Le radar de recul

**Objectifs :**

* Travailler avec la carte Arduino™ et différents modules
* Comprendre le principe physique de l’écho

**Introduction :**

Le radar de recul est grandement utilisé dans les voitures modernes. Il permet, par un signal sonore, de détecter un obstacle, que ce soit en marche avant ou en marche arrière. Nous allons donc ici réaliser cette fonction.

**Problématique :** Comment peut-on réaliser un radar de recul ?

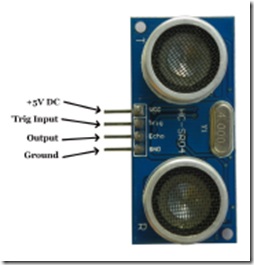
Document n°1 : **Extrait d'une notice de « radar de recul » (aide au stationnement) d’après sujet bac pondichéry 2015**

- En marche arrière le « radar de recul »   
se met en fonction automatiquement.

- L'afficheur indique la distance de   
l'obstacle détecté pour des valeurs   
comprises entre 0,3 m et 2 m.

- L'afficheur dispose d'un buzzer intégré   
qui émet un signal sonore dont la   
fréquence évolue en fonction de la   
distance à l'obstacle.

1. **Préparation du module à ultrason**



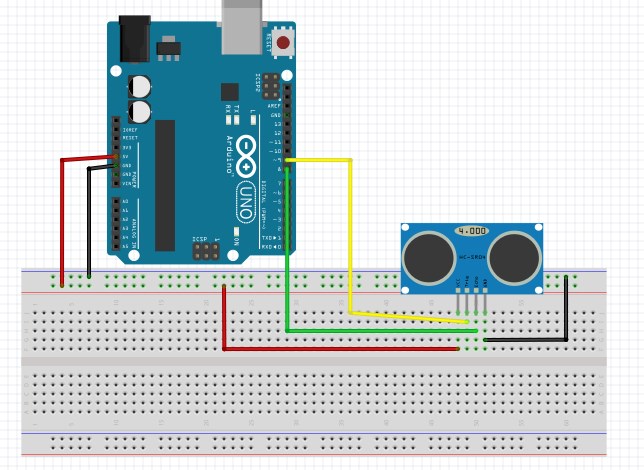
Dans cette partie, vous allez réaliser le câblage et le codage du module à ultrason.

Ce petit module possède 4 broches. Les 2 broches situées sur le côté (VCC et GND) servent à alimenter le module. Il faut donc brancher VCC à la borne +5V et le GND à la borne GND.

Vous avez ensuite la borne trig. Celle-ci va permettre d’envoyer une impulsion (à programmer) d’une onde ultrasonore. Dans le montage, nous allons donc demander à la carte Arduino™ de pouvoir envoyer une information.

La dernière borne écho sert cette fois-ci à recevoir un signal.

Le but ici sera donc de mesurer la durée qu’il s’est écoulé entre l’émission et la réception du signal.

Voici le schéma du montage :

**Travail à réaliser :**

1. Expliquer le principe de la télémétrie sonore qui va s’appliquer ici. Vous pourrez vous aider d’un schéma.
2. Quelle relation a-t-on entre la vitesse de l’ultrason, la distance et le temps qui s’écoule entre l’émission et la réception ?
3. Une fois que le professeur a vérifié votre travail, commencez à réaliser le montage
4. Commenter les différentes étapes du code:

const int Trig = 9;

const int Echo = 8;

void setup() {

pinMode(Trig, OUTPUT);

pinMode(Echo, INPUT);

}

void loop() {

digitalWrite(Trig, HIGH);

delayMicroseconds(10) ;

digitalWrite(Trig, LOW);

unsigned long temps = pulseIn(Echo, HIGH);}

1. Vérifier votre code et le téléverser dans le microcontrôleur.
2. **Préparation du buzzer**

Un buzzer est un élément piezoélectrique. On ne peut pas jouer de mélodie avec.

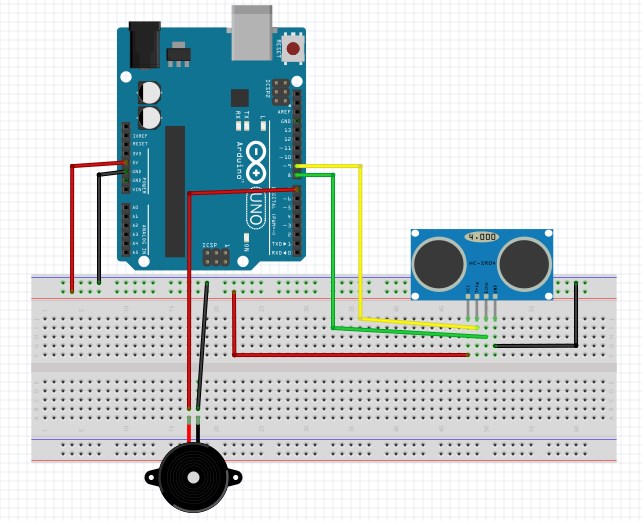
Ce buzzer a une pate plus longue que l’autre. Le pôle + est la patte la plus longue, tandis que le pôle – est la patte la plus courte.

Le branchement de ce composant est très simple. Il faut relier le – au GND et le + à une broche de votre choix (sauf la 3 et la 10)

En ce qui concerne le code, il suffit juste d’indiquer au début à quelle broche est reliée la patte positive.

On utilisera ensuite dans le code la fonction tone(broche, fréquence, Durée). Dans la fonction tone, il faut exactement énumérer 3 paramètres : la broche sur laquelle se trouve la borne + du buzzer, la fréquence du son que vous souhaitez, et la durée d’émission.

Pour le montage, ne débranchez pas ce qui a été utilisé auparavant, n’ajoutez que la partie concernant le buzzer. On va ensuite s’intéresser uniquement au codage du buzzer dans cette partie



**Travail à réaliser :**

1. Utilisez le code suivant :

char Buzzer=7; // On branche le buzzer sur la broche 7

void setup() { }

void loop() {

tone(Buzzer, 16000,500);// On utilise la fonction tone qui nous permet d'avoir un signal sonore tone(n°broche, fréquence de l’onde sonore, durée d’émission de l’onde sonore)

delay(500);// On ajoute un temps entre chaque signal sonore

}

1. Faites varier la fréquence de l’onde sonore d’abord grave, puis aigu.
2. Parmi les fréquences audibles par l’homme, vous choisirez une fréquence en adéquation avec le problème proposé. Vous expliquerez votre choix.
3. **Fonctionnement du radar de recul**

**Travail à réaliser :**

1. Réaliser le montage
2. Copiez-collez le code ci-dessous :

const int Trig = 9; // Le Trig correspond à l'envoi d'une sonore sur la broche 9

const int Echo = 8; // L'écho correspond à la réception de cette sur la broche 8

char Buzzer=7; // On branche le buzzer sur la broche 7

void setup() {

pinMode(Trig, OUTPUT); // On indique que l'on envoie un signal ultrasonore

pinMode(Echo, INPUT); // On indique que l'on va recevoir le signal en retour (écho)

}

void loop() {

digitalWrite(Trig, HIGH); // On va réaliser des impulsions toutes les 10µs

delayMicroseconds(10); //on réalise une impulsion qui dure 10 µs

digitalWrite(Trig, LOW); // On remet a l'état bas la broche Trig

unsigned long temps = pulseIn(Echo, HIGH);// On va mesurer le temps que va durer l'écho

int distance = (temps\*(340))/2; //On utilise la formule d=(v\*t)/2, La vitesse vaut 340 m/s, La durée

// On va réaliser différents types de sons en fonction de la distance

if (distance < 50){

tone(Buzzer, 16000,500);// On utilise la fonction tone qui nous permet d'avoir un signal sonore tone(n°broche, fréquence de l’onde sonore, durée d’émission de l’onde sonore)

delay(500);}// On ajoute un temps entre chaque signal sonore

}

1. Expliquez ce que réalise cette partie du code?

if (distance < 50){

tone(Buzzer, 16000,500);// On utilise la fonction tone qui nous permet d'avoir un signal sonore tone(n°broche, fréquence de l’onde sonore, durée d’émission de l’onde sonore)

delay(500);}// On ajoute un temps entre chaque signal sonore

1. Ajouter des conditions dans le code pour qu’entre une distance comprise entre 50 et 100, vous ayez un son un peu plus grave, entre 100 et 150, un son encore plus grave.