

Module d'échographie (HC_SR04)

Résumé

Dans ce projet, nous utiliserons un module d'ultrasons pour mesurer la distance et l'imprimer sur le port série Arduino.

Matériaux

Arduino Uno x 1

HC_SR04 x 1

Un certain nombre de fils Dupont

Description du produit



Ce module mesure la distance avec précision et offre des performances stables. La distance minimum est assez proche à 2cm.

Il peut être utilisé pour l'évitement d'obstacles, la mesure de distance, la mesure de niveau de liquide, la sécurité publique et la détection de stationnement.

spécification

Tension de fonctionnement: DC 5 V

Courant de repos: moins de 2mA

Sortie de niveau électrique: haute - 5V

Niveau de sortie électrique: faible - 0V

Angle de détection: moins de 15 degrés

Gamme de détection: 2cm-450cm

Haute précision: jusqu'à 0.3cm

Principe opérationnel

L'appareil est déclenché par une impulsion de 10µs sur son entrée de déclenchement

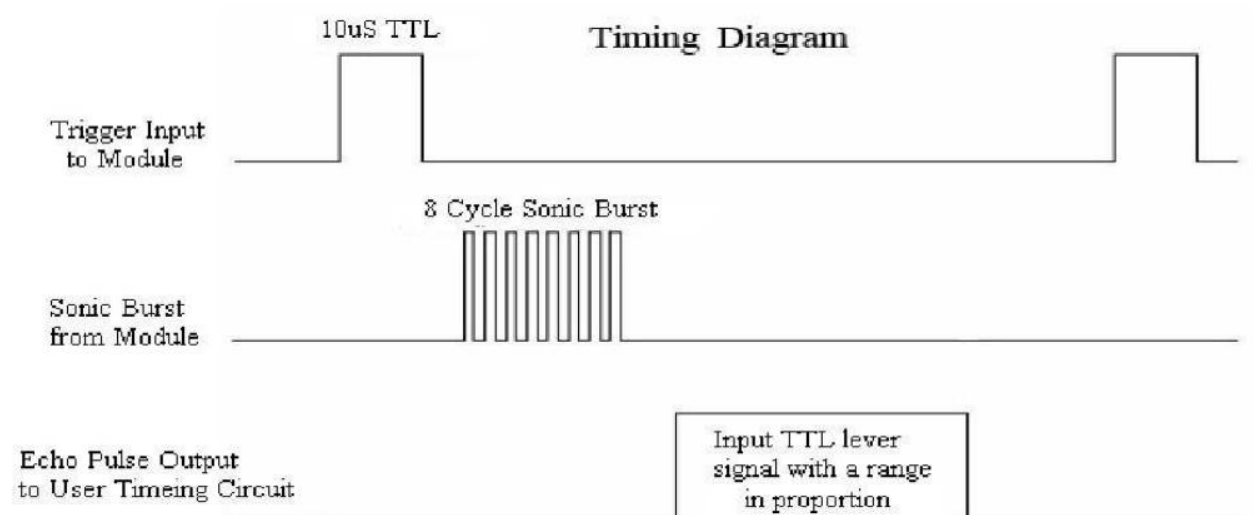
Le module transmet automatiquement des ondes carrées 8x40khz, et détecte si un signal est renvoyé.

Si un signal est renvoyé, il émet une impulsion sur son canal de sortie égale en longueur au temps de l'impulsion ultrasonore de la transmission à la réception.

La distance détectée de l'objet est alors donnée par

$(\text{la période de haut niveau électrique} * \text{vitesse du son (340M / S)}) / 2$

Méthode d'utilisation et diagramme de séquence



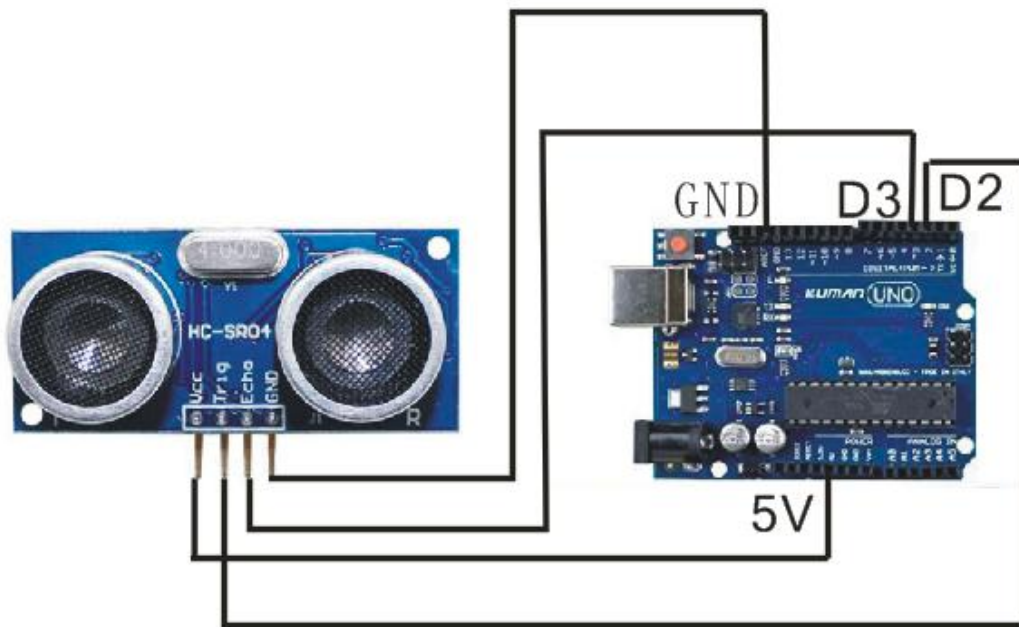
Pour utiliser avec un Arduino, l'Arduino est utilisé pour générer une impulsion de 10µs sur la broche de déclenchement de la SR04 pour démarrer la mesure de distance.

L'appareil déclenchera alors des impulsions ultrasoniques 8x40KHz.

Utiliser Arduino et adopter une broche numérique pour offrir un signal de niveau électrique d'au moins 10µs pour la broche Trig de SR04, pour déclencher la fonction de mesure de distance du module SR04 et écouter les retours détectés.

Si un retour de signal est détecté, la broche de sortie sur le SR04 ira haut pendant le temps de la transmission détectée à retourner. La fonction Arduino `pulseIn()` peut ensuite être utilisée pour mesurer cette durée et utilisée pour calculer la distance réelle à partir de l'objet mesuré.

Schéma de câblage



Exemple de code:

```
// Define the SR04 connection to the Arduino pin
const int TrigPin = 2;
const int EchoPin = 3;
float distance;
void setup()
{   // Initialization of serial communication and SR04's pins
    Serial.begin(9600);
    pinMode(TrigPin, OUTPUT);
    pinMode(EchoPin, INPUT);
    Serial.println("Ultrasonic sensor:");
}
void loop()
{
    // Generate a high pulse of 10us to trigger TrigPin
    digitalWrite(TrigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(TrigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(TrigPin, LOW);
    // Detecting the width of the pulse and calculating the distance
    distance = pulseIn(EchoPin, HIGH) / 58.00;
    Serial.print(distance);
    Serial.print("cm");
    Serial.println();
    delay(1000);
}
```

Remarques

pulseIn (): Fonction de détection de la largeur d'impulsion pour les niveaux haut et bas mesurés sur une broche.

pulseIn (broche, valeur)

pulseIn (pin, valeur, timeout)

Pin --- la broche qui doit lire impulsion.

Valeur --- le type d'impulsion à lire, HAUT ou BAS.

Timeout --- l'unité est la microseconde (µs), le type de données est unsigned long int.

Résultats

