = analogRead(sensor);
  }
  else {
```

```
for(i=0; i<80; i++) { // Premier son à une fréquence
}
```

```
LoRa.sleep(); // move LoRa transceiver in Sleep mode
sensorValue = analogRead(sensor);
valeur = sensorValue - sensorReference;
Serial.println(valeur);
if (valeur < 20) {
  leds[0] = CRGB(lum, 0, 0);
  leds[1] = CRGB(lum, 0, 0);
  leds[2] = CRGB(lum, 0, 0);
  leds[3] = CRGB(lum, 0, 0);
  leds[4] = CRGB(lum, 0, 0);
  leds[5] = CRGB(lum, 0, 0);
  leds[6] = CRGB(lum, 0, 0);
  leds[7] = CRGB(lum, 0, 0);
  leds[8] = CRGB(lum, 0, 0);
  FastLED.show();
  Serial.println("Pas mal bg");
}
```

```
else {
  leds[0] = CRGB(0, lum, 0);
  leds[1] = CRGB(0, lum, 0);
  leds[2] = CRGB(0, lum, 0);
  leds[3] = CRGB(0, lum, 0);
  leds[4] = CRGB(0, lum, 0);
  leds[5] = CRGB(0, lum, 0);
  leds[6] = CRGB(0, lum, 0);
  leds[7] = CRGB(0, lum, 0);
  leds[8] = CRGB(0, lum, 0);
  FastLED.show();
  Serial.println("DANGER");
  digitalWrite(buzzer, HIGH); // Faire du bruit

  delay(1000); // Attendre 10ms
  digitalWrite(buzzer, LOW); // Silencelayer(10); // Attendre 10ms

  delay(1000); // Attendre 10ms
}
```

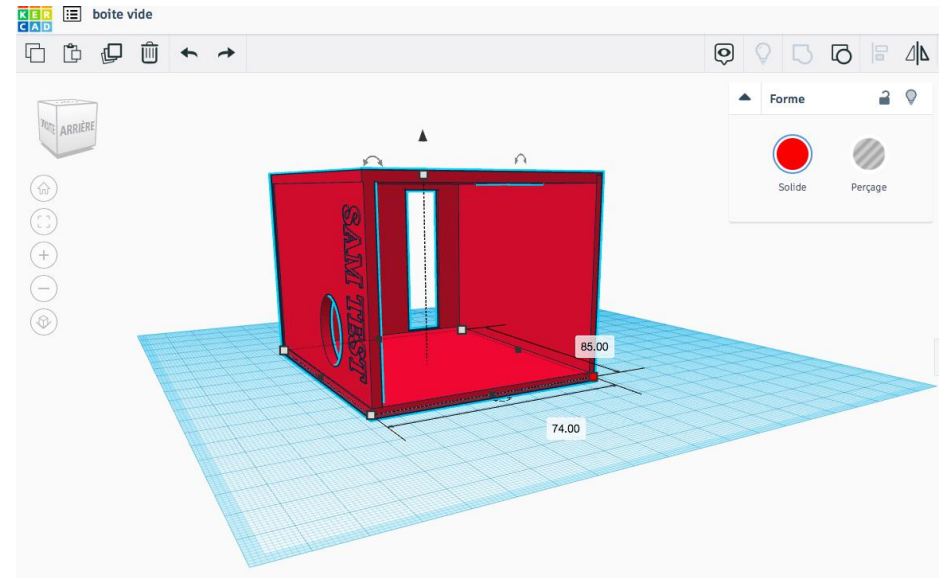
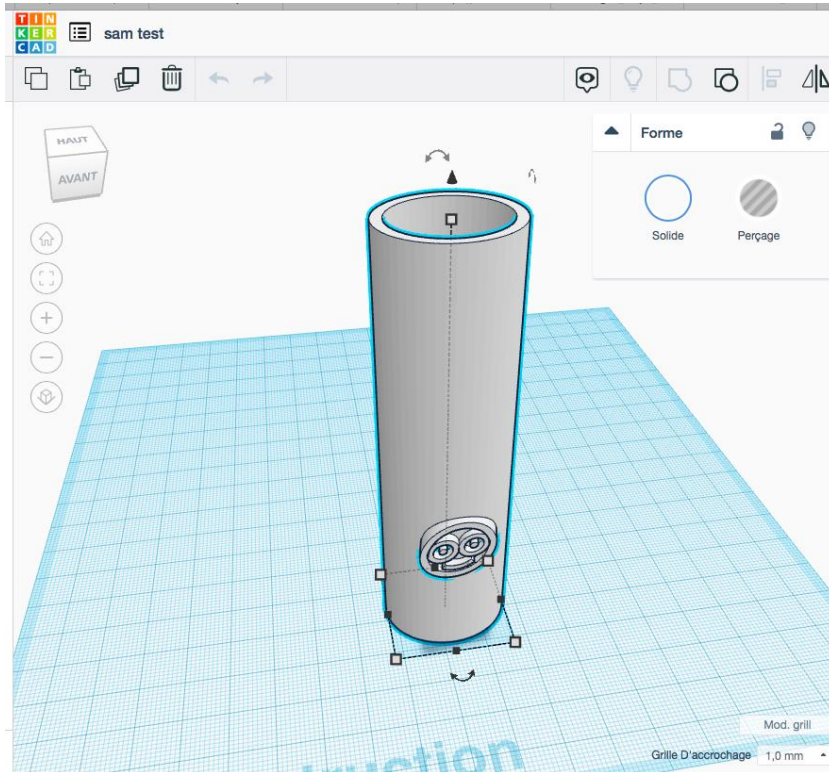
```
for(i=0; i<40; i++); // Deuxième son à une autre fréquence
{
```

```
delay(20); // Attendre 20ms
digitalWrite(buzzer, LOW); // Silence

delay(20); // Attendre 20ms
}
debut = debut + 1;
}
```



# PLAN D'EXECUTION



# CE QU'ON AURAIT PU FAIRE

- A la suite des résultats, permettre ou non l'accès au véhicule. Ainsi, si le conducteur est testé positif, il ne pourrait pas ouvrir la voiture.

OU

- Le système étant relié au faisceau électrique du véhicule permettrait de ne pas pouvoir démarrer après la réalisation du test

# PROBLEMES RENCONTRES

---

- La forte sensibilité du capteur qui a nécessité un calibrage.
- Les impressions des objets que l'on a du recommencé