

**TÉLÉMÈTRE À ULTRA-SONS****CODE ARDUINO**

```
// Croquis P11-HCSR04.ino

//----- LIBRAIRIE ADDITIONNELLE -----
#include <LiquidCrystal.h> // Librairie pour l'utilisation de l'afficheur LCD

//----- CONSTANTES -----
// création de l'objet lcd (notre afficheur LCD)
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); // RS=8, RW=GND, EN=9, D4=10, D5=11, D6=12, D7=13

#define Buzzer 3 // buzzer sur ligne E/S 3
#define Trig 5 // broche Trig sur ligne E/S 5
#define Echo 6 // broche Echo sur ligne E/S 6

//----- VARIABLES -----

//----- PROCEDURE D'INITIALISATION -----
void setup() {
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  lcd.begin(16, 2); // déclaration du LCD à 16 caractères sur 2 lignes
  lcd.clear(); // Effacement du LCD
  // Message de présentation sur 2 lignes durant 3 secondes
  lcd.setCursor(0, 0); // placement : 1ere colonne, 1ere ligne
  lcd.print(" CLG ST GILBERT"); // on écrit...
  lcd.setCursor(0, 1); // placement : 1ere colonne, 2eme ligne
  lcd.print(" LCD - ARDUINO ");
  delay (3000);
  lcd.clear(); // Effacement du LCD
  // Caractères qui ne seront pas effacés
  lcd.setCursor(1, 0);
  lcd.print("Distance en cm :");
}

//... à suivre...
```



## CODE ARDUINO

```
//... suite...
//----- BOUCLE PRINCIPALE -----
void loop(){
  lcd.setCursor(5, 1);
  lcd.print("      "); // Effacement pour afficher la nouvelle valeur
  float Distance = Mesure(); // Appel de la fonction "Mesure", résultat dans "Distance"
  lcd.setCursor(5, 1);
  lcd.print(Distance,1); // Avec 1 chiffre après la virgule
  tone(Buzzer, (Distance*3)+300, 500); // Emission sonore en fonction de la distance
  delay(500);
}

//----- FONCTION DE MESURE -----
float Mesure(){ // Nom et déclaration de la fonction
  digitalWrite(Trig, LOW); // Forcée au 0 logique (masse)
  delayMicroseconds(2); // Durant 2 microsecondes
  digitalWrite(Trig, HIGH); // Forcée au 1 logique (+5V)
  delayMicroseconds(10); // Durant 10 microsecondes (impulsion haute)
  digitalWrite(Trig, LOW); // Remise au repos au 0 logique
  long Temps = pulseIn(Echo, HIGH); // Lecture de l'impulsion reçue en microsecondes.
  float Longueur = (float)Temps * 0.017; // conversion de la variable Temps en type float
  return(Longueur); // Retour de la valeur de la mesure
}
```



**CODE ARDUINO**

```
// Croquis P11-HCSR04_Q1.ino

//----- LIBRAIRIE ADDITIONNELLE -----
#include <LiquidCrystal.h> // Librairie pour l'utilisation de l'afficheur LCD

//----- CONSTANTES -----
// création de l'objet lcd (notre afficheur LCD)
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); // RS=8, RW=GND, EN=9, D4=10, D5=11, D6=12, D7=13

#define Buzzer 3 // buzzer sur ligne E/S 3
#define Trig 5   // broche Trig sur ligne E/S 5
#define Echo 6   // broche Echo sur ligne E/S 6
#define Led 7    // LED sur ligne E/S 7

//----- VARIABLES -----

//----- PROCEDURE D'INITIALISATION -----
void setup() {
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  pinMode(Led, OUTPUT);
  digitalWrite(Led, LOW);
  lcd.begin(16, 2); // déclaration du LCD à 16 caractères sur 2 lignes
  lcd.clear();      // Effacement du LCD
  // Message de présentation sur 2 lignes durant 3 secondes
  lcd.setCursor(0, 0); // placement : 1ere colonne, 1ere ligne
  lcd.print(" CLG ST GILBERT"); // on écrit...
  lcd.setCursor(0, 1); // placement : 1ere colonne, 2eme ligne
  lcd.print(" LCD - ARDUINO ");
  delay (3000);
  lcd.clear();      // Effacement du LCD
  // Caractères qui ne seront pas effacés
  lcd.setCursor(1, 0);
  lcd.print("Distance en cm :");
}
//... à suivre...
```



## CODE ARDUINO

```
//... suite...
//----- BOUCLE PRINCIPALE -----
void loop(){
  lcd.setCursor(5, 1);
  lcd.print("      "); // Effacement pour afficher la nouvelle valeur
  float Distance = Mesure(); // Appel de la fonction "Mesure", résultat dans "Distance"
  lcd.setCursor(5, 1);
  lcd.print(Distance,1); // Avec 1 chiffre après la virgule
  if (Distance < 20) {
    digitalWrite(Led, HIGH);
  }
  else digitalWrite(Led, LOW);
  //tone(Buzzer, (Distance*3)+300, 500); // Emission sonore en fonction de la distance
  delay(200);
}

//----- FONCTION DE MESURE -----
float Mesure(){ // Nom et déclaration de la fonction
  digitalWrite(Trig, LOW); // Forcée au 0 logique (masse)
  delayMicroseconds(2); // Durant 2 microsecondes
  digitalWrite(Trig, HIGH); // Forcée au 1 logique (+5V)
  delayMicroseconds(10); // Durant 10 microsecondes (impulsion haute)
  digitalWrite(Trig, LOW); // Remise au repos au 0 logique
  long Temps = pulseIn(Echo, HIGH); // Lecture de l'impulsion reçue en microsecondes.
  float Longueur = (float)Temps * 0.017; // conversion de la variable Temps en type float
  return(Longueur); // Retour de la valeur de la mesure
}
```

