Niveau Zombie

Arduino : Mesures 2

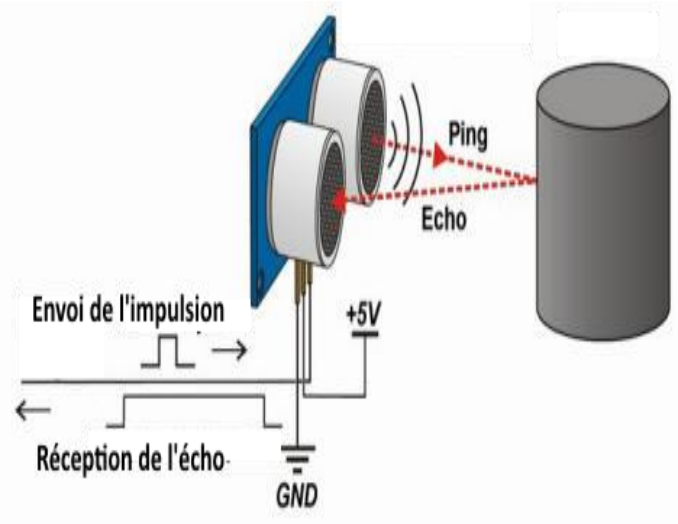
Fiche n°5

Le capteur ultra-son :

On a vu qu'avec un Arduino, il est possible de mesurer des grandeurs telles que la température, la luminosité mais il est également possible de mesurer des distances, notamment grâce à un émetteur récepteur à ultrason, le HC-SR04.

Ce capteur utilise le principe du sonar, le haut-parleur T émet un ultra-son qui va se réfléchir contre l'obstacle et revenir vers le micro R.

Le temps mesuré entre l'envoi de l'impulsion et de sa réception correspond au temps nécessaire pour effectuer un aller-retour jusqu'à l'obstacle. On sait que dans l'air, le son parcourt environ 340 mètres en une seconde, on peut alors déterminer la distance parcourue par le son et en déduire la distance nous séparant de l'objet en divisant par deux cette dernière.



Il faut dans un premier temps se focaliser sur les caractéristiques du capteur :

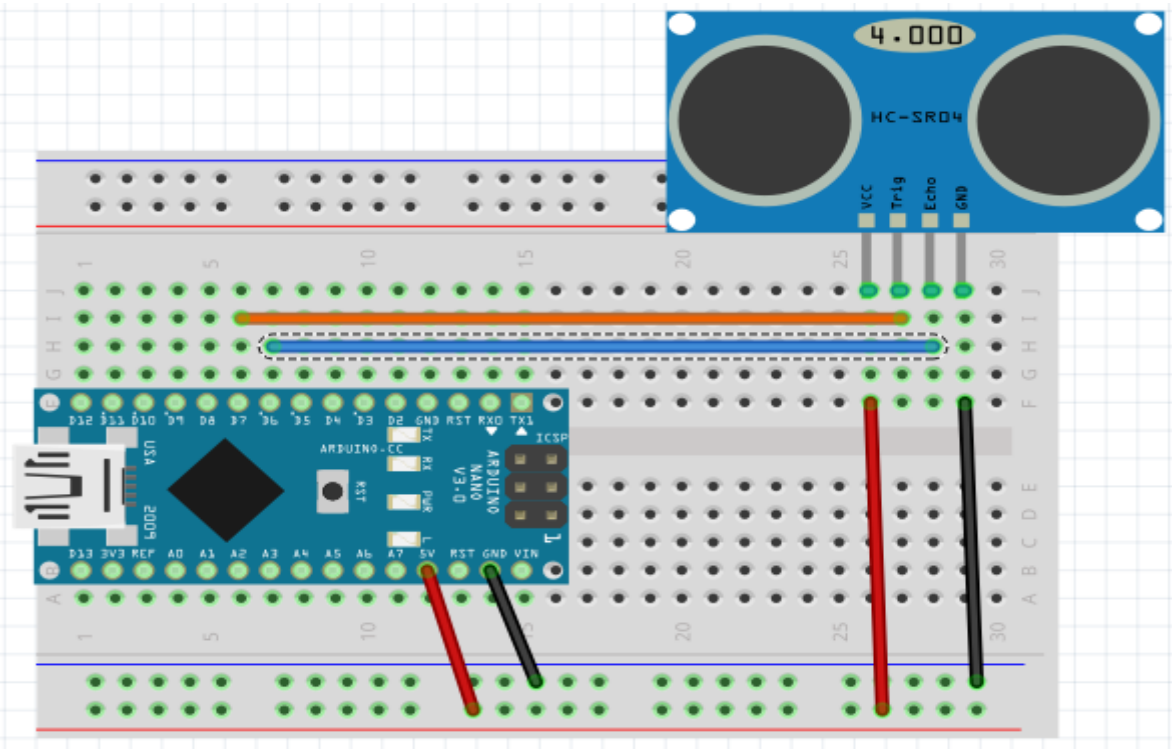
- Distance de lecture de 2 à 450cm
- Tension d'entrée : 5V
- Angle de lecture : 15° maximum
- Signal d'initialisation : 10 µS

Le montage :

Cette étape est la plus facile, le capteur présente quatre broches :

- VCC est l'alimentation électrique du capteur et que nous prendrons directement à la pin 5V de l'Arduino.
- GND est la masse (ground) et qui se prendra sur la pin homonyme de l'Arduino
- TRIG est la pin envoyant le signal (pulsation) ultrason, dans notre exemple on la mettra sur la pin D7.
- ECHO est la pin recevant le signal (pulsation) ultrason, dans notre exemple on la mettra sur la pin D6.

On obtient alors le montage suivant :



Le code :

Le code n'est pas évident mais il sera commenter directement au-dessus de chaque ligne pouvant poser problème au cours de la programmation.

```
1  /* Utilisation du capteur de distance à Ultrason SR04  */
2
3  //Assignation des pins de l'Arduino
4
5  #define echoPin 6
6  #define trigPin 7
7
8  //On initialise les valeurs utiles par la suite
9  int mesure = 0;
10 int total = 0;
11 //La vitesse du son variant en fonction de la température
12 //et de l'atmosphère, elle pourra être réglée  par
13 //étalonnage
14 int vitesse=340;
15
16 void setup()
17 {
18   Serial.begin(9600);
19   //On définit les modes des pins
20   pinMode(echoPin, INPUT);
21   pinMode(trigPin, OUTPUT);
22   //On s'assure que l'émetteur est éteint
23   digitalWrite(trigPin, LOW);
24 }
```

```
26  void loop()
27  {
28    //On définit le temps et la distance pour le SR04
29    long temps, distance;
30    //variable servant à mesurer plusieurs fois la distance
31    //avant d'en afficher la moyenne
32    mesure=mesure+1;
33    //On allume l'émetteur ultrason
34    digitalWrite(trigPin, HIGH);
35    //Durée d'initialisation du capteur
36    delayMicroseconds(10);
37    //Pulsation ultrasonore émise, on éteint l'émetteur
38    digitalWrite(trigPin, LOW);
39    //On définit la pin Echo comme en attente de message
40    pinMode(echoPin, INPUT);
41    //On recoit l'impulsion sur le récepteur
42    temps = pulseIn(echoPin, HIGH);
43    //On définit la distance comme étant le produit de la
44    //vitesse par le temps divisé par deux pour l'A-R.
45    //Le temps étant en microsec(10^-6) et pour avoir d
46    //en cm (10-2), on divise par 10 000 (10^-4)
47    distance = temps * vitesse/(2*10000);
48    //On créé une variable total qui sera la distance totale
49    // pour 10 mesures.
50    total=total+distance;
51    //Si les 10 mesures sont faites :
52    if (mesure==10){
53      //On fait la moyenne des 10 mesures
54      distance=total/10;
55      // on affiche l'information sur le moniteur série:
56      Serial.print("Distance:  ");
57      Serial.print(distance);
58      Serial.println(" cm");
59      //On remet mesure à 0 pour recommencer une série
60      mesure=0;
61      }
62  }
```

Le capteur à ultrason SR04 ne peut être qualifié de capteur numérique mais il pourra être utilisé aussi bien sur une pin numérique qu'analogique.

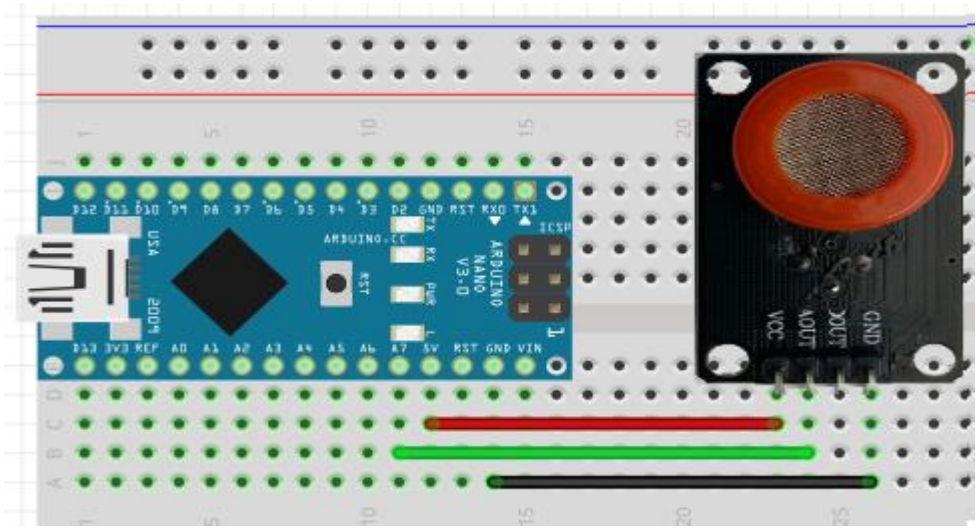
Le capteur de gaz :

Le capteur a besoin d'une résistance de charge à placer entre la sortie et la masse. Sa valeur peut être comprise entre deux et quarante-sept kilo Ohms. Plus sa valeur sera faible et moins le capteur sera sensible mais moins il sera possible de mesurer de fortes concentrations en gaz (saturation). Si seulement un gaz spécifique est mesuré alors la résistance de charge peut-être calibrée en

utilisant d'une concentration connue de ce gaz. Si ce n'est pas le cas, alors la résistance de charge doit être choisie de sorte à ce que la valeur de sortie soit de l'ordre de 1V dans une atmosphère "normale".

Capteur	Sensibilité	Préchauffage		Autre
MQ-2	Méthane, Butane, GPL et fumées	48h	5V	
MQ-3	Alcool, Ethanol et fumées	24h	5V	
MQ-4	Méthane (CH_4).	48h	5V	De 300 à 10000 ppm
MQ-5	Gaz naturel, GPL	24h	5V	De 300 à 50000 ppm
MQ-6	GPL, butane	48h	5V	De 200 à 10000 ppm
MQ-7	Monoxyde de carbone (CO).	48h		De 20 à 2000 ppm. La tension alterne entre 5 et 1.4V(librairie spec)
MQ-8	Hydrogène	24h	5V	De 100 à 10000 ppm
MQ-9	Monoxyde de carbone (CO), méthane (CH_4)			La tension alterne entre 5 et 1.5V. Si seulement CO testé : 1.5V
MQ131	Ozone		6V	La résistance de charge doit être entre 100 et 200 kOhms car les mesures ne sont pas en ppm mais en ppb (partie par milliard).
MQ135	Qualité de l'air	24h	5V	
MQ136	Sulfure d'hydrogène gazeux (H_2S).	24h	5V	De 1 à 1000 ppm
MQ137	Ammoniac.			De 5 à 500ppm
MQ138	Benzène, Toluène, Alcool, Acétone, Propane, Formaldéhyde, Hydrogène	24h		De 10 à 1000ppm sauf pour NH_3, de 10 à 3000 ppm
MQ214	Méthane propane butane GPL	24h		3000ppm à 20000ppm 500ppm à 10000ppm 500ppm à 10000ppm
MQ216	Gaz naturel, gaz de houille, Propane, CH_4	24h		
MQ303A	Alcool, Ethanol, fumées	+ 48h		
MQ306A	GPL, butane	+ 48h		Identique au MQ-6 mais avec une tension de chauffage plus basse (0.9V)
MQ307A	Monoxyde de carbone (CO)	+ 48h		Identique au MQ-7 mais avec une tension variant de 0.2 à 0.9V
MQ309A	Monoxyde de carbone, gaz inflammables	+ 48h		Identique au MQ-9 mais avec une tension variant de 0.2 à 0.9V

Une fois le capteur choisi, si celui-ci ne possède pas de potentiomètre au dos, prenons une résistance de 2kOms et passons au montage :



Le capteur choisi ici possède quatre pins et nous n'en utilisons que trois. La pin inutilisée DOUT est la sortie digitale, c'est-à-dire numérique qui ne donne donc comme unique information : Présence (1) ou Absence (0) de gaz. Nous privilégierons la sortie analogique qui elle permet une précision sur 1024 niveaux de concentration.

Le Code :

Pour le code, il s'agit d'un simple code de lecture d'une pin analogique, ici la A7. Pour ce qui est de la valeur affichée, on pourra choisir d'afficher directement la valeur lue, comprise entre 0 et 1023 ou si on connait la gamme de mesure du capteur, on pourra faire un petit calcul d'après la concentration maximale mesurable. Dans notre exemple du MQ5, nous connaissons la gamme : De 300 à 50000 ppm.

```
/*Lecture entrée analogique*/
int concentration_max=50000;
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  // Lecture de la valeur de la pin A7:
  int sensorValue = analogRead(A7);
  // Conversion de la valeur (0 - 1023) en concentration:
  float concentration = sensorValue * (concentration_max / 1023.0);
  // print out the value you read:
  Serial.print(concentration);
  Serial.println(" ppm");
}
```

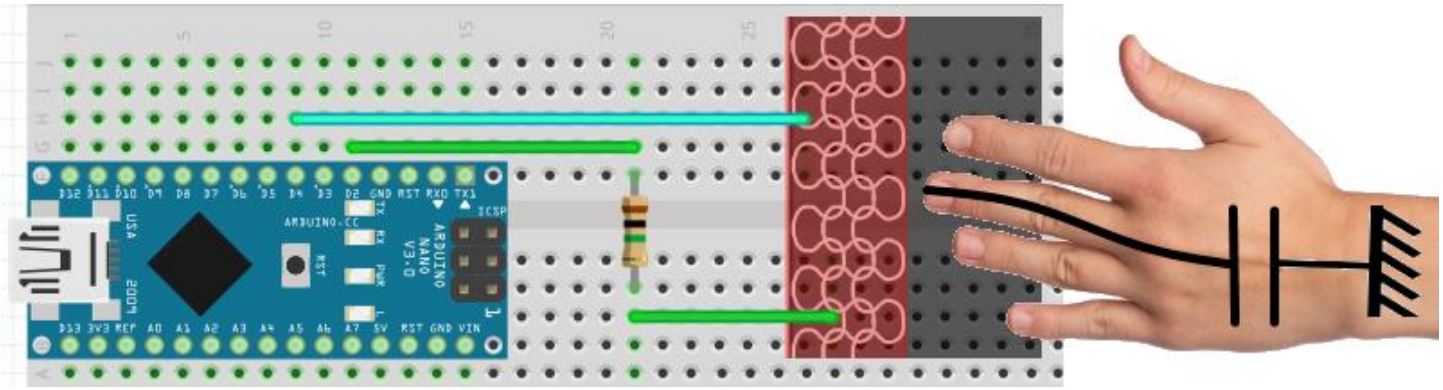
Le bouton inductif :

Le principe du bouton inductif ou bouton tactile est simple notre corps est chargé d'électricité statique, et considéré comme une masse électrique. On place une surface conductrice reliée à l'Arduino et lorsque notre peau entre en contact avec cette surface conductrice, cette électricité statique est captée et le potentiel électrique entre les deux pins change d'un coup.

Dans notre exemple nous pourrons prendre du papier aluminium de cuisine ou une plaque métallique sur laquelle il sera plus facile de souder des fils. Il nous faudra également une très grande résistance, de l'ordre de 100 kilo à 50 Méga Ohms. Cette résistance sera à ajuster lors des tests, plus elle sera élevée et plus le capteur sera sensible mais moins ce dernier répondra vite.

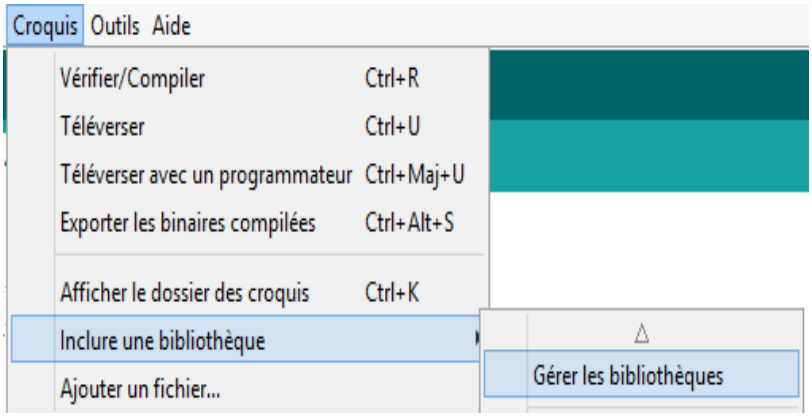
Remarque : Pour plus d'informations vous pouvez effectuer une recherche sur l'oscillateur électrique RC.

On se place entre les pins D2 et D4 afin d'obtenir le montage suivant :

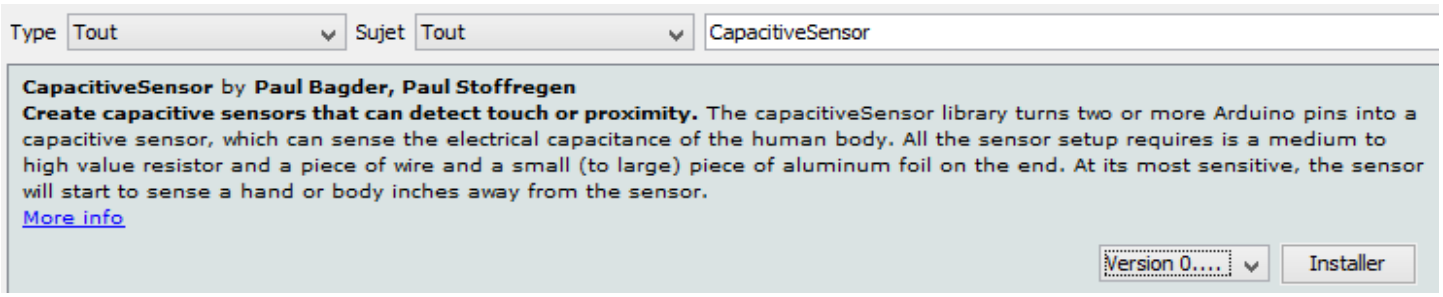


Le Code :

Avant de passer à la programmation à proprement parler et surtout pour se simplifier la vie, nous allons installer une bibliothèque : ***CapacitiveSensor***. Pour cela dans le logiciel Arduino, rendez-vous dans *Croquis* puis *Inclure une bibliothèque* et enfin *Gérer les bibliothèques*.



Dans la ligne de recherche à droite, saisissez le nom de la bibliothèque ***CapacitiveSensor*** puis appuyez sur Entrée. Deux bibliothèques vous sont proposées, choisissez celle des deux Paul, Bagder et Stoffregen :



Vous pouvez alors cliquer sur *Installer*.

Grâce à cette librairie nous n'aurons que très peu de code à saisir pour mesurer la valeur de la capacité électrique. Vous pourrez ensuite l'adapter pour allumer des diodes ou déclencher des actions par exemple. Passons donc au sketch Arduino relatif à la bibliothèque :

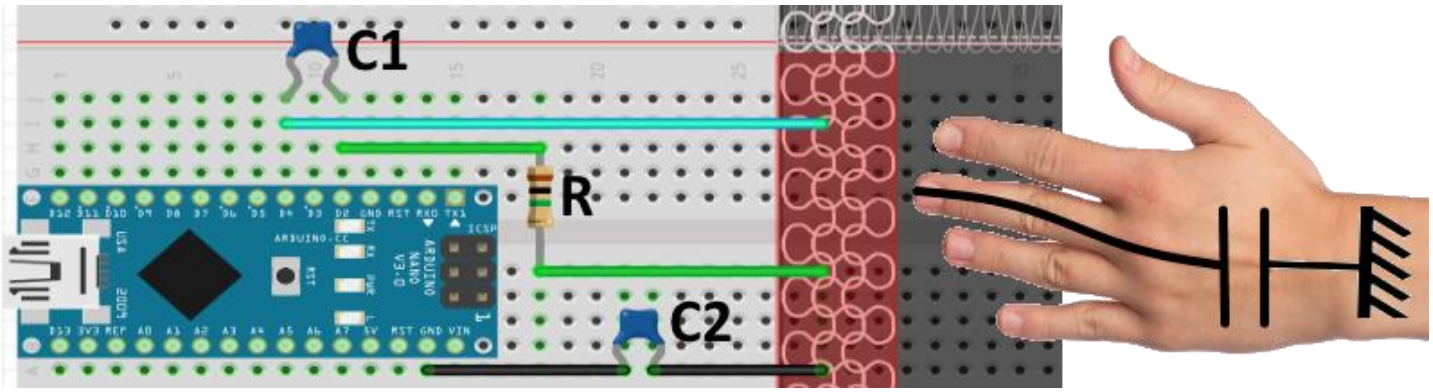
```ino
#include <CapacitiveSensor.h>
//La pin qui envoie le signal est la D4 (sendPin) et
//la variation est lue sur la pin 2 (ReceivePin) :
CapacitiveSensor surface = CapacitiveSensor(4,2);

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

void loop() {
  long valeur = surface.capacitiveSensor(30);
  Serial.println(valeur);
  delay(200);
}
```

Améliorations :

Les expériences ont montré que rajouter des condensateurs permet d'améliorer la stabilité et répétabilité de la mesure.



On rajoute **C1** de l'ordre de 20 - 400 pF en parallèle de la mesure de la capacité de l'individu.

On rajoute **C2** de l'ordre de 100 pF placé entre la masse et la touche tactile.

On peut également remplacer la résistance **R** par un potentiomètre pour effectuer un réglage plus fin de la résistance.