



# Buggy exploreur

Maxime Roman et Suzanne Grand



# Problématique

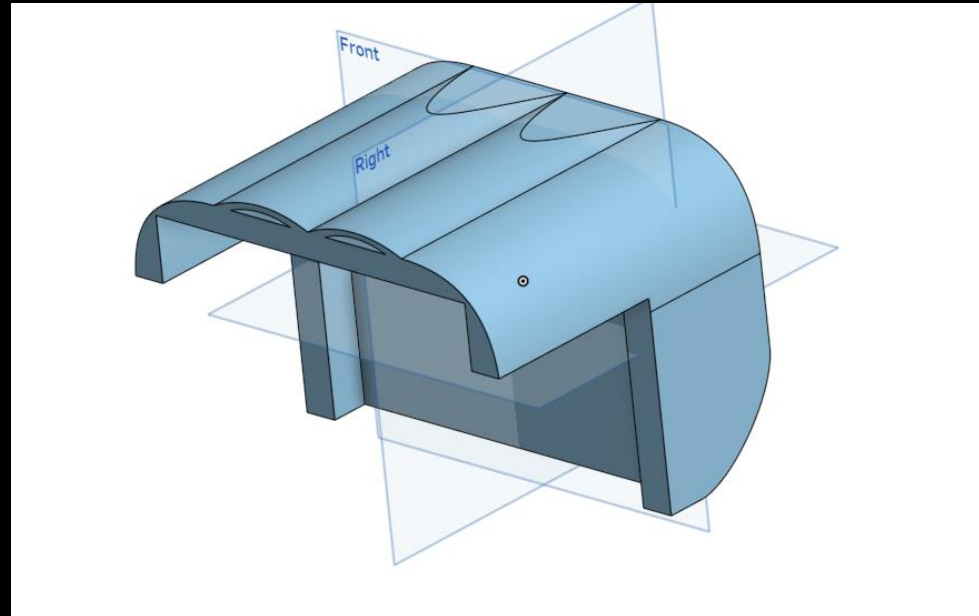
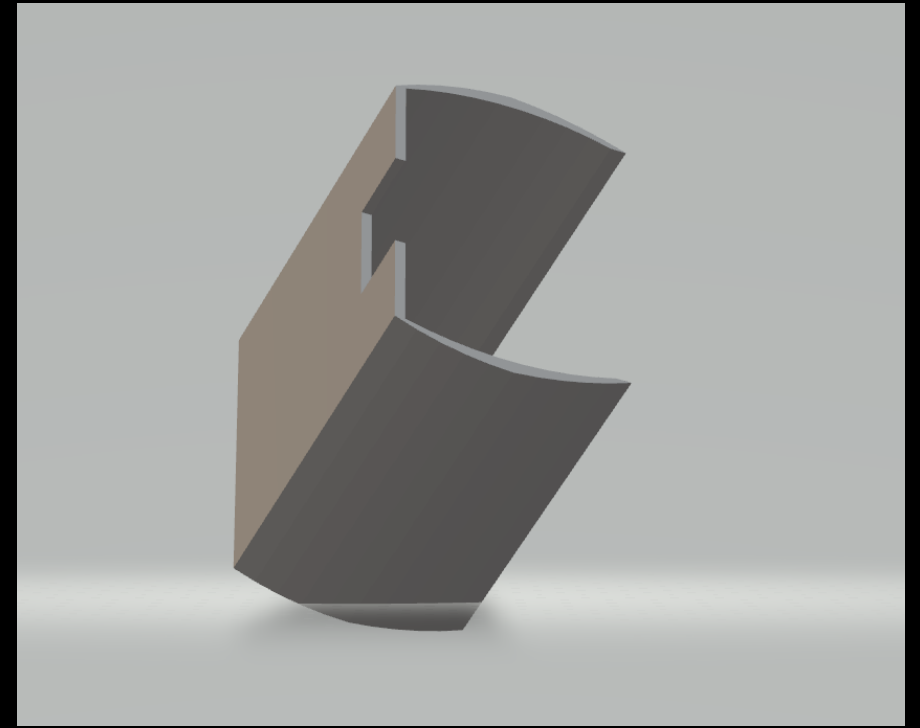
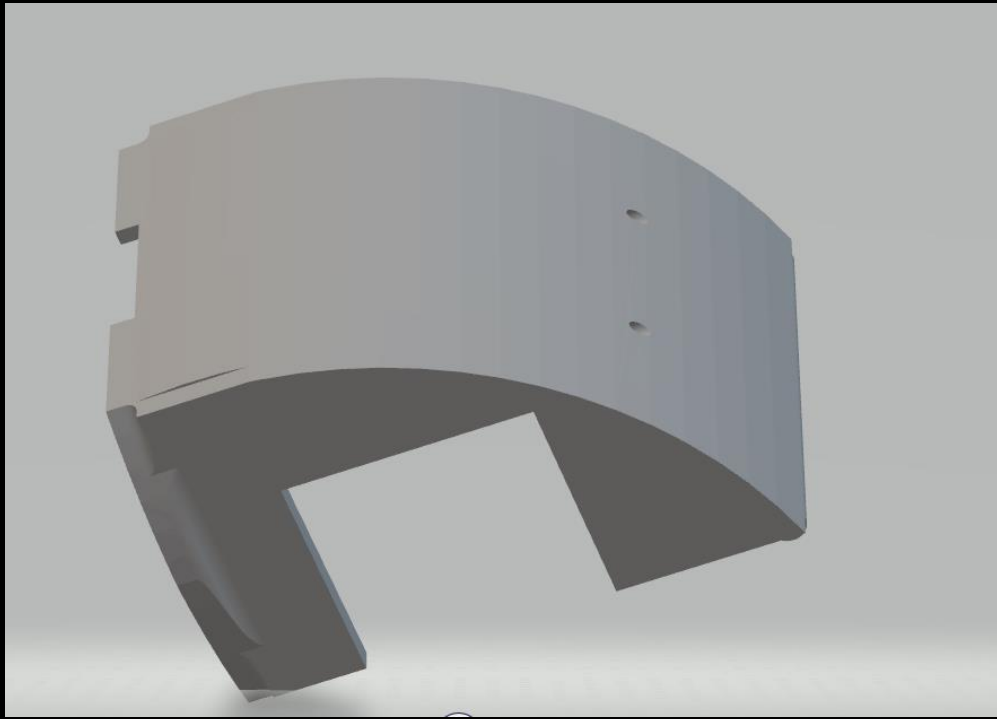
Crée un rover capable de parcourir des zones non accessibles aux êtres humains et nous transmettre les données en temps réel.

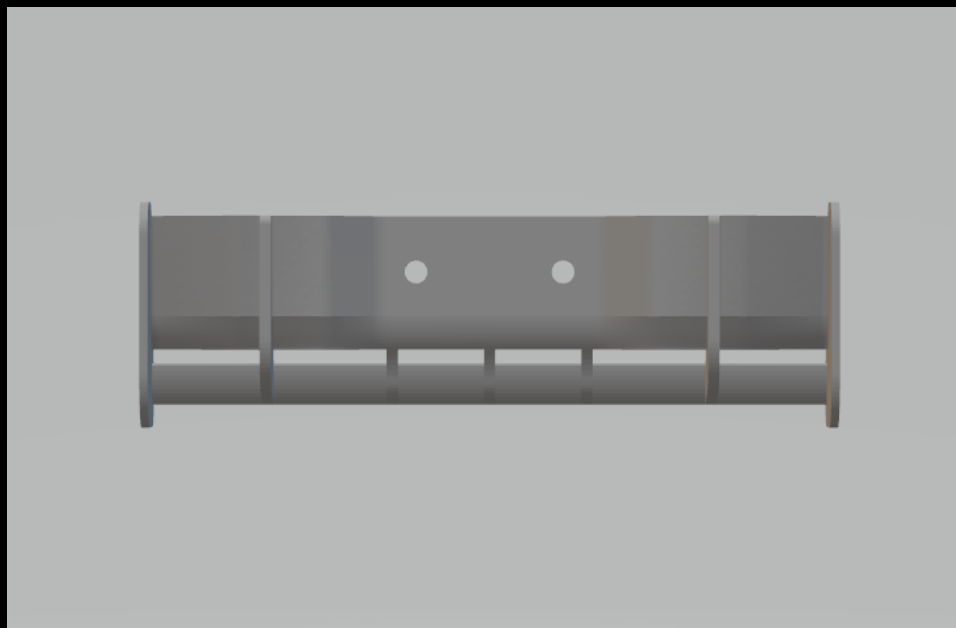
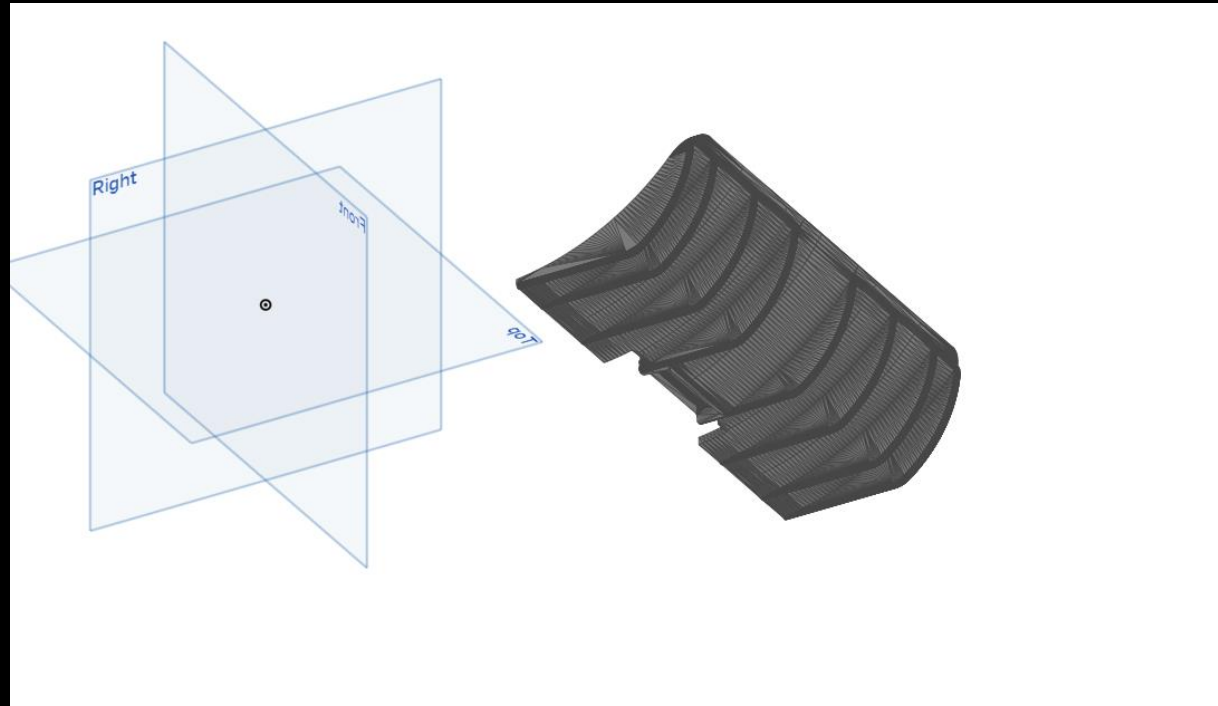
# Les fonctionnalités du projet

- Capteur de température (DHT22)
- Deux modules Bluetooth
- Afficher la température sur l'écran LCD

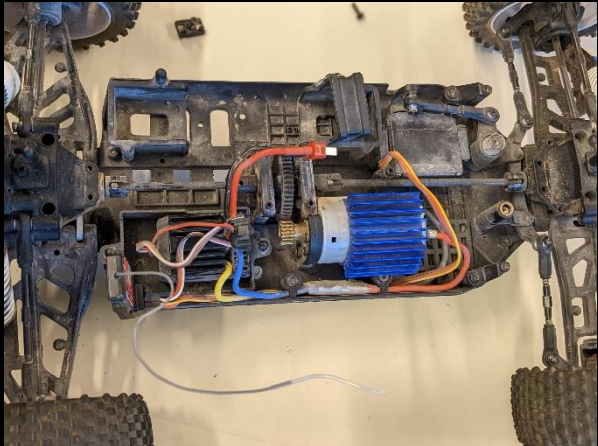
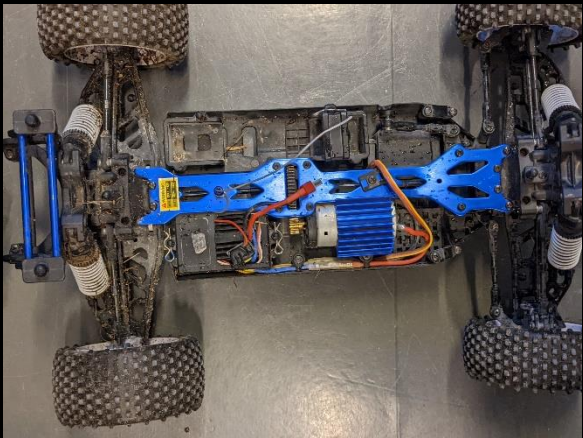
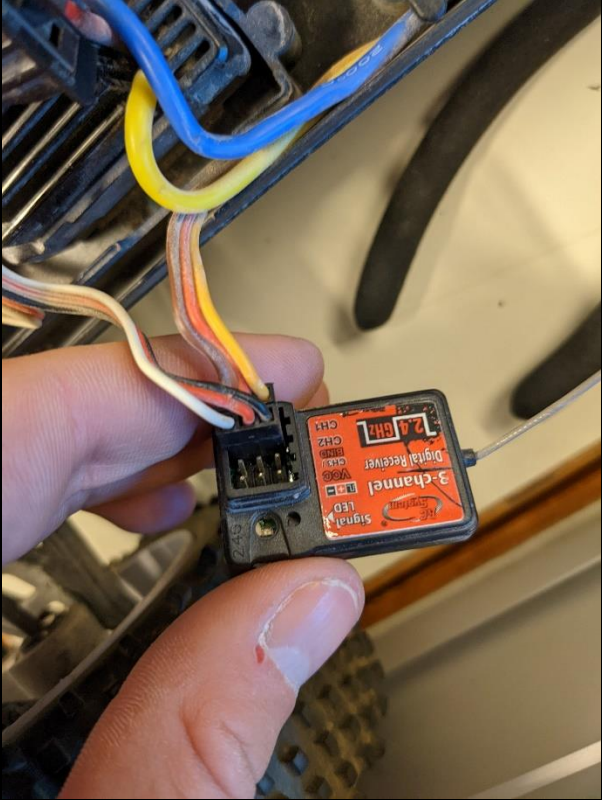
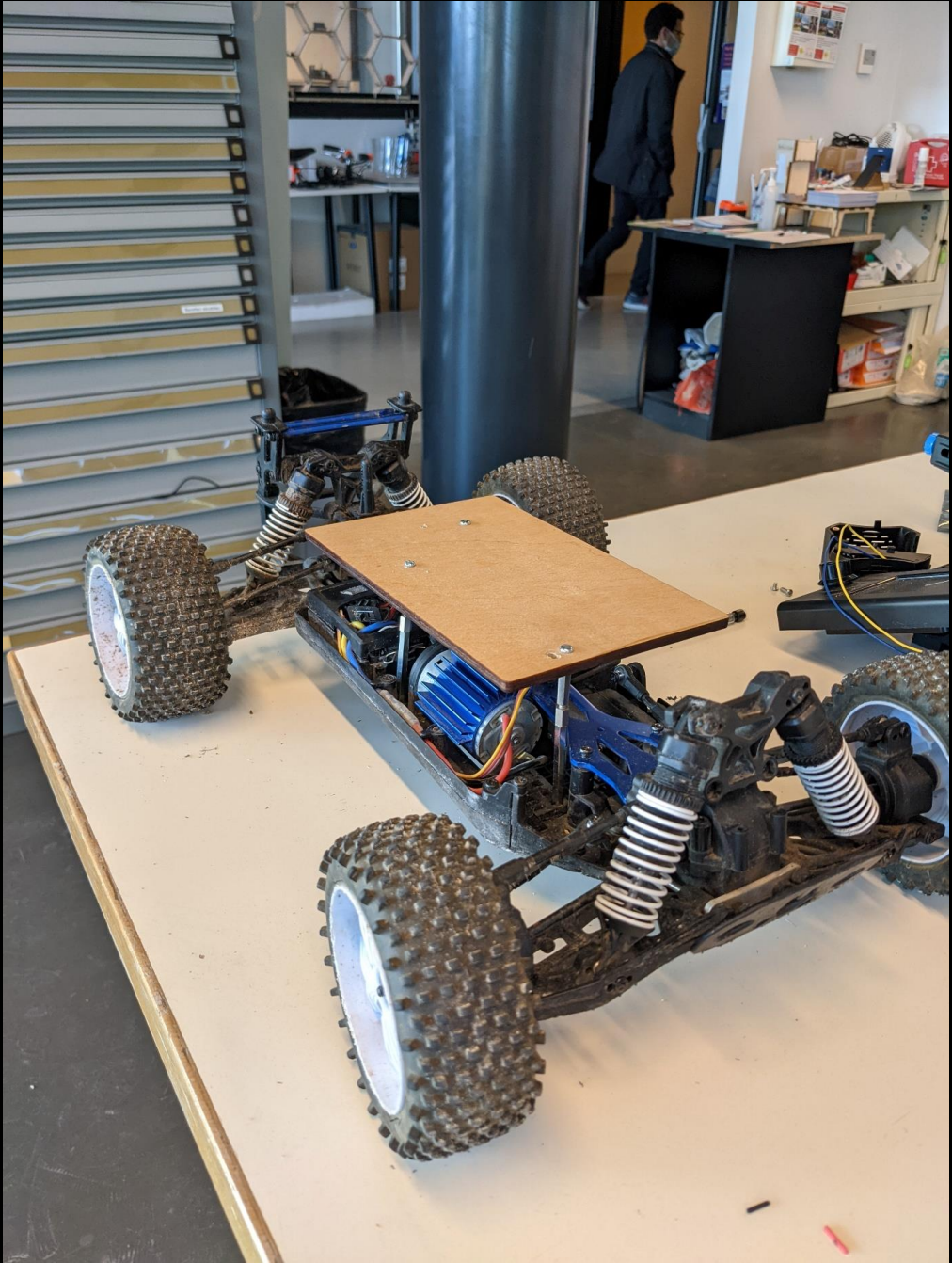


# Modélisation du projet











# Programmation et montage

- Codage de la temperature, de l'écran LCD et des modules Bluetooth
- Deux montages distincts



```

SoftwareSerial BlueT(RX,TX);
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

void setup() {
  Serial.begin(9600);
  Serial.println("6");
  lcd.begin(20, 4);
  BlueT.begin(38400);
  pinMode(bouton, INPUT);
}

void loop() {
  // capter puis envoyer la pression du bouton
  B=digitalRead(bouton);
  if (B==LOW){
    BlueT.write("B");
    BlueT.print(0);
  }

  //récupération des données :
  if (BlueT.available()){
    Data=BlueT.read();
    Serial.println(Data);
    if (Data=='H') {
      PWN=BlueT.parseFloat();
      Serial.println(PWN);
    }
  }
}

```

```

//Affichages sur écran LCD :
// Ligne 1:
lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
lcd.print("Bienvenue chez Rover");

// Ligne 2:
lcd.setCursor(0, 1); // positionne le curseur à la colonne 1 et à la ligne 2
lcd.print("Humidite: ");
lcd.setCursor(10, 1);
lcd.print(PWN);
lcd.setCursor(15, 1);
lcd.print("%");

// Ligne 3:
lcd.setCursor(0, 2); // positionne le curseur à la colonne 1 et à la ligne 3
lcd.print("TempRess: ");
lcd.setCursor(10, 2);
lcd.print(PWL);
lcd.setCursor(15, 2);
lcd.print("deg");

// Ligne 4:
lcd.setCursor(0, 3); // positionne le curseur à la colonne 1 et à la ligne 4
lcd.print("Temperature: ");
lcd.setCursor(13, 3);
lcd.print(PWM);
lcd.setCursor(17, 3);
lcd.print("deg");

```

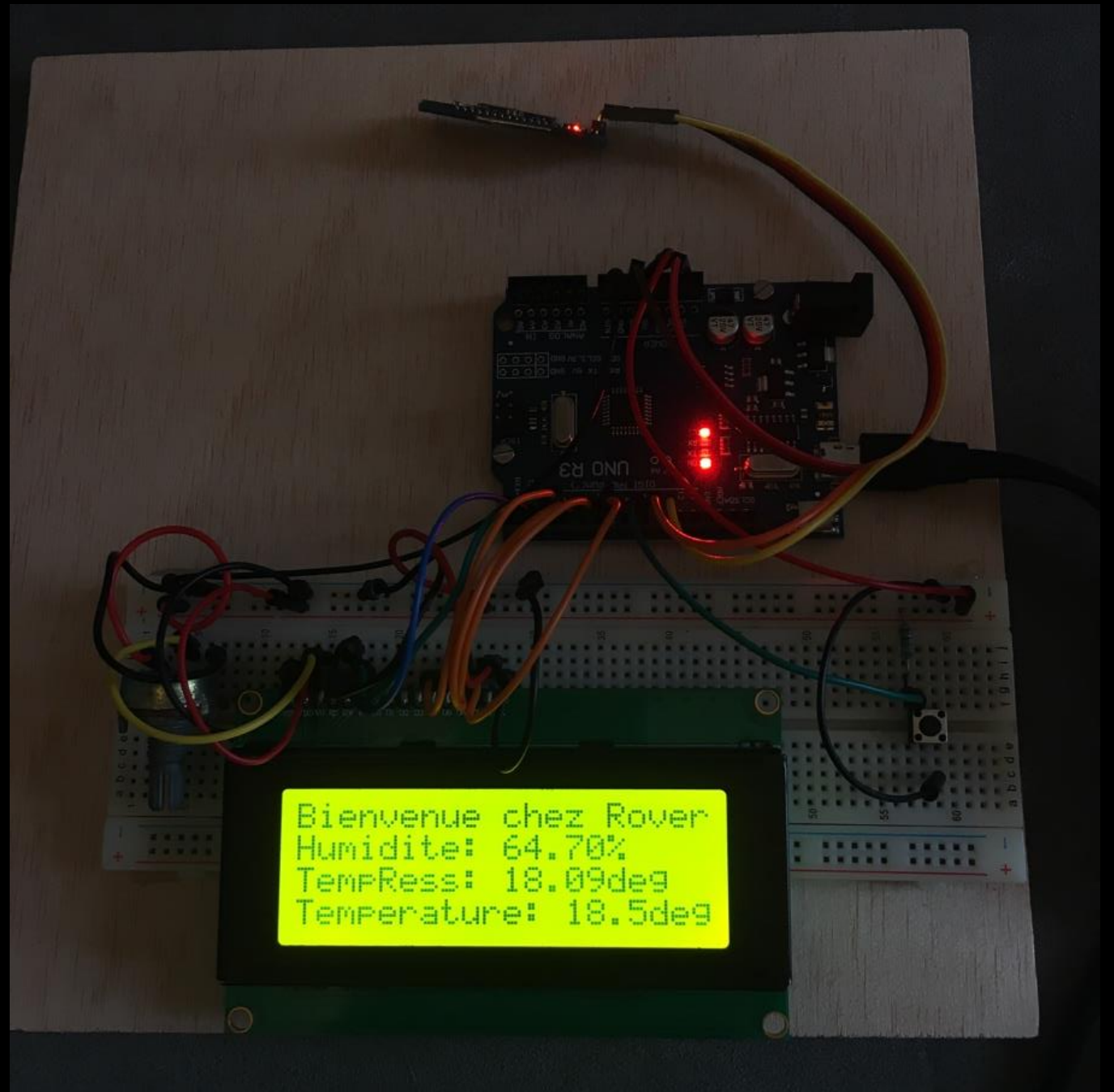
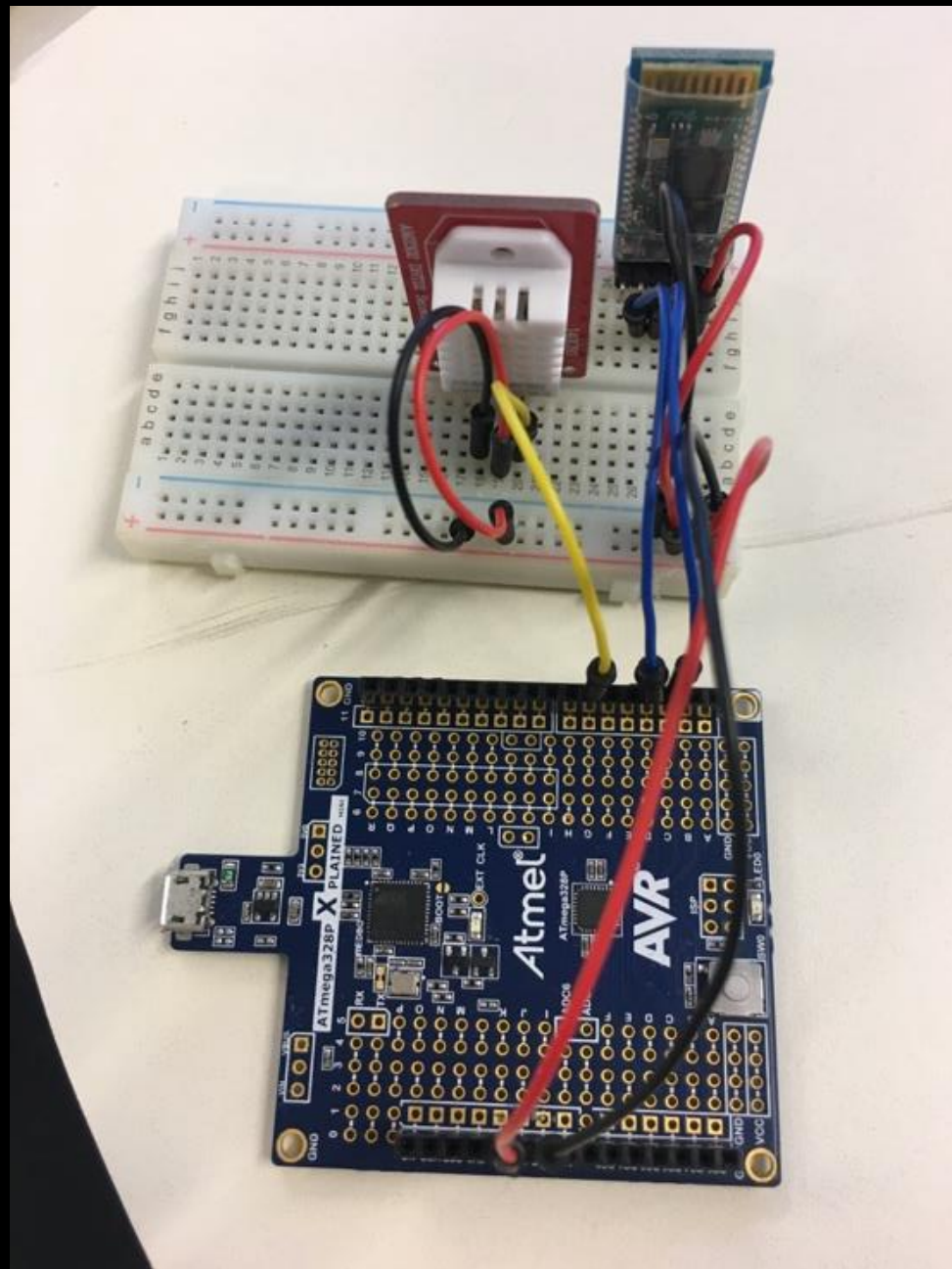
```
void setup() {
  Serial.begin(9600);
  Serial.println("5");
  Serial.println("Bonjour -Pret pour les commandes AT");
  Serial.println();
  dht.begin();
  BlueT.begin(38400);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
}

void loop() {
  if (BlueT.available()){
    Data=BlueT.read();
    Serial.println(Data);
    if (Data=='B') {
      v=BlueT.parseInt();
      Serial.println(v);
    }
  }
}
```

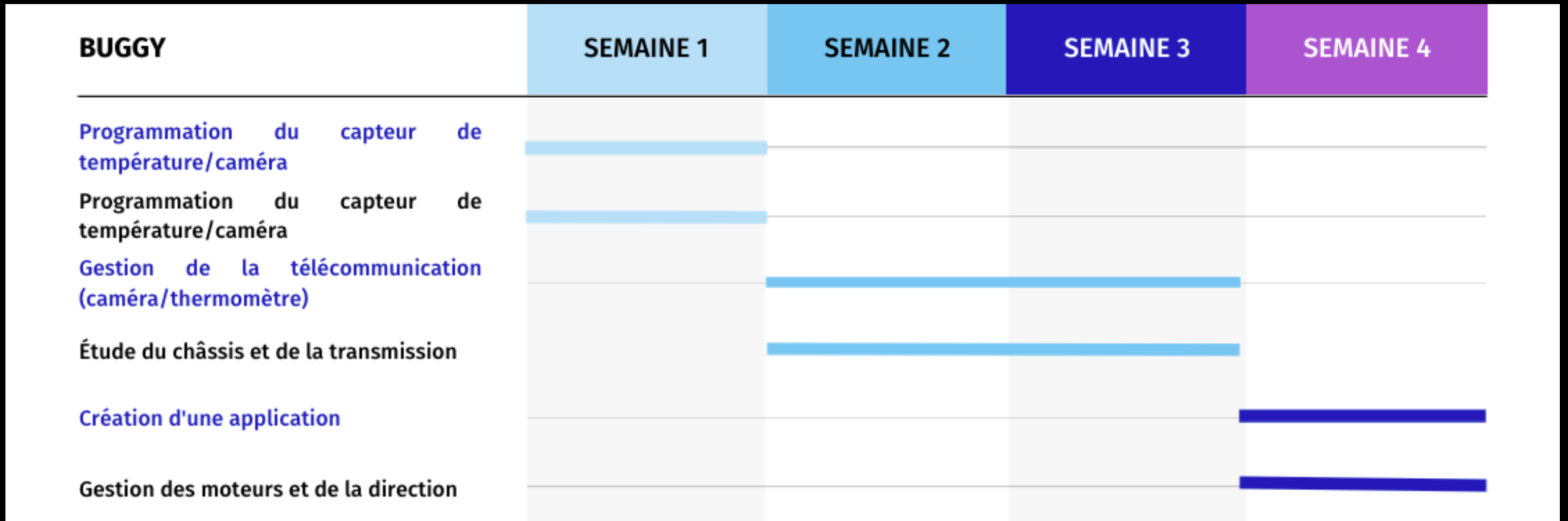
```
//envoyer data vers esclave
  Serial.print("Le maître envoie : ");
  Serial.println(tauxHumidite);
  BlueT.write("H");
  BlueT.print(tauxHumidite);

  Serial.print("Le maître envoie : ");
  Serial.println(tempRessentieCelsius);
  BlueT.write("R");
  BlueT.print(tempRessentieCelsius);

  Serial.print("Le maître envoie : ");
  Serial.println(tempCelsius);
  BlueT.write("T");
  BlueT.print(tempCelsius);
```



# Diagramme de Gantt



## BUGGY

SEMAINE 5

SEMAINE 6

SEMAINE 7

SEMAINE 8

Création d'une application

Création liaisons télécommande buggy

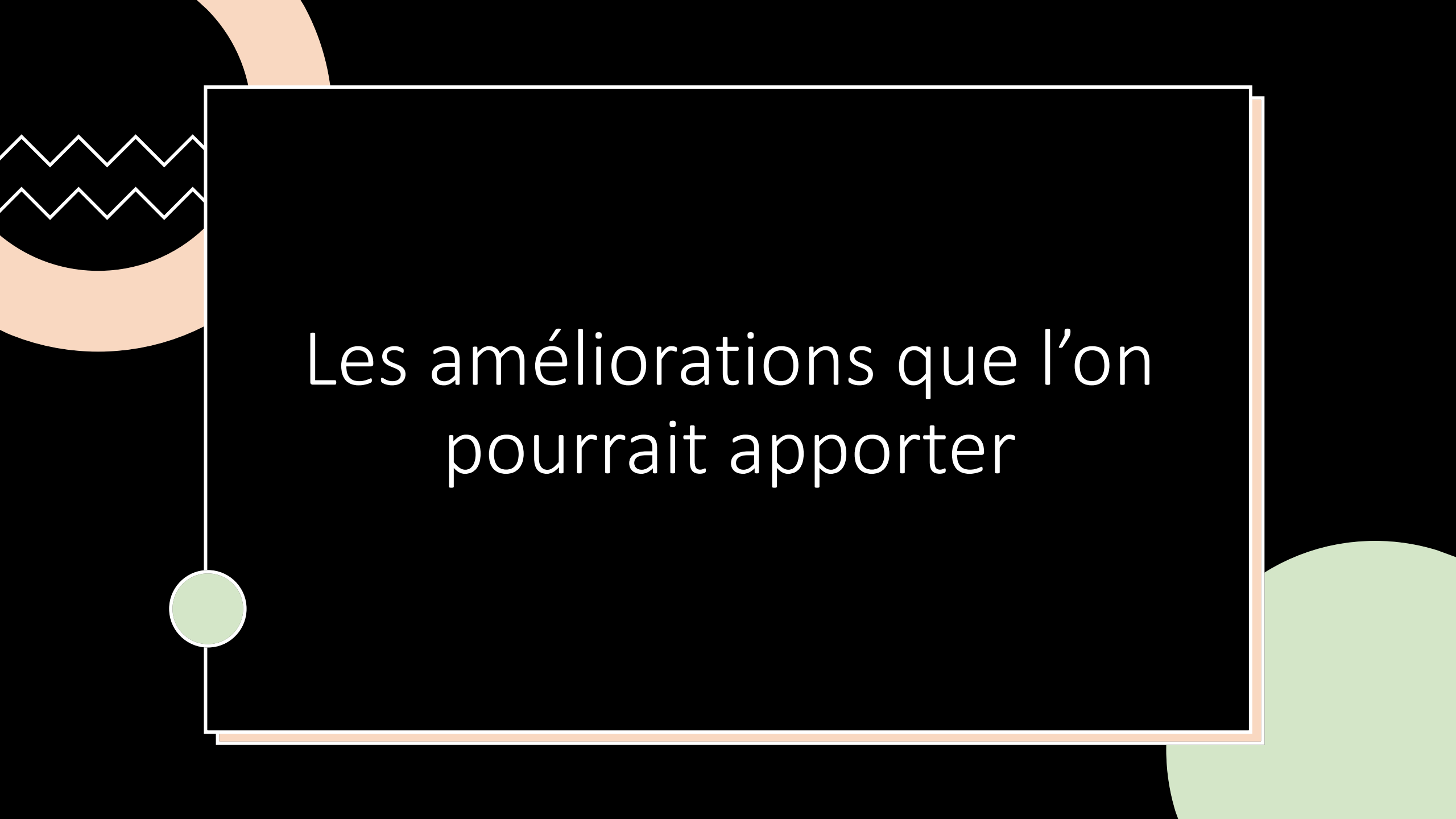
Création de liaisons entre  
caméra/thermomètre et application

Révisions des programmes moteurs et  
directions avec la télécommande

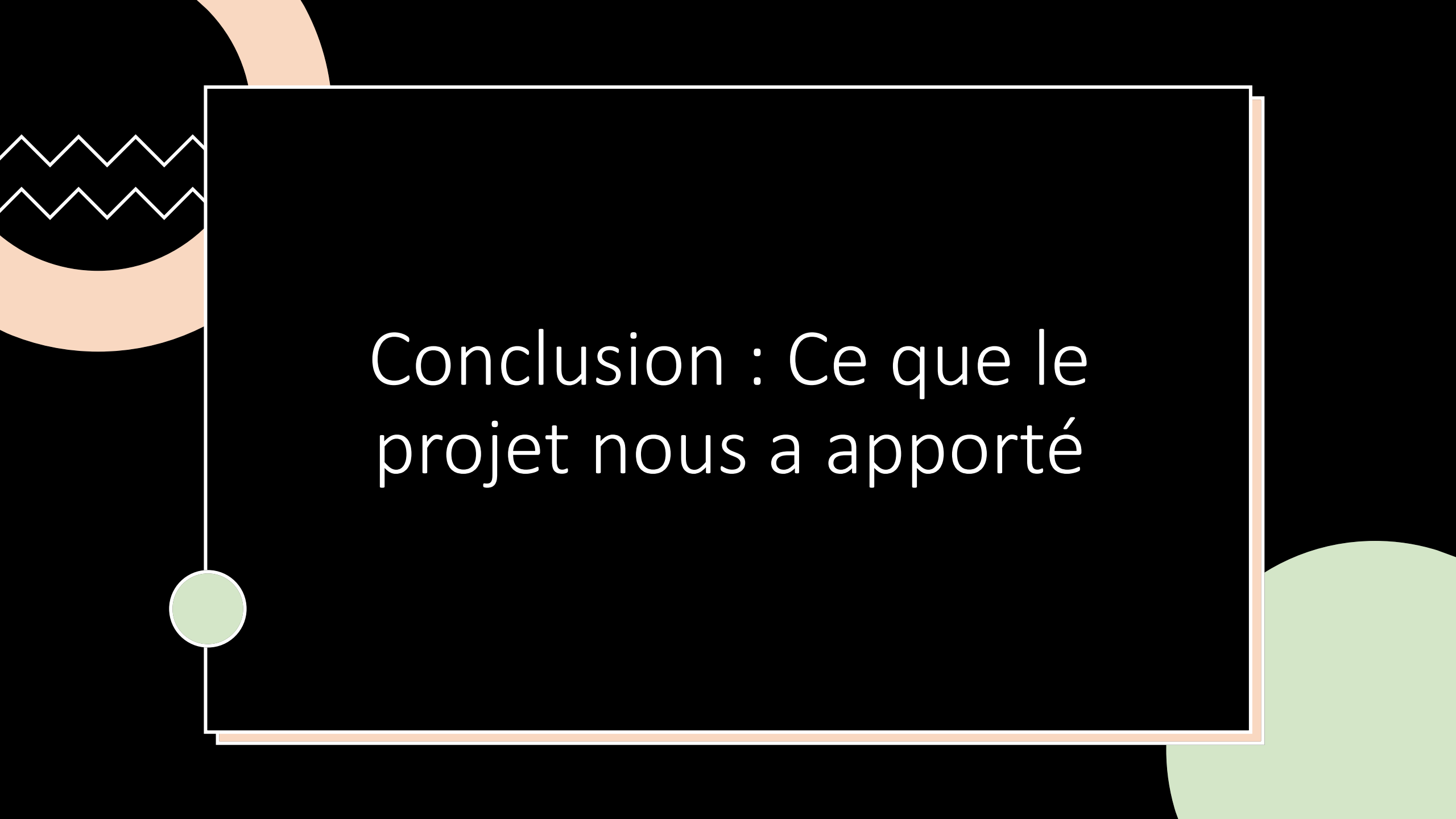
Création de la coque

Finalisation/finitions et préparation oral

Finalisation/finitions et préparation oral



Les améliorations que l'on  
pourrait apporter



Conclusion : Ce que le  
projet nous a apporté