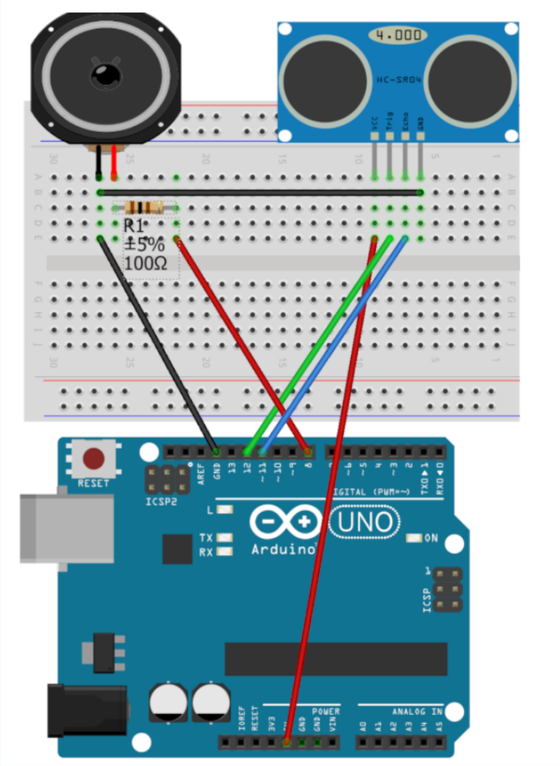
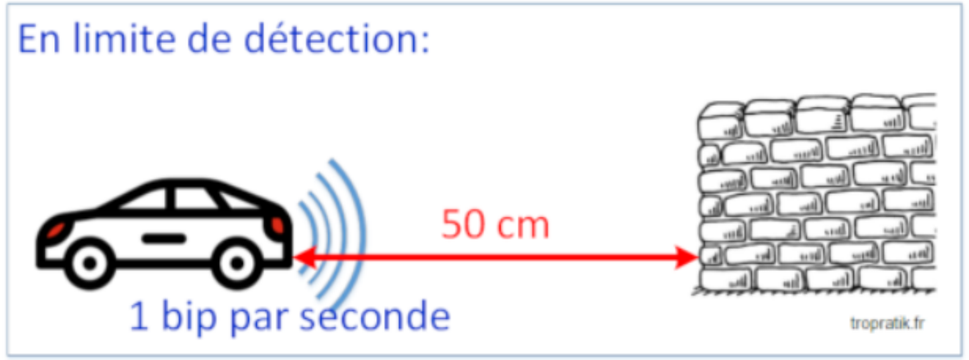
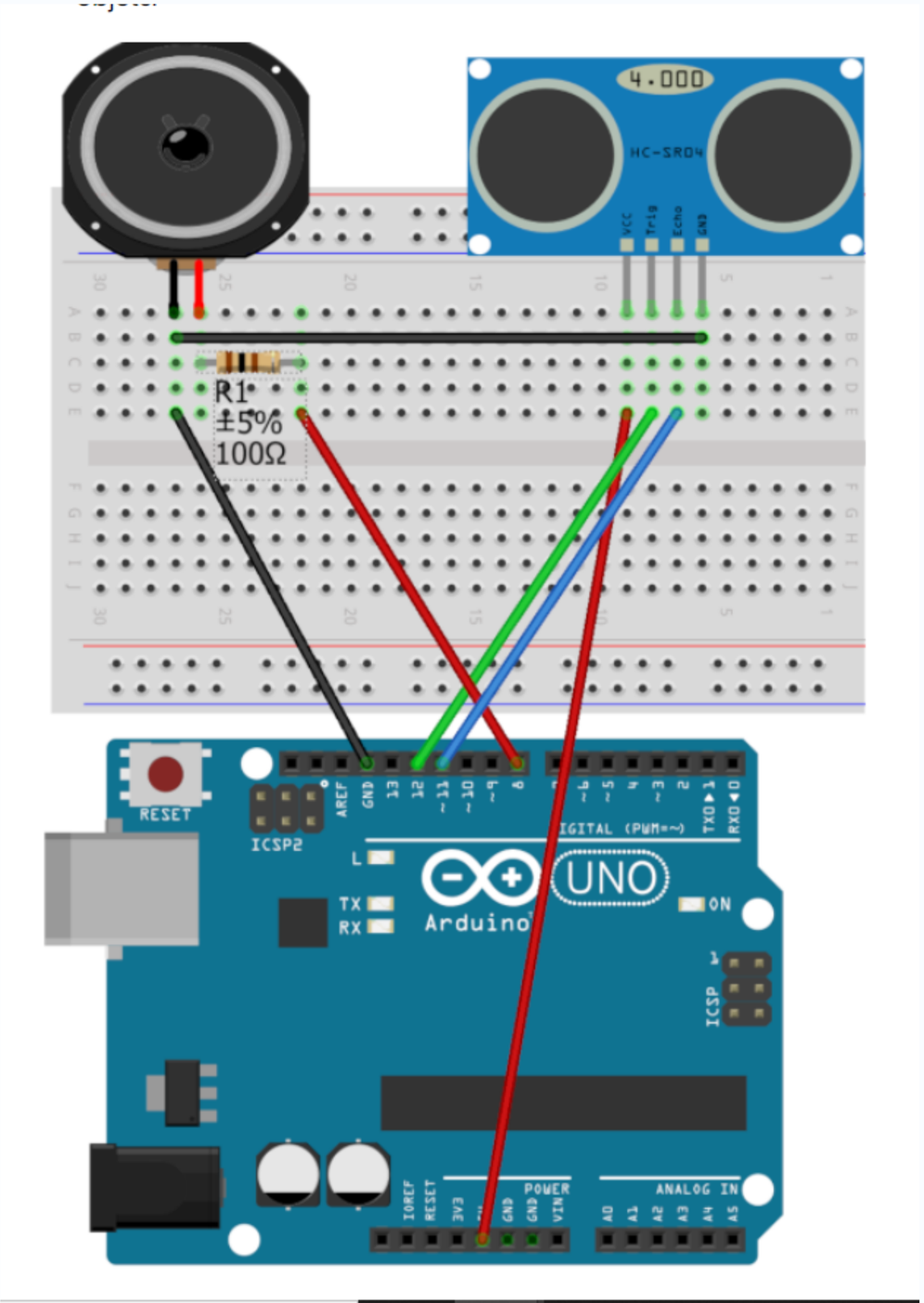
CRÉATION DU RADAR

SUIVI DU TUTO :

https://tropratik.fr/radar-de-recul-avec-arduino



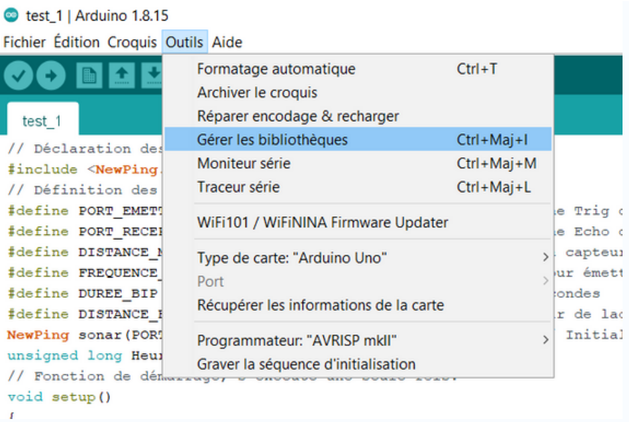


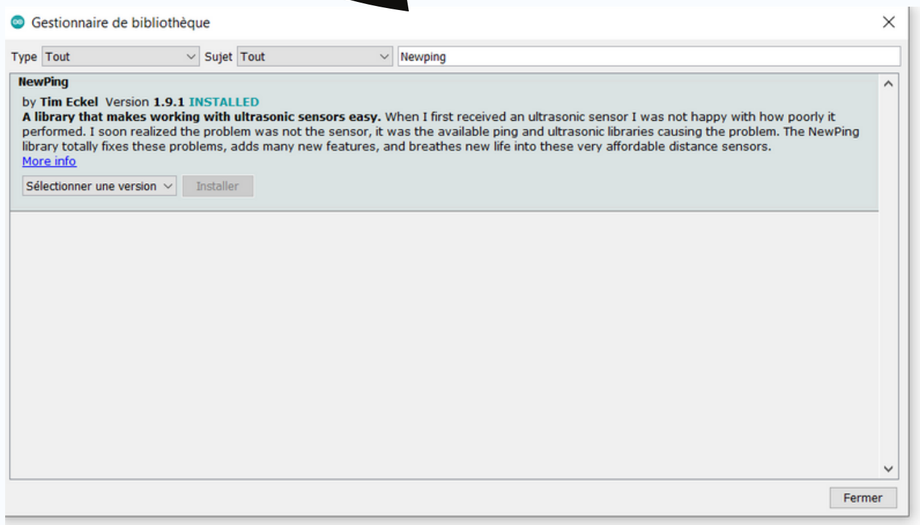

4.000
HC-SR04
VCC
Trig
Echo
GND
R1
±5%
100Ω
RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
Arduino
UNO
ON
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

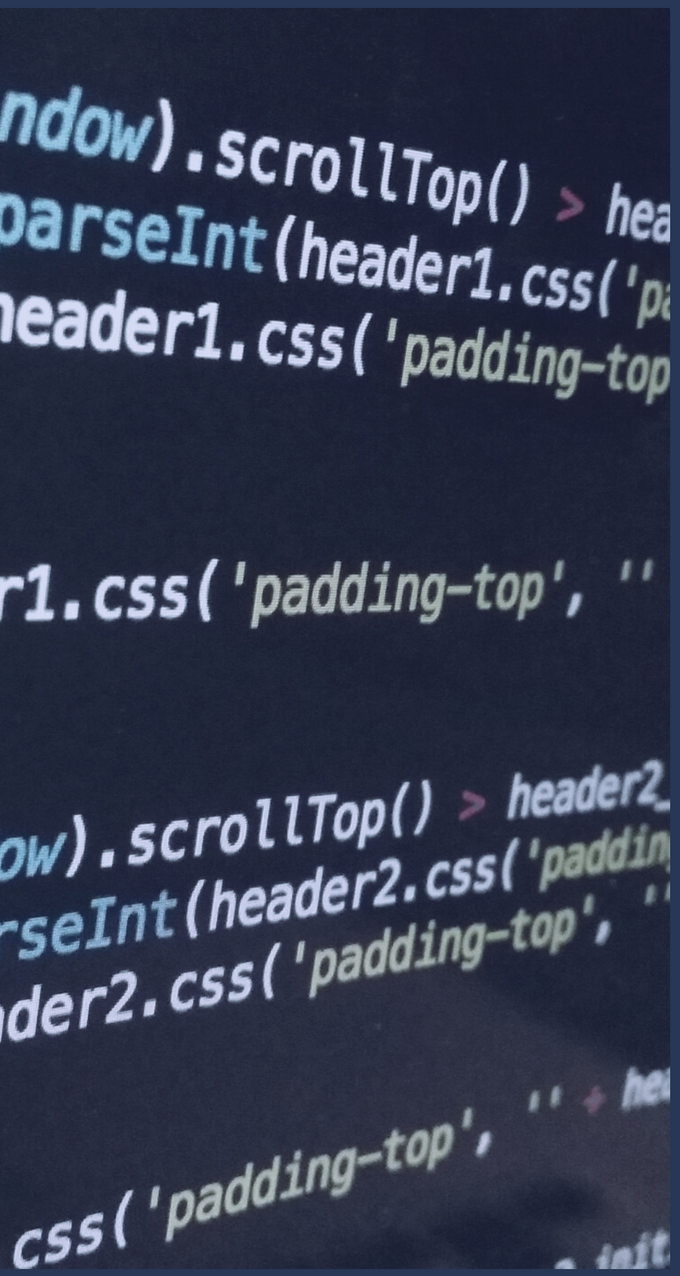
TÉLÉCHARGER UNE NOUVELLE BIBLIOTHÈQUE

Installation d'une bibliothèque pour le capteur HC-SR04

Nous allons utiliser la bibliothèque Arduino **NewPing** nommée **"NewPing by Tim Eckel"** dans ce projet. Je vous invite à l'installer avant de pour suivre.







RENTRER LE CODE

```
// Déclaration des librairies utilisées
#include <NewPing.h>

// Définition des constantes globales
#define PORT_EMETTEUR   12  // Port Arduino branché à la broche Trig du HC-SR04.
#define PORT_RECEPTEUR  11  // Port Arduino branché à la broche Echo du HC-SR04.
#define DISTANCE_MAX 50 // Distance maximale que sait gérér la capteur HC-SR04.
#define FREQUENCE_DU_BIP 784 // Fréquence qui sera utilisée pour émettre les bips sonores
#define DUREE_BIP 250 // Définit la durée d'un bip en millisecondes
#define DISTANCE_BIP_CONTINU 5 // Définit la proximité à partir de laquelle on a un bip continu

NewPing sonar(PORT_EMETTEUR, PORT_RECEPTEUR, DISTANCE_MAX); // Initialisation de la librairie NewPing.
unsigned long HeureDernierBip_G;

// Fonction de démarrage, s'exécute une seule fois:
void setup()
{
  HeureDernierBip_G = 0;
}
// Fonction principale du programme, s'exécute en boucle:
void loop()
{
  // Variables locale de la fonction
  unsigned long DistanceLue_L;
  unsigned long Delai_L;

  // Valeurs par defaut
  Delai_L = 1000; // Par défaut un bip par seconde

  delay(50);  // Attente de 50ms
  DistanceLue_L = sonar.ping_cm(); // Envoi des ultrasons, écoute de leur réflexion, puis calcul de la distance en centimètres.
  if(DistanceLue_L == 0)
  {
     // Pas d'obstacle détecté
    Delai_L = 1000; // Par défaut un bip par seconde
  }
  else if(DistanceLue_L < DISTANCE_BIP_CONTINU)
  {
     // On est proche de la collision
    Delai_L = DUREE_BIP; // bip continu
  }
  else
  {
    Delai_L = (float)DistanceLue_L * (float)(1000 - DUREE_BIP)/(float)(DISTANCE_MAX - DISTANCE_BIP_CONTINU) + DUREE_BIP;
  }

  if(millis()> HeureDernierBip_G + Delai_L)
  {
    HeureDernierBip_G = millis();
    tone(8, FREQUENCE_DU_BIP, DUREE_BIP);
  }
}
```

CORRIGER LE CODE

Pour cela, naviguez vers le dossier “**Documents > Arduino > libraries > NewPing > src**“. Une fois dans ce dossier, ouvrez le fichier “**NewPing.h**” avec un éditeur de code logiciel comme par exemple le logiciel open source “**Notepad++**” disponible ici.

Une fois ce fichier ouvert, recherchez la ligne…

```
    #define TIMER_ENABLED true      // Set to "false" to disable the timer ISR (if getting "__vector_7" compile errors set this to false). Default=true
```

… et remplacez le “**true**” par “**false**“. Enregistrez votre modification et voilà, votre bibliothèque **NewPing** n’utilisera plus le “Timer2”.