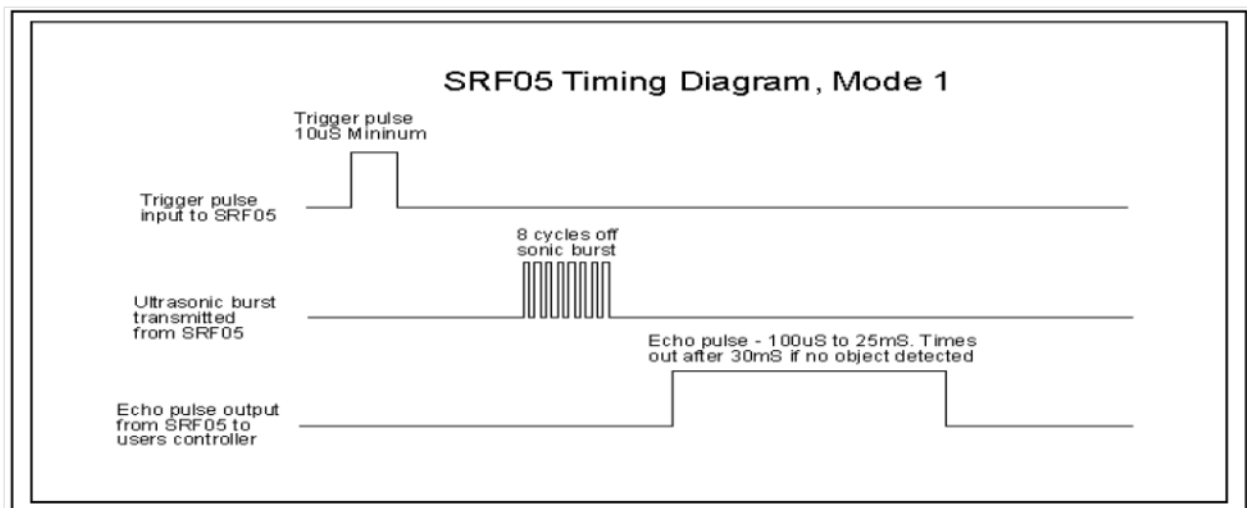
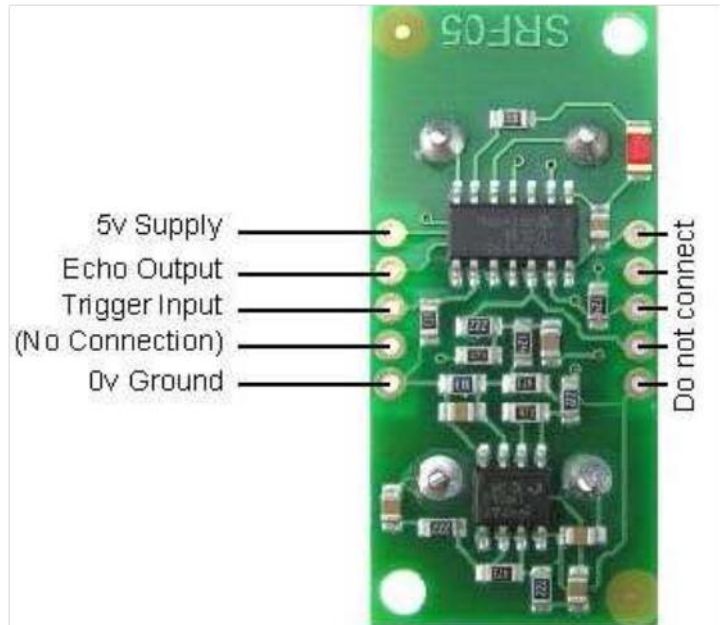


MISE EN OEUVRE DU CAPTEUR A ULTRASONS

Introduction

Le SRF05 est une évolution du SRF04. En soi, le SRF05 est entièrement compatible avec le SRF04. La gamme passe de 3 mètres jusqu'à 4 mètres. Un nouveau mode d'opération permet au SRF05 d'utiliser un seul fil pour le déclenchement et l'écho économisant une e/s sur le microcontrôleur. Quand l'entrée de mode est laissée en l'air. Le SRF05 fonctionne avec E/S séparées de déclenchement et d'écho, comme le SRF04. Le SRF05 inclut un petit retard avant l'impulsion écho pour permettre le fonctionnement avec des microcontrôleurs plus lents.



Calcul de la distance

Les diagrammes de synchronisation SRF05 sont montrés ci-dessus pour chaque mode. Vous devez seulement fournir une impulsion 10µs courte à l'entrée de déclenchement.

Le SRF05 enverra un éclat de 8 cycles d'ultrason à 40kHz et place sa ligne d'écho au niveau 1.

Dès qu'il détectera un écho il abaisse la ligne d'écho au niveau 0.

La ligne d'écho est donc une impulsion dont la largeur est proportionnelle à la distance à l'objet. Si rien n'est détecté le SRF05 abaissera sa ligne d'écho après environ 30ms.

Le SRF04 fournit une impulsion d'écho proportionnelle à la distance. Si la largeur de l'impulsion est mesurée en µs alors une division par 58 donnera la distance en cm. Le SRF05 peut être déclenché aussi rapidement que chaque 50ms, ou 20 fois chaque seconde. Vous devrez attendre au moins 50ms avant le prochain déclenchement, même si le SRF05 détecte un objet étroit et l'impulsion d'écho est plus courte. C'est de s'assurer que le « signal sonore » ultrasonique précédent est terminé et ne causera pas un faux écho sur le prochain.

EXEMPLE DE PROGRAMME ("*ultrason_base*")

Pour un simple affichage dans la console fourni par sketch, on pourra utiliser le code ci-dessous.

```
int trig =12; // signal émis sur broche 12
int echo=11; // signal reçu sur broche 11
int lecture; //
float cm;

void setup() {
  // put your setup code here, to run once:
  pinMode(trig,OUTPUT); // broche trig en sortie
  digitalWrite(trig,LOW); // broche trig à la valeur 0
  pinMode(echo,INPUT); // broche echo en entrée
  Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:

  digitalWrite(trig,HIGH); // émission signal
  delayMicroseconds(10); // signal pendant 10 microsecondes
  digitalWrite(trig,LOW); // arrêt émission signal
  lecture = pulseIn(echo,HIGH); // met la durée de l'impulsion haut
  dans la valeur lecture
  cm=lecture/59; // calcul de la distance de l'obstacle en cm
  Serial.print("distance = "); // Ecrire distance =
  Serial.println(cm); // Ecrire le distance calculée

  delay (1000); // attente 1s
}
```