### 1. Einführende Informationen

Nun wollen wir einen Ultraschallsensor als Entfernungsmesser verwenden anhand der Schallgeschwindigkeit.

Der Schall legt in der Sekunde 330 Meter zurück. Wird die Temperatur mit einbezogen, so sind es 343,5 m/s, also 34350 cm/s, also 0,03434 cm/Mikrosekunde. Wenn man also weiß, wieviel Mikrosekunden es dauert, bis der Ultraschallsensor sein eigenes Echo empfängt, dann kann man leicht ausrechnen, wieviel Meter bzw. Zentimeter der Schall zurückgelegt hat. Diese errechnete Entfernung muss man dann allerdings noch durch 2 teilen, denn wir wollen ja nur die einmalige Wegstrecke berechnen und nicht den gesamten Hin- und Rückweg.

Nun zum Ultraschallsensor HC-SR04:

Der Sensor hat die vier Anschlüsse 5V(+), GND (–), echo und trigger. Die Anschlüsse 5V und GND versorgen den Sensor mit Energie. Der Pin „trigger“ bekommt vom Mikrocontroller-Board ein kurzes Signal (5V), wodurch eine Schallwelle vom Ultraschallsensor ausgelöst wird. Sobald die Schallwelle gegen eine Wand oder einen sonstigen Gegenstand stößt, wird sie reflektiert und kommt irgendwann zum Ultraschallsensor zurück. Sobald der Sensor diese zurückgekehrte Schallwelle erkennt, sendet der Sensor auf dem „echo“ Pin ein 5V Signal an das Mikrocontroller-Board. Dieses misst dann lediglich die Zeit zwischen dem Aussenden und der Rückkehr der Schallwelle und rechnet diese Zeit in eine Entfernung um.

Nun zum seriellen Monitor:

Die Anzeige der Entfernung soll auf dem seriellen Monitor erscheinen. Du hast ihn schon auf dem Arbeitsblatt M3 kennengelernt. 3 Befehle sin für den seriellen Monitor wichtig:

|  |  |
| --- | --- |
| Serial.begin (9600); | Die serielle Kommunikation wird gestartet, damit man sich die Werte auf dem seriellen Monitor ansehen kann. Dieser Befehl steht am Anfang des Loop-Bereichs. |
| Serial.print(entfernung); | Der Wert der Entfernung wird auf dem seriellen Monitor ausgegeben, der Cursor bleibt in der Zeile. Dieser Befehl steht am Ende des Loop-Bereichs. |
| Serial.println(" cm"); | Hinter dem Wert der Entfernung wird am seriellen Monitor die Einheit "cm" angegeben. Dieser Befehl steht am Ende des Loop-Bereichs. |

### 2. Aufgabe

Wir wollen ein Programm erstellen, das mit Hilfe eines Ultraschallsensors die Entfernung zu einem Gegenstand misst und auf dem seriellen Monitor in cm ausgibt.

Erfasse dazu den Sketch mit Variablen und speichere ihn ab.

Notiere in der Tabelle die fehlenden Beschreibungen zum Sketch.

Färbe alle Befehle rot, die den Ultraschallsensor programmieren.

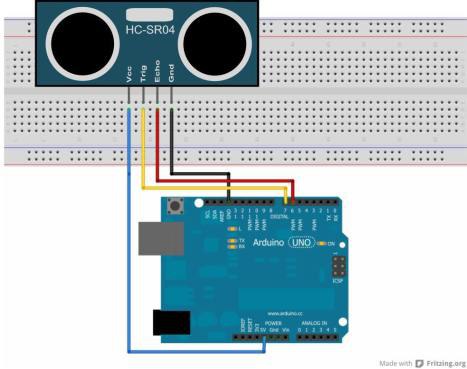
Färbe alle Befehle blau, die den seriellen Monitor programmieren.

### 3. Notwendiges Material

Breadboard, Arduino, ein Ultrschallsensor, Kabel

### 4. Tipp: Schaltung

Der Ultraschallsensor wird mit seinen 4 Beinchen auf das Breadboard aufgesteckt und hier mit 5V, GND und den Pins 6 und 7 per Kabel verbunden.



### 5. Tipp: Sketch

|  |  |
| --- | --- |
|  | Beschreibungen |
| int trigger=7; |  |
| int echo=6; |  |
| long dauer=0; |  |
| long entfernung=0; | Unter der Variablen entfernung wird die berechnete Entfernung gespeichert . Anstelle von „int“ steht hier vor den beiden Variablen „long“. Das hat den Vorteil, dass eine größere Zahl gespeichert werden kann. Nachteil: Die Variable benötigt mehr Platz im Speicher. |
| void setup() { |  |
| Serial.begin (9600); |  |
| pinMode(trigger, OUTPUT); |  |
| pinMode(echo, INPUT); |  |
| } |  |
| void loop(){ |  |
| digitalWrite(trigger, LOW); | Die Spannung wird für kurze Zeit vom Trigger-Pin genommen, damit man beim Senden des Trigger-Signals ein rauschfreies Signal hat. |
| delay(5); |  |
| digitalWrite(trigger, HIGH); | Es wird eine Ultraschallwelle gesendet. |
| delay(10); |  |
| digitalWrite(trigger, LOW); | Der „Ton“ wird abgeschaltet. |
| dauer = pulseIn(echo, HIGH); | Der Mikrokontroller zählt die Zeit in Millisekunden, bis der Schall zum Ultraschallsensor zurückkehrt. |
| entfernung = (dauer/2) \* 0.03432; |  |
| Serial.print(entfernung); |  |
| Serial.println(" cm"); |  |
| delay(1000); } |  |