

# Dispositif d'assistance aux personnes malvoyantes dans un musée



# Sommaire

1/Présentation

2/Diagrammes systèmes

3/Études de solutions

4/Contraintes techniques

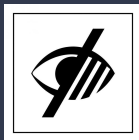
5/Étude de sous-systèmes

6/Problèmes rencontrés et améliorations

7/Programme et conception

8/Répartition des tâches

9/Bilan et conclusion



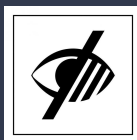
# Présentation

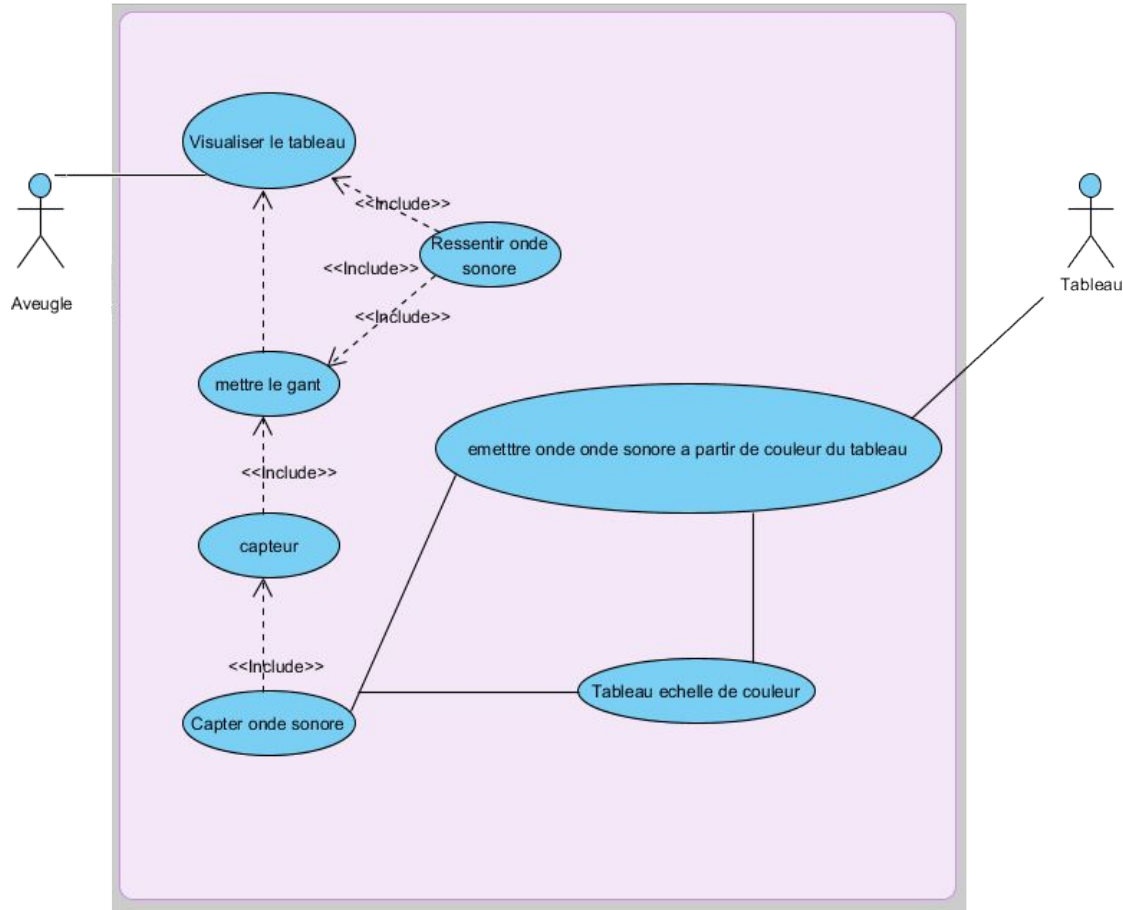


257 millions



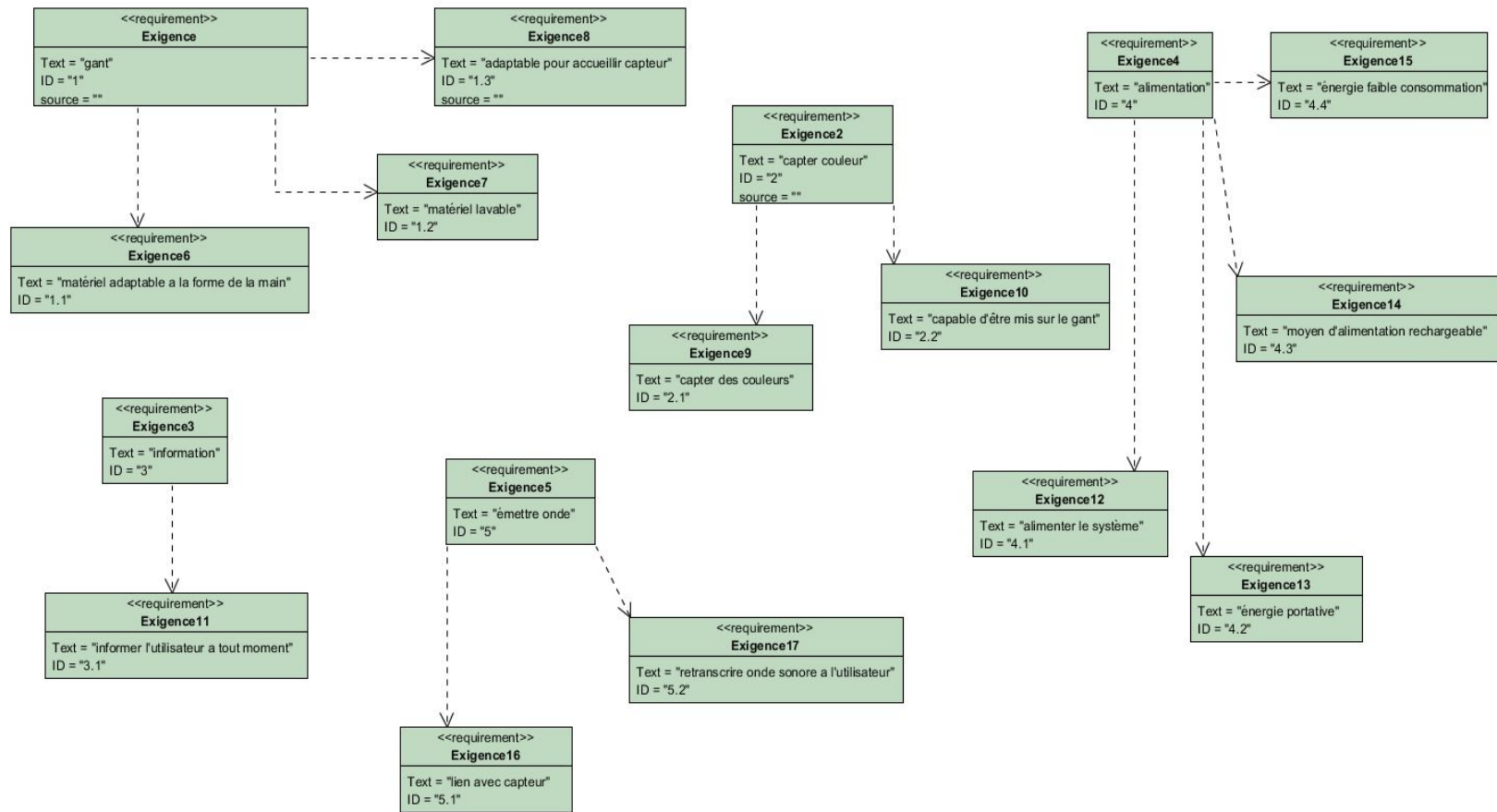
1,7 millions  
(2,5% de la population)





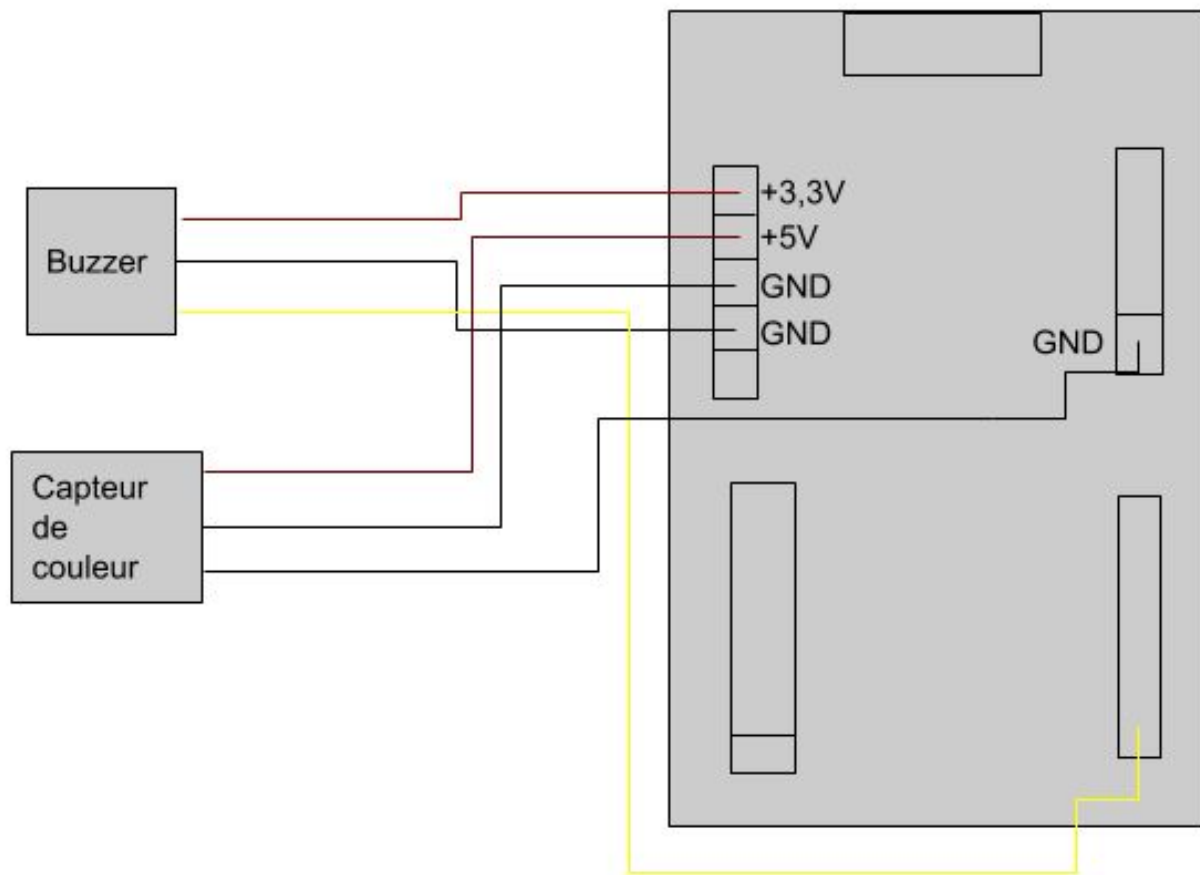
# Diagrammes systèmes

## Diagramme de cas d'utilisation



# Diagrammes systèmes

## Diagrammes d'exigences



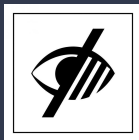
## Diagrammes systèmes


### Schéma de câblage



# Études de solutions

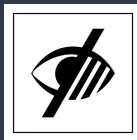
Capteur de couleur



Référence	TCS230	TCS3200
Alimentation	2,7 à 5,5 V	2,7 à 5,5 Vcc
Dimension	24 x 30 mm	Ø 37 x 18 mm
Couleur	RGB	RGB
Sortie max fréquence	12/120/600 KHz	12/120/600 KHz
Sortie	digital	digital
Interface	analogique	analogique
Photo du composant		

# Études de solutions

Retranscription fréquence sonore

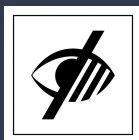


Référence	WV14542	JBLT110BLK
Compatible avec arduino	oui	non
Qualité du son	basse	haute
Photo du composant		



# Études de solutions

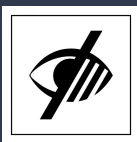
Énergie



Référence	ALIM-BL4AA1
Type	Bloc Alimentation pour 4 piles AA
Tension	6V
Photo du composant	

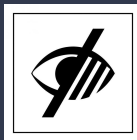
# contraintes techniques

	Contrainte	Spécification	Commentaire
13€	unité central utilisé	carte arduino Uno	simple d'utilisation
2.5€	émetteur	Grove buzzer	compatible avec arduino
6.5€	récepteur	TCS 34725	compatible avec arduino
Total	22€		

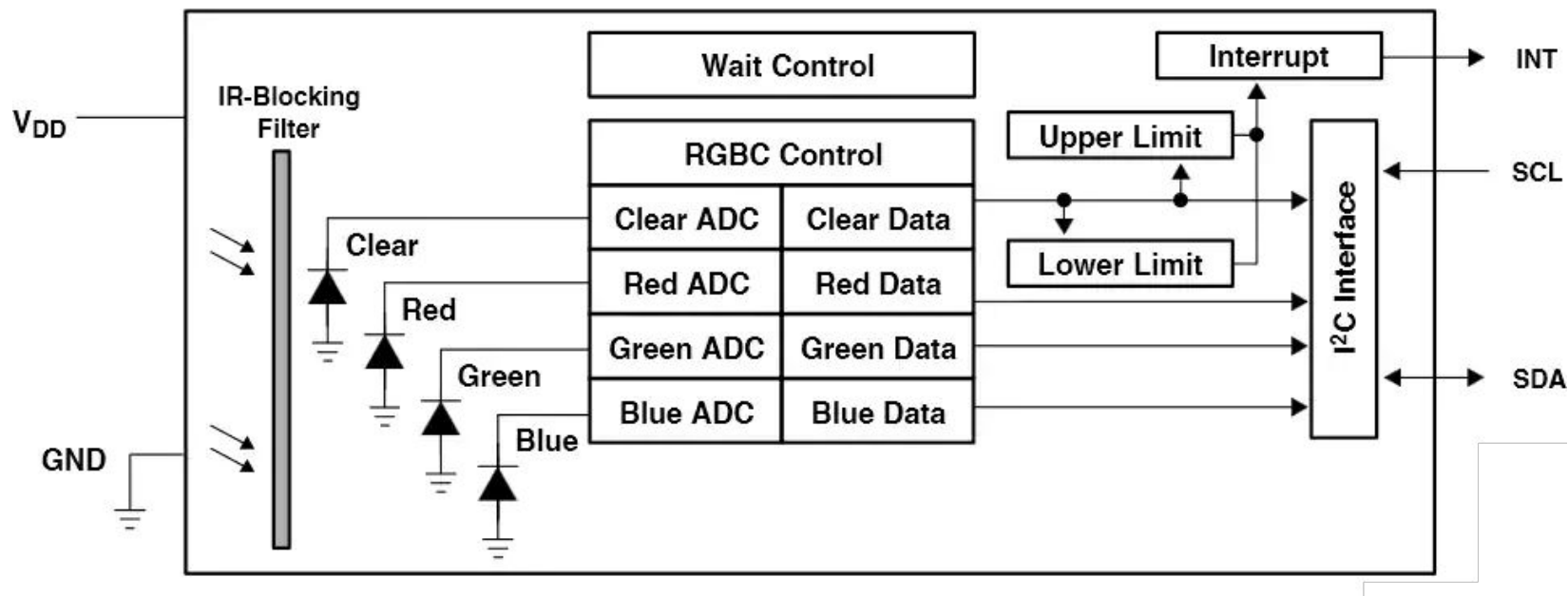


# Étude de sous-systèmes

Capteur

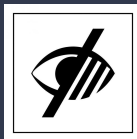


Référence	TCS 34725
Alimentation	5V
Dimension	20 x 20 mm
Couleur	RGBC
Sortie	numérique
Interface	VDD
Température	-40 à 85
Photo du composant	

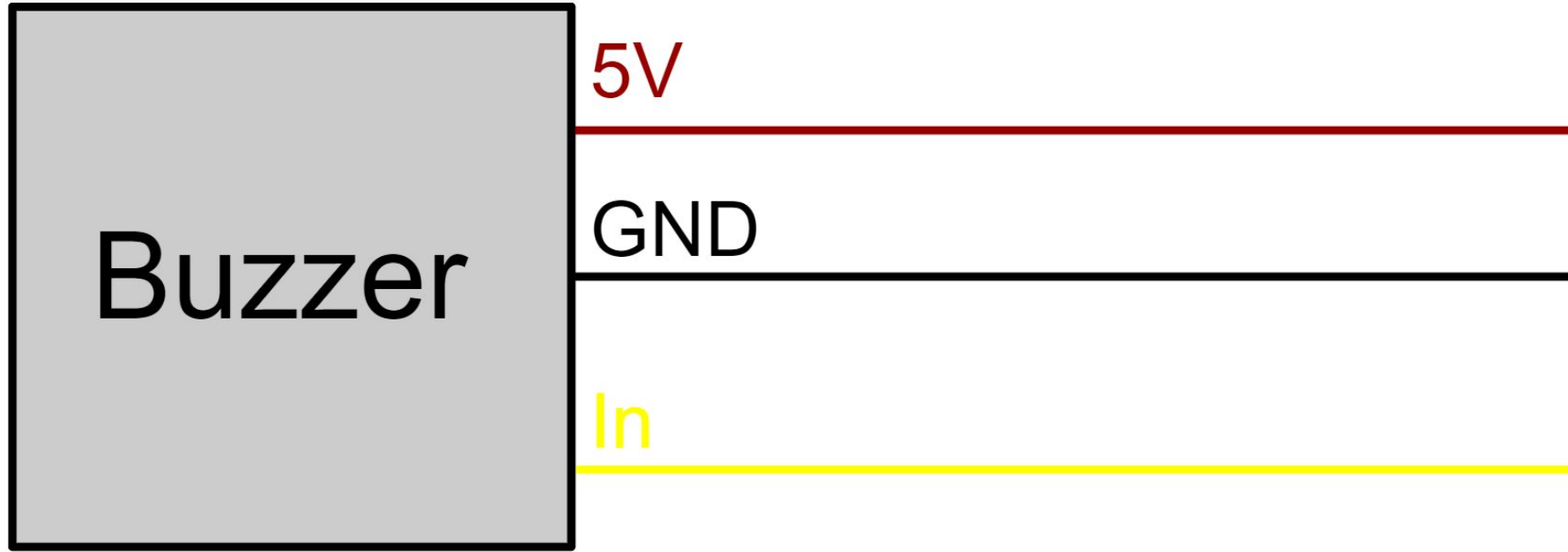


# Étude de sous-systèmes

Actionneur



Référence	SKU 107020000
Alimentation	3,3V
Dimension	20 x 20 mm
Commande	PWM
Module	grove
Sortie sonore	≥85dB
Fréquence de résonance	2300±300Hz
Photo du composant	



# Problèmes rencontrés et améliorations

Problèmes



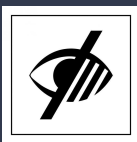
Fréquence non variable



Bruit fort et désagréable



Distance de détection faible



# Problèmes rencontrés et améliorations

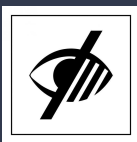
Améliorations



Distance de détection élevé



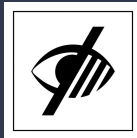
Émetteur sonore personnel





# Programme et conception

Présentation du programme



Initialisation  
du programme

```
#include <Wire.h>
#include "Adafruit_TCS34725.h"
int buzzer = 7;
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_700MS, TCS34725_GAIN_1X);

void setup(void) {
  Serial.begin(9600);
  if (tcs.begin()) {
    Serial.println("Found sensor");
  } else {
    Serial.println("No TCS34725 found ... check your connections");
    while (1);
  }
}
```

Partie  
principale du  
programme

```
void loop() {
  uint16_t r, g, b, c, colorTemp, lux;
  tcs.getRawData(&r, &g, &b, &c);
  colorTemp = tcs.calculateColorTemperature(r, g, b);
  lux = tcs.calculateLux(r, g, b);
  Serial.print("Color Temp: "); Serial.print(colorTemp, DEC); Serial.print(" K - ");
  Serial.print("Lux: "); Serial.print(lux, DEC); Serial.print(" - ");
  Serial.print("R: "); Serial.print(r, DEC); Serial.print(" ");
  Serial.print("G: "); Serial.print(g, DEC); Serial.print(" ");
  Serial.print("B: "); Serial.print(b, DEC); Serial.print(" ");
  Serial.print("C: "); Serial.print(c, DEC); Serial.print(" ");
  Serial.println(" ");
  if (r > g)
  {
    if (r > b)
    {
      sonrouge();
      Serial.print("rouge");
    } else if (b > r)
    {
      sonbleu();
      Serial.print("bleu");
    }
  }
  if (g > r)
  {
    if (g > b)
    {
      sonvert();
      Serial.print("vert");
    } else if (b > g)
    {
      sonbleu();
      Serial.print("bleu");
    }
  }
}
```

Sous  
programmes

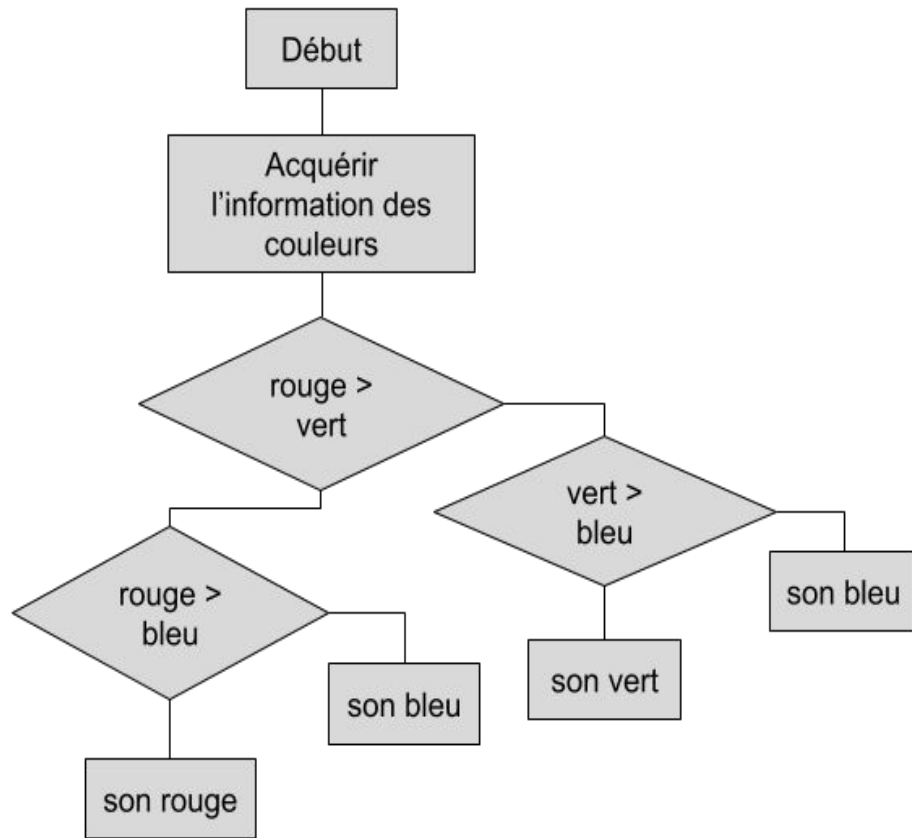
```
void sonvert()
{
  digitalWrite(buzzer, HIGH);
  delay (100);
  digitalWrite(buzzer, LOW);
  delay (900);
}

void sonbleu()
{
  digitalWrite(buzzer, HIGH);
  delay (500);
  digitalWrite(buzzer, LOW);
  delay (500);
}

void sonrouge()
{
  digitalWrite(buzzer, HIGH);
  delay (900);
  digitalWrite(buzzer, LOW);
  delay (100);
}
```

```
1 #include <Wire.h>
2 #include "Adafruit_TCS34725.h"
3 int buzzer = 7;
4 Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_700MS, TCS34725_GAIN_1X);
5 void setup(void) {
6     Serial.begin(9600);
7     if (tcs.begin()) {
8         Serial.println("Found sensor");
9     } else {
10         Serial.println("No TCS34725 found ... check your connections");
11         while (1);
12     }
13 }
```

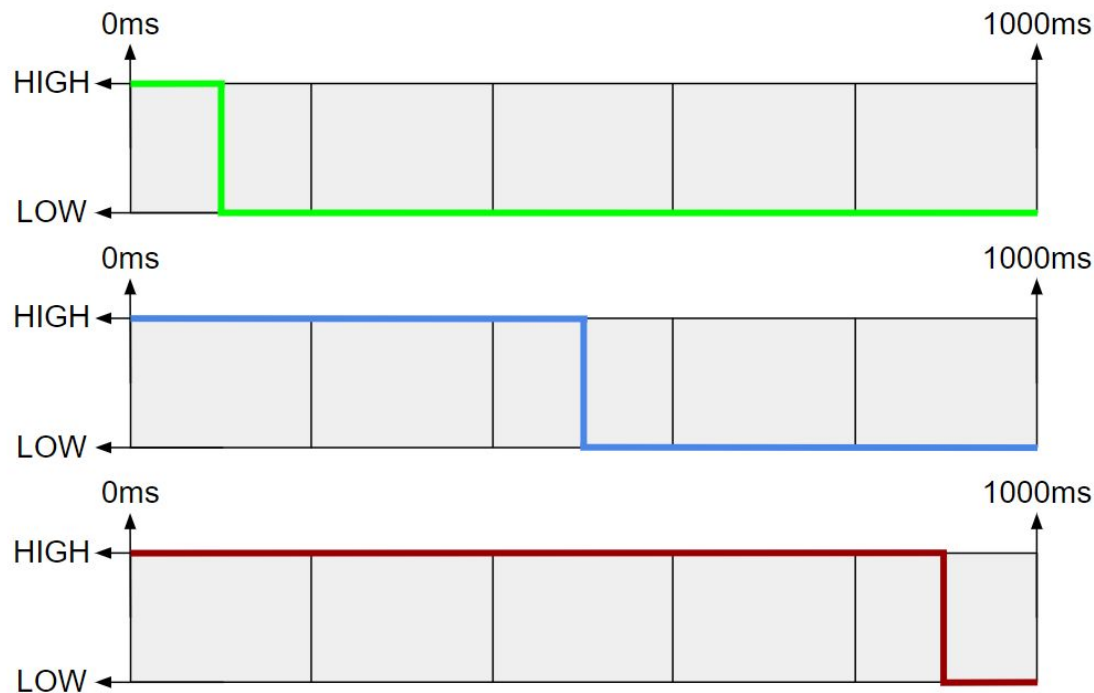




```

14 void loop() {
15   uint16_t r, g, b, c, colorTemp, lux;
16   tcs.getRawData(&r, &g, &b, &c);
17   colorTemp = tcs.calculateColorTemperature(r, g, b);
18   lux = tcs.calculateLux(r, g, b);
19   Serial.print("Color Temp: "); Serial.print(colorTemp, DEC); Serial.print(" K - ");
20   Serial.print("Lux: "); Serial.print(lux, DEC); Serial.print(" - ");
21   Serial.print("R: "); Serial.print(r, DEC); Serial.print(" ");
22   Serial.print("G: "); Serial.print(g, DEC); Serial.print(" ");
23   Serial.print("B: "); Serial.print(b, DEC); Serial.print(" ");
24   Serial.print("C: "); Serial.print(c, DEC); Serial.print(" ");
25   Serial.println(" ");
26   if (r > g)
27   {
28     if (r > b)
29     {
30       sonrouge();
31       Serial.print("rouge");
32     } else if (b > r)
33     {
34       sonbleu();
35       Serial.print("bleu");
36     }
37   }
38   if (g > r)
39   {
40     if (g > b)
41     {
42       sonvert();
43       Serial.print("vert");
44     } else if (b > g)
45     {
46       sonbleu();
47       Serial.print("bleu");
48     }
49   }
50 }
  
```



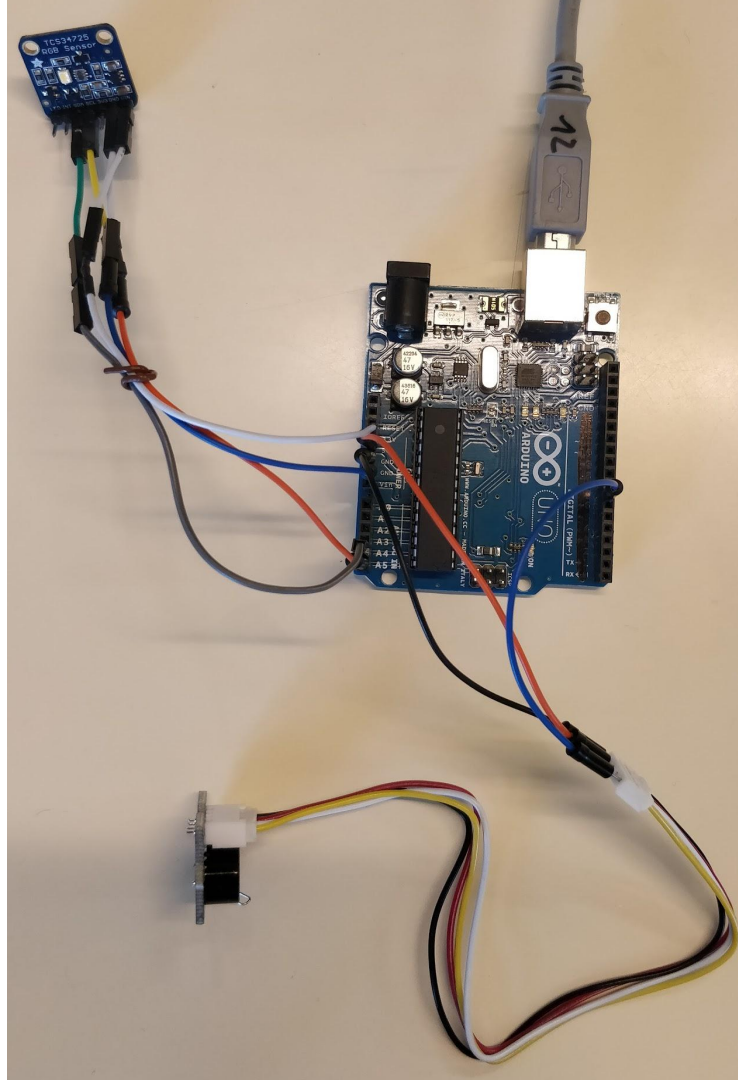
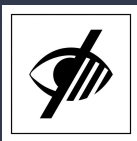


```
51 void sonvert()  
52 {  
53     digitalWrite(buzzer, HIGH);  
54     delay (100);  
55     digitalWrite(buzzer, LOW);  
56     delay (900);  
57 }  
58 void sonbleu()  
59 {  
60     digitalWrite(buzzer, HIGH);  
61     delay (500);  
62     digitalWrite(buzzer, LOW);  
63     delay (500);  
64 }  
65 void sonrouge()  
66 {  
67     digitalWrite(buzzer, HIGH);  
68     delay (900);  
69     digitalWrite(buzzer, LOW);  
70     delay (100);  
71 }
```



# Programme et conception

Conception



Tâche	1	2	3	4	5	6	7	8	9	10
Répartition des tâches										
analyse fonctionnelle										
planification										
recherche des solutions pour le son										
recherche des solutions pour la détection des couleurs										
études du sous-système										



# Répartition des tâches

## Préparation et recherches

Tâche	1	2	3	4	5	6	7	8	9	10
établir une relation entre une couleur et une fréquence sonore										
extraire de la donnée issue des capteur, l'information des couleurs										
Assembler les différentes parties / Phase de test										



## Répartition des tâches

### Conception

Tâche	1	2	3	4	5	6	7	8	9	10
Préparer la synthèse										
restitution oral										

Équipe
Léo / Robin
Léo / Axel



## Répartition des tâches

### Préparation oral



# Bilan et conclusion



- Ensemble
- Cohésion de groupe
- Résultat



# Bilan et conclusion



- Ordinateurs lents
- Logiciel à mettre à jour constamment
- tâche mal réparties



# Bilan et conclusion

