

Camion de pompiers

Fait par: Benjamin Sun LIU,
Ranthosh MAHALINGAM, Sören
MAINVILLE



Introduction

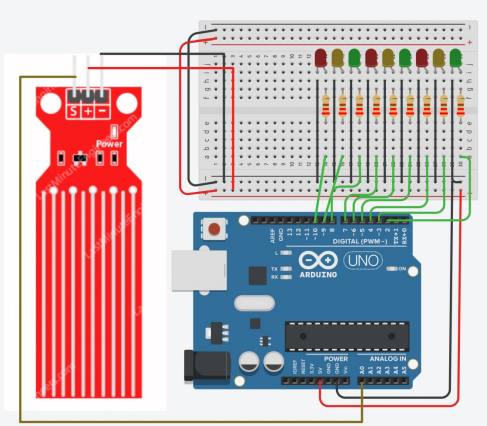
- Le grand incendie de Québec en 1866 brûle presque 3000 bâtiments en une seule journée.
- Les feux de forêt enregistrés en 2021 montrent une augmentation de 40% par rapport à la moyenne de la dernière décennie.
- Avec les nombreux progrès technologiques et leur grande disponibilité, ces évènements nous ont inspirés à développer un véhicule pouvant détecter les flammes et les éteindre rapidement, avec précision.



Capteur à niveau d'eau

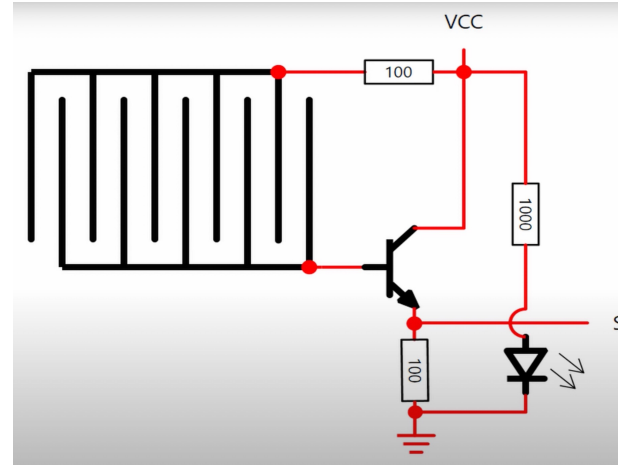


PARTIE
INFORMATIQUE



Pin S = analogue

PARTIE
INFORMATIQUE



Composantes du capteur:

1. 5 traces de puissance fait à partir cuivre
2. 5 traces de détection fait à partir cuivre
3. 3 résistances: deux 100 ohm
un 1000 ohm
4. J3Y (transistor)
5. LED

```
1 const int read = A0;
2 int value;
3 int redLED1 = 10;
4 int yellowLED1 = 9;
5 int greenLED1 = 8;
6 int redLED2 = 7;
7 int yellowLED2 = 6;
8 int greenLED2 = 5;
9 int redLED3 = 4;
10 int yellowLED3 = 3;
11 int greenLED3 = 2;
12
13 void setup()
14 {
15   Serial.begin(9600);
16
17   pinMode(redLED1, OUTPUT);
18   pinMode(greenLED1, OUTPUT);
19   digitalWrite(redLED1, LOW);
20   digitalWrite(yellowLED1, LOW);
21   digitalWrite(greenLED1, LOW);
22
23   pinMode(redLED2, OUTPUT);
24   pinMode(greenLED2, OUTPUT);
25   digitalWrite(redLED2, LOW);
26   digitalWrite(yellowLED2, LOW);
27   digitalWrite(greenLED2, LOW);
28
29   pinMode(redLED3, OUTPUT);
30   pinMode(greenLED3, OUTPUT);
31   digitalWrite(redLED3, LOW);
32   digitalWrite(yellowLED3, LOW);
33   digitalWrite(greenLED3, LOW);
34
35   void loop()
36   {
37     value = analogRead(read);
38
39     if (value==80){
40       Serial.println("Water level: 0mm - Empty!");
41       digitalWrite(redLED1, HIGH);
42       digitalWrite(yellowLED1, LOW);
43       digitalWrite(greenLED1, LOW);
44       digitalWrite(yellowLED2, LOW);
45       digitalWrite(redLED2, LOW);
46       digitalWrite(greenLED2, LOW);
47       digitalWrite(yellowLED3, LOW);
48       digitalWrite(greenLED3, LOW);
49     }
50     else if (value==80 && value==30){
51       Serial.println("Water level: 0mm to 5mm");
52       digitalWrite(redLED1, HIGH);
53       digitalWrite(yellowLED1, HIGH);
54       digitalWrite(greenLED1, LOW);
55       digitalWrite(yellowLED2, LOW);
56       digitalWrite(greenLED2, LOW);
57       digitalWrite(yellowLED3, HIGH);
58       digitalWrite(greenLED3, HIGH);
59     }
60     else if (value==65 && value==60){
61       Serial.println("Water level: 10mm to 15mm");
62       digitalWrite(redLED1, HIGH);
63       digitalWrite(yellowLED1, HIGH);
64       digitalWrite(greenLED1, HIGH);
65       digitalWrite(redLED2, HIGH);
66       digitalWrite(yellowLED2, LOW);
67       digitalWrite(greenLED2, LOW);
68       digitalWrite(yellowLED3, LOW);
69       digitalWrite(greenLED3, LOW);
70     }
71     else if (value==60 && value==50){
72       Serial.println("Water level: 15mm to 20mm");
73       digitalWrite(redLED1, HIGH);
74       digitalWrite(yellowLED1, HIGH);
75       digitalWrite(greenLED1, HIGH);
76       digitalWrite(redLED2, HIGH);
77       digitalWrite(yellowLED2, HIGH);
78       digitalWrite(greenLED2, LOW);
79       digitalWrite(yellowLED3, LOW);
80       digitalWrite(greenLED3, LOW);
81     }
82     else if (value==50 && value==40){
83       Serial.println("Water level: 20mm to 25mm");
84       digitalWrite(redLED1, HIGH);
85       digitalWrite(yellowLED1, HIGH);
86       digitalWrite(greenLED1, HIGH);
87       digitalWrite(redLED2, HIGH);
88       digitalWrite(yellowLED2, HIGH);
89       digitalWrite(greenLED2, HIGH);
90       digitalWrite(yellowLED3, LOW);
91       digitalWrite(greenLED3, LOW);
92     }
93     else if (value==50 && value==70){
94       Serial.println("Water level: 25mm to 30mm");
95       digitalWrite(redLED1, HIGH);
96       digitalWrite(yellowLED1, HIGH);
97       digitalWrite(greenLED1, HIGH);
98       digitalWrite(redLED2, HIGH);
99       digitalWrite(yellowLED2, HIGH);
100      digitalWrite(greenLED2, HIGH);
101      digitalWrite(yellowLED3, HIGH);
102      digitalWrite(greenLED3, LOW);
103      digitalWrite(greenLED3, LOW);
104     }
105     else if (value==70 && value==75){
106       Serial.println("Water level: 30mm to 35mm");
107       digitalWrite(redLED1, HIGH);
108       digitalWrite(yellowLED1, HIGH);
109       digitalWrite(greenLED1, HIGH);
110       digitalWrite(redLED2, HIGH);
111       digitalWrite(yellowLED2, HIGH);
112       digitalWrite(greenLED2, HIGH);
113       digitalWrite(yellowLED3, HIGH);
114       digitalWrite(greenLED3, HIGH);
115     }
116     else if (value==75){
117       Serial.println("Water level: 35mm to 40mm");
118       digitalWrite(redLED1, HIGH);
119       digitalWrite(yellowLED1, HIGH);
120       digitalWrite(greenLED1, HIGH);
121       digitalWrite(redLED2, HIGH);
122       digitalWrite(yellowLED2, HIGH);
123       digitalWrite(greenLED2, HIGH);
124       digitalWrite(yellowLED3, HIGH);
125       digitalWrite(greenLED3, HIGH);
126     }
127     delay(5000);
128   }
129 }
```



Capteur de flamme

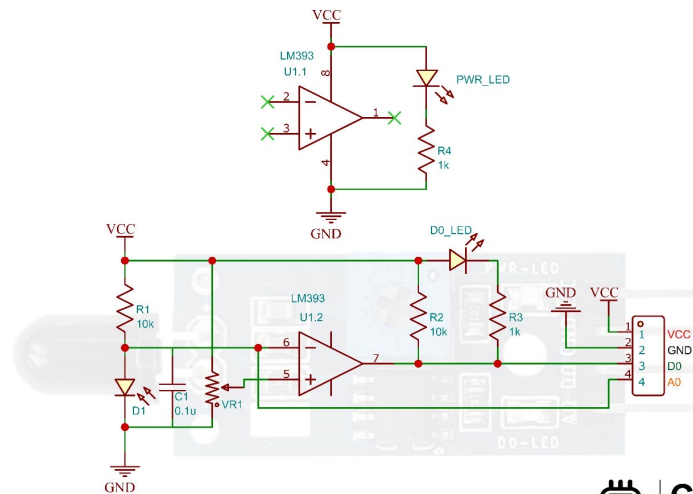
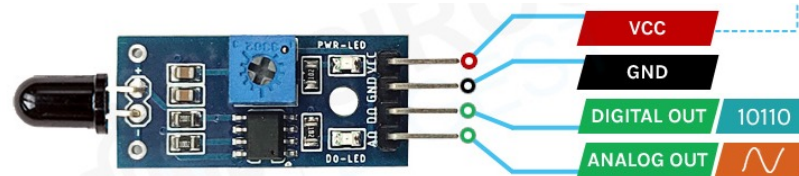
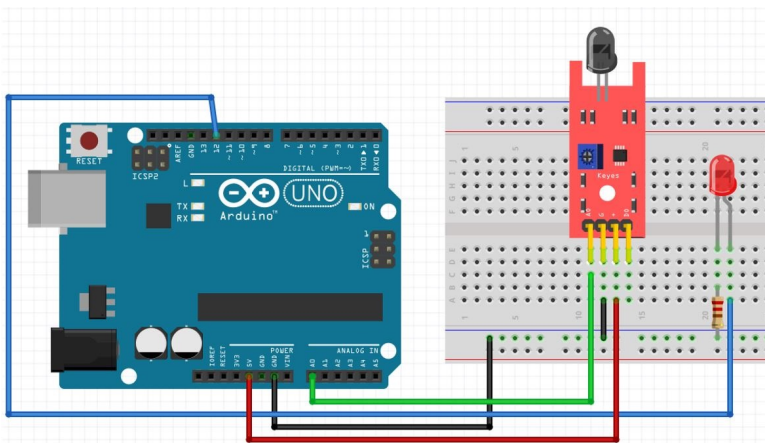


PARTIE
INFORMATIQUE



PARTIE
PHYSIQUE

```
1 int pinSirene = 12; //DEL
2 int pinCapteurDeFeu = A0; //sortie analogue du capteur de feu
3 int valeur = 1023; //valeur analogue du capteur de feu utilisé pour comparer avec la valeur du seuil
4
5 void setup() {
6   pinMode(pinSirene, OUTPUT);
7 }
8
9 void loop () {
10  valeur = analogRead(pinCapteurDeFeu); //lecture analogue
11  if (valeur < 100) { //valeur seuil = 100
12    digitalWrite(pinSirene, HIGH); //DEL = HIGH lorsqu'il y a un feu
13  }
14  else {
15    digitalWrite(pinSirene, LOW); //DEL = LOW lorsqu'il n'y a pas de feu
16  }
17 }
```



Principe:

Mesure l'intensité du rayonnement infrarouge 760-1100 nm

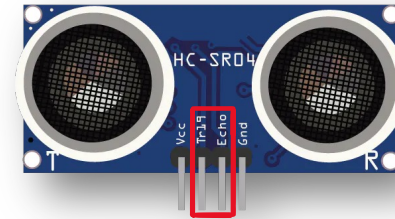
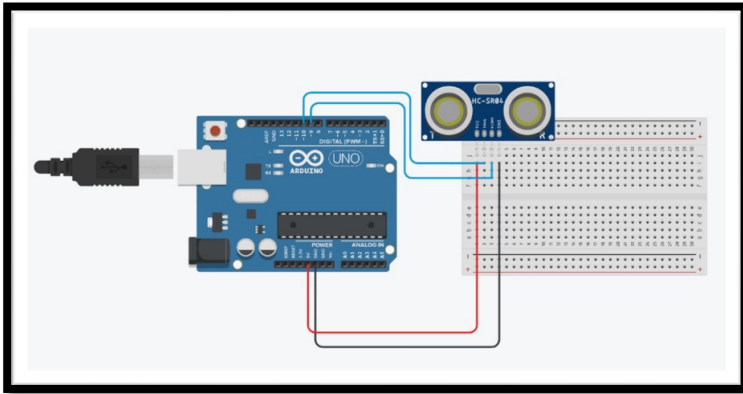
Utilité:

1. Detecter la présence de feu
2. Calculer la position générale d'un feu

Composantes du capteur:

1. Photodiode recouverte d'époxy noir YG1006
2. Comparateur LM393
3. Potentiomètre 10kΩ
4. DEL courant et DEL signal
5. Résistances condensateur céramiques 0.1 uF

Capteur Ultrasonique

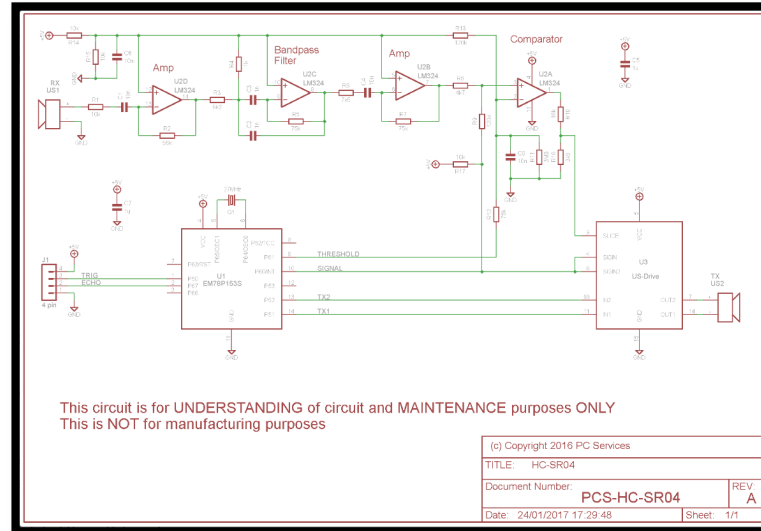


```
//Librairies:
#include "SR04.h"

//Constantes:
const int TRIG_PIN = 9;
const int ECHO_PIN = 10;
const int BUZZER_PIN = 8;
```

```
//Variables:
long distance = sr04.Distance();
```

```
if (distance < 5){
  //Si un obstacle est trop proche, on arrete les moteurs et on allume une sirene
  motorreset();
  siren();
}
else if (distance > 5){
  //Lorsque l'obstacle n'est plus une menace, on arrete la sirene et
  //On laisse les autres conditions s'exécuter (les conditions pour permettre d'éteindre le feu)
  stopsiren();
}
```



Utilité:

- Mesurer la distance des obstacles
- Calcul de vitesse
- **Évitement d'obstacles**

Fonctionnement:

1. **Émission** d'impulsions électriques à l'aide d'un oscillateur de quartz, convertis en ultrasons de 40kHz
2. Les impulsions ultrasoniques sont **réfléchis** sur l'obstacle
3. Les impulsions réfléchis sont **captées**
4. En **multipliant** la vitesse du son par le temps que l'onde prend pour revenir, nous obtenons la distance

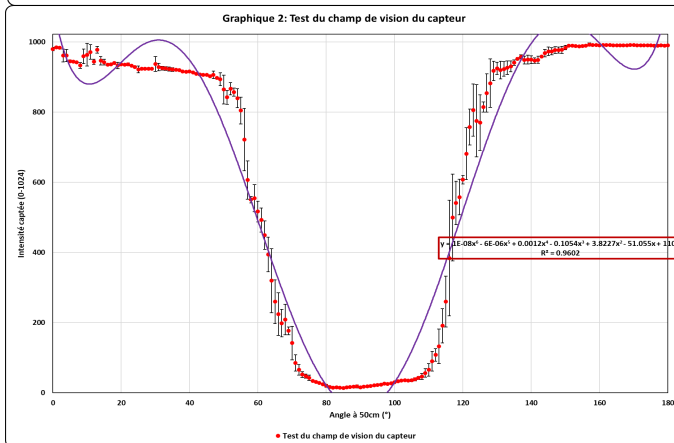
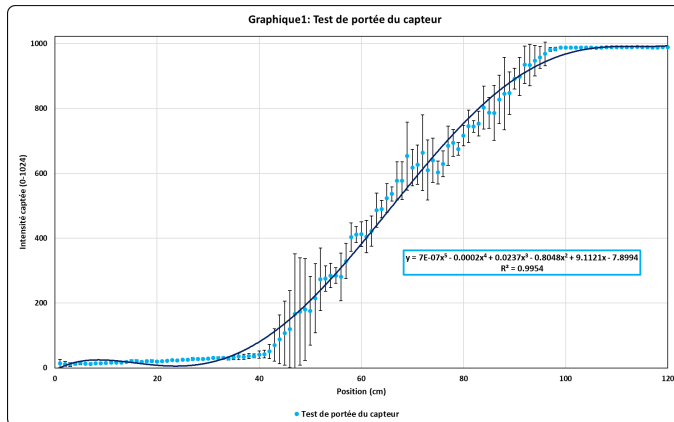
Physique:

- Amplificateur
- Filtre passe-bande
- Comparateur



Test de flamme

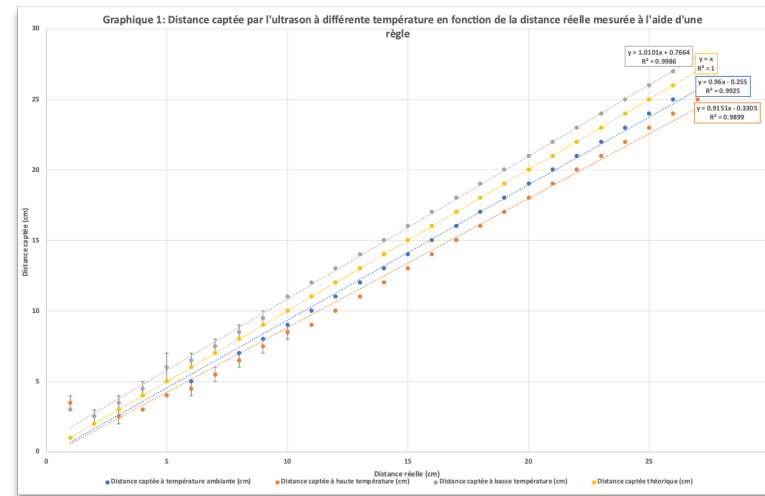
- Tester la portée et le champ de vision du capteur de flamme



- Source d'interférence : le soleil

Test d'ultrason

- Comparer la distance obtenue par le capteur et la distance mesurée par une règle
- Test à haute et basse température
- La vitesse du son change selon la température.

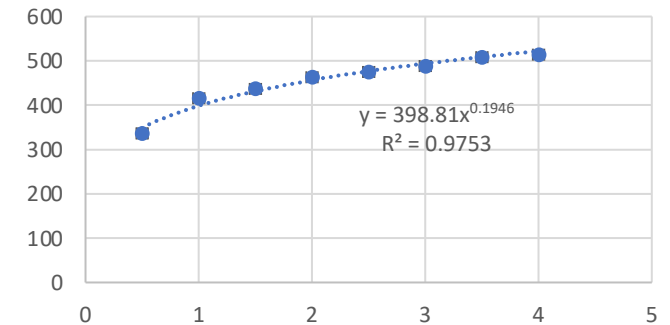


↑Temp ↓Dist || ↓Temp ↑Dist

Test niveau d'eau

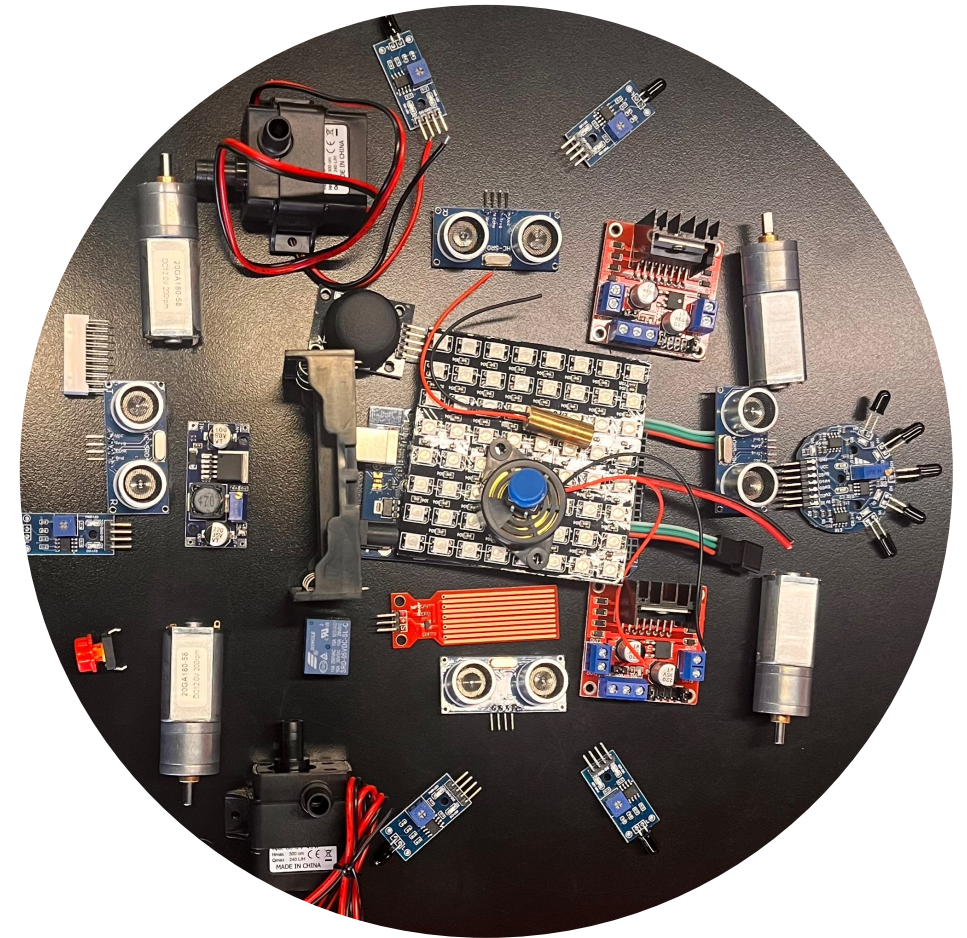
- Tester la valeur qui return sur le serial monitor selon la submersion du capteur.

Graphique #1: valeur du return en fonction de la submersion



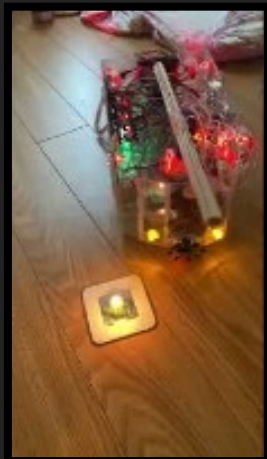
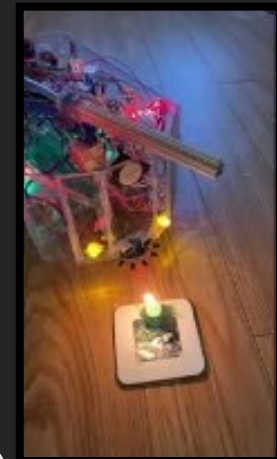
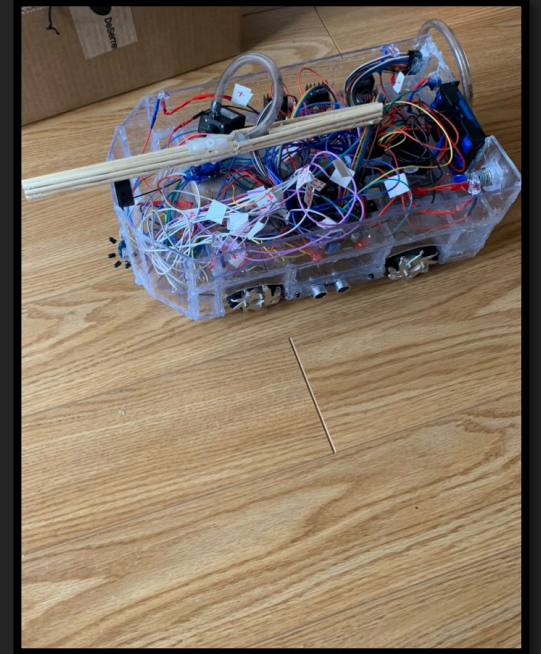
Composantes du camion

1. 4 roues moteurs
2. 1 capteur de niveau d'eau
3. 5 capteurs de flammes
4. 1 capteur de flammes à 5 direction
5. 4 capteurs ultrasons
6. 2 pompes à eau
7. 1 buzzer
8. 1 bouton on/off
9. 1 Arduino méga
10. 1 batterie à 12V
11. 1 servo moteur
12. 3 L298N (module pilote de moteur à courant continu)
13. LM2596 (convertisseur de voltage)
14. Lumières et boutons





Présentation d'un système fonctionnel



Discussion

Facilité:

- Trouver les composantes nécessaire
- Trouver des alternatives pour que le véhicule puisse fonctionner

Difficulté:

- Faire un contenant d'eau qui ne coule pas
- La construction du véhicule était longue et compliqué
- Moteurs défectueux
- Pompes trop faibles
- Capteurs de mauvaises qualités
- Coder avec du matériel de mauvaise qualité



Conclusion

Amélioration:

- Trouver un autre capteur de niveau d'eau.
- Trouver un meilleur capteur de flamme
- Étudier la relation entre le poids et la force des moteurs pour avoir des mouvements plus précis

Conclusion:

- Difficile de créer un véhicule miniature
- Beaucoup d'ajustements de pièce
- Beaucoup de problèmes avec le matériel

