

# Funktionen laufen

## Bastelanleitung

### Benötigte Hardware:

- Arduino (Nano ist ausreichend)
- Ultraschallsensor HC-SR04
- Breadboard
- 4 Jumperkabel ausreichender Länge
- USB-Kabel (passend zum Arduino-Modell)

### Hinweis

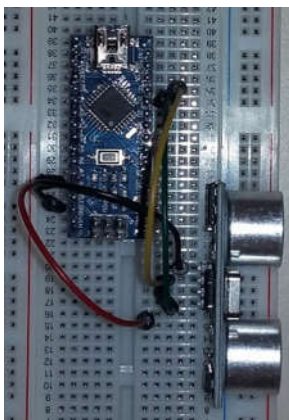
Zur Durchführung wird ein Brett ausreichender Größe benötigt, da der oben genannte Sensor Menschen nur schwer detektieren kann.

### Aufbau

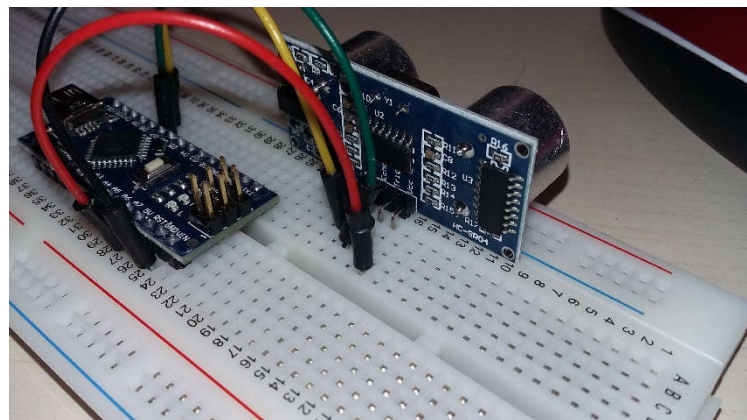
Der Aufbau ist recht einfach. Stecken Sie den Arduino so auf das Breadboard, dass keine Kontakte kurzgeschlossen werden. Stecken Sie den Ultraschallsensor ebenfalls so auf das Breadboard, dass keiner seiner Pins einen direkten Kontakt zu einem Arduino-Pin hat (zur Verdrahtung siehe: <http://l.aripe.de/breadboard>). Verbinden Sie nun mit den Jumper-Kabeln folgende Pins.

Pin am Sensor	Pin am Arduino
Vcc	5V
Trig	D8
Echo	D9
Gnd	GND

Der Aufbau sollte in etwa so aussehen:



Fertiger Aufbau



Detailaufnahme: Verdrahtung am Sensor

Diese Verdrahtung stellt sicher, dass der Arduino-Sketch funktioniert. Der Sketch sowie das Python-Skript für den PC können Sie unter <http://l.aripe.de/Indw> herunterladen.

## Hinweise zur Durchführung

Das PC-Programm besteht aus zwei Teilen:

1. Ein Programm zur Kalibrierung
2. Das eigentliche „Spiel“

Es lief sowohl auf einem Linux- als auch auf einem Windows-Rechner und sollte auch auf einem Mac laufen.

### Software

Sie benötigen zur Programmierung

- die Arduino-IDE (<https://www.arduino.cc/en/main/software>) sowie
- einen Python-Interpreter (<https://www.python.org/downloads/>, nehmen Sie einfach die letzte stabile Version).

### Vorbereiten des Arduino

Öffnen Sie den Arduino-Sketch (dst.ino) in der Arduino-IDE. Verbinden Sie den Arduino mit dem USB-Kabel.

- Wählen Sie unter **Werkzeuge > Port** den Port aus, mit dem der Arduino verbunden ist (meist kann man nur einen wählen).
- Wählen Sie unter **Werkzeuge > Board** die Arduino-Bauart, die Sie verwenden.
- Laden Sie den Sketch auf den Arduino, indem Sie den markierten Button drücken:



- Testen Sie unter **Werkzeuge > Serieller Monitor**, ob das Skript auch funktioniert, es müsste eine Zeichenkette mit der Entfernung in cm anzeigen.

### Eingabeaufforderung/Terminal

Alle Befehle, die im Folgenden genannt werden, müssen Sie in ihrer Eingabeaufforderung (Windows) bzw. im Terminal (Linux/Mac) eingeben.

### Benötigte Module

Sie benötigen zwei Module, die nicht Teil der Python-Standard-Installation sind. Geben Sie daher folgende Befehle ein:

```
python -m pip install PyQt5
python -m pip install pyserial
```

Dies muss bei jedem verwendeten Computer einmalig erfolgen. Die Module bleiben installiert, auch wenn Sie ihn ausgeschaltet haben.

## Kalibrierung

Im Laufe des Spiels benötigt das Programm folgende Informationen:

- Port der seriellen Schnittstelle, über die Arduino und PC kommunizieren
- Anzahl der Schritte (maximaler Funktionswert)
- Abstand der 0-Marke vom Sensor
- Abstand des maximalen Funktionswerts vom Sensor.

Starten Sie deswegen **einmalig** vor dem Spiel das Skript kalibrierung.py, gehen Sie dazu mit der Eingabeaufforderung (Windows) bzw. mit dem Terminal (Linux) in das Verzeichnis, in dem Sie die Python-Skripte (.py-Dateien) ablegten. Sie starten das Skript mit dem Befehl:

```
python kalibrierung.py
```

Nach Wahl des Ports können Sie die Anzahl der Schritte wählen. **Das aktuelle Skript geht bei den Funktionsbeschreibungen der Zielfunktion davon aus, dass Sie 5 Schritte wählen. Es wird in Kürze eine Anpassung erfolgen, in der diese Einschränkung behoben wird.** Stellen Sie nun den Abstand zum 0-Punkt ein, indem Sie den Sensor eine entsprechende Strecke messen lassen und die Taste „Minimum“ drücken. Der aktuelle Abstand in cm wird in einem großen Font angezeigt. Verfahren Sie ebenso mit dem maximalen Funktionswert, indem Sie einen entsprechenden Abstand messen lassen, dann jedoch die Taste „Maximum“ drücken.

Es ist unerheblich, ob das Minimum oder das Maximum näher am Sensor liegt. In den meisten Fällen ist es sogar ratsam, das Maximum näher an den Sensor zu postieren, weil man dann auf den meist neben dem Sensor stehenden Rechner zuläuft, um einen steigenden Graphen zu erhalten. Das Skript kommt mit beiden Situationen zurecht. **Beenden Sie das Kalibrierungsprogramm auf jeden Fall mit der Taste „Kalibrierung beenden“, erst dann werden die Daten in eine Datei geschrieben.**

## Spiel starten

Das eigentliche Spiel starten Sie mit

```
python starten.py
```

Da das Spiel auch im Kiosk-Modus laufen soll, kann es regulär nur durch eine bestimmte Tastenkombination abgebrochen werden. Drücken Sie daher in folgender Reihenfolge die Tasten <F10> <q> und <x> (die eckigen Klammern sollen die Tasten symbolisieren). Lassen Sie beim Drücken der Tasten die vorherigen gedrückt („Affengriff“).

### Selbst Zielfunktionen schreiben

Wenn Sie eine der Dateien `f1*.txt` bzw. `f2*.txt` ändern, können Sie Funktionen des ersten bzw. zweiten Levels ändern. In diesen Textdateien sind die Funktionen in ihren kritischen Punkten definiert, z.B.

```
0.0,2.3  
3.4,5.0  
8.0,0.5  
10.0,0.5
```

Beispiel für eine Funktionsdatei

Diese Funktion steigt von (0;2,3) linear an auf (3,4;5). Von dort fällt sie linear auf (8;0,5), um dann konstant auf dem Wert 0,5 zu bleiben. Wichtig ist, dass der erste Wert des Definitionsbereichs 0,0 ist und der letzte 10,0. Ferner sollten die Punkte nach ihren x-Koordinaten aufsteigend sortiert sein. Eine Doppeldefinition ist selbstredend nicht zulässig.

Derzeit müssen beide Levels gleich viele Funktionen haben und die Anzahl der Funktionen pro Level ist in den vorhandenen Skripten auf drei festgelegt. Wenn Sie dies ändern wollen, müssen Sie die Datei `starten.py` an einer Reihe von Stellen anpassen.