

Versuch 2: Abstandsmessung

[Arduino UNO](#)[Arduino IDE](#)[Version Control System Git](#)[Microsoft VSC](#)[Adobe PDF](#)[Download zip](#)

Inhaltsverzeichnis

1. [Wie funktioniert der Ultraschallsensor](#)
2. [Versuchsaufbau](#)
3. [Aufgabenstellung](#)
4. [Literaturverzeichnis](#)

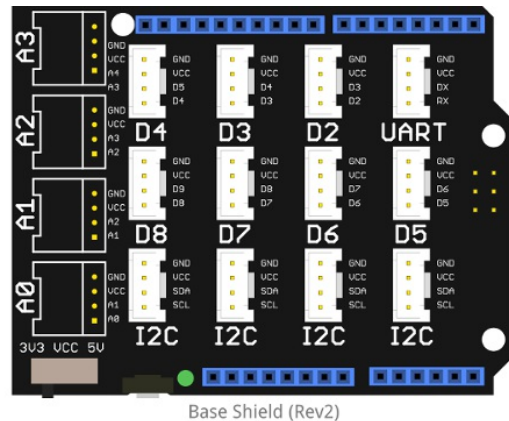
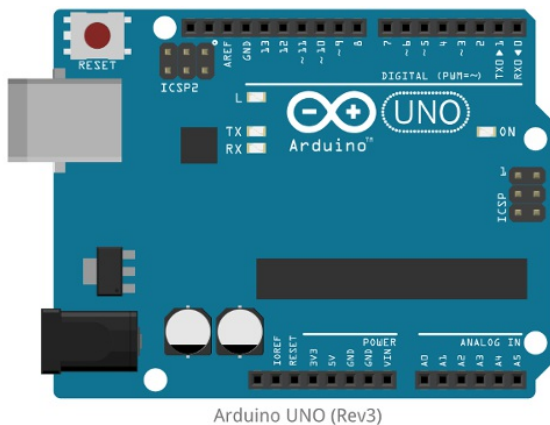
Hinweis: Wenn Sie dieses Repository downloaden möchten, können Sie es direkt als [ZIP](#)-Datei herunterladen, oder mit Hilfe des Git-Repository-Links auf Ihren Rechner klonen.

 **Git-Repository-Link:**

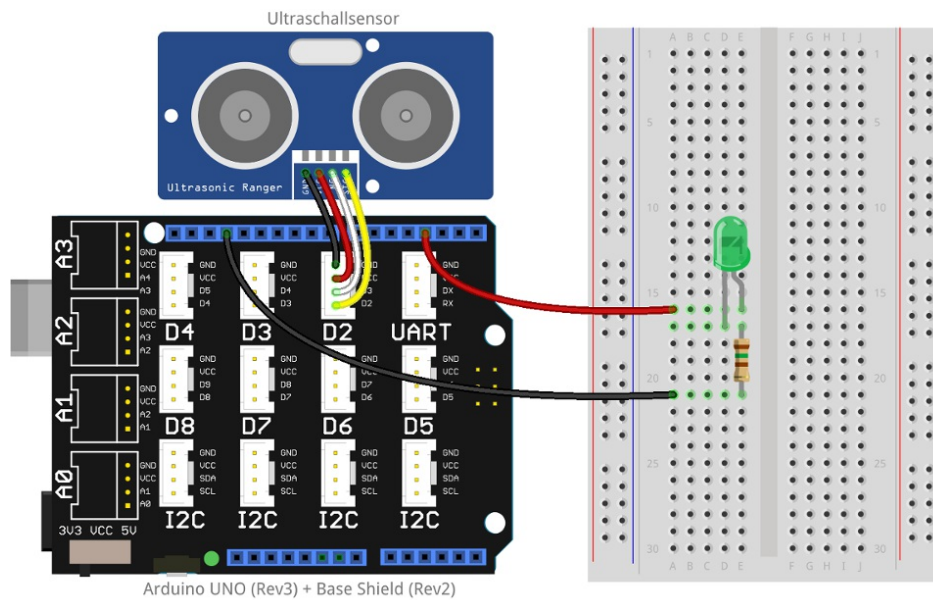
<http://es-lab.de:60085/Maschinenprogrammierung/V2.Abstandsmessung>

2. Versuchsaufbau

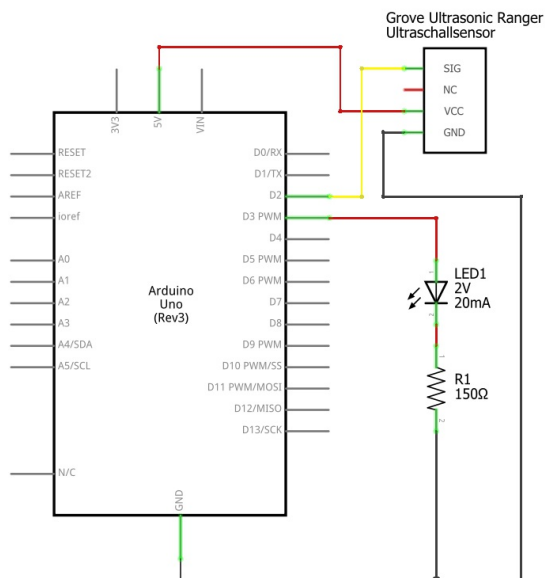
Im Praktikum bekommen Sie ein Base Shield, Breadboard und Verbindungskabel, sowie die Bauelemente: Widerstand und LED, ausgehändigt. Das Base Shield ist eine Adapterplatine und wird auf den Arduino UNO gesteckt. Die Adapterplatine verknüpft die GPIO-Pins des Arduino UNO zu entsprechenden Sensor-Anschluss terminals. Über diese können später Sensoren, wie der Ultraschallsensor, angeschlossen werden.



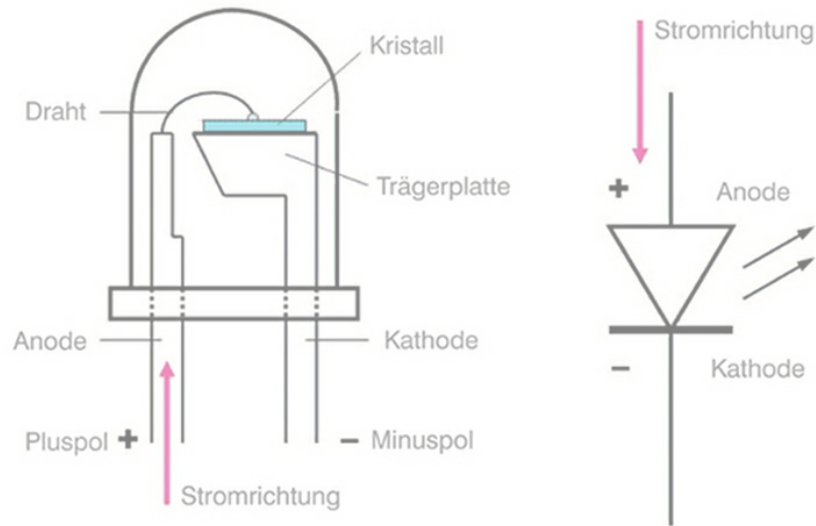
Stecken Sie das Base Shield auf den Arduino. Die ausgehändigten Bauteile sind wie im unten stehenden Bild aufzubauen bzw. anzuordnen. Der Ultraschallsensor wird mit dem Port **D2** und die LED mit dem GPIO-Pin **3** verbunden.



Der unten stehende Schaltplan verdeutlicht den Versuchsaufbau.



Beachten Sie beim Aufbau die Polarität der LED. Anderenfalls wird die LED nicht leuchten. Die unten stehende Abbildung verdeutlicht den Aufbau der LED sowie dessen Polarität bzw. Stromrichtung.



Bildquelle: [http://letsgoingwiki.reutlingen-](http://letsgoingwiki.reutlingen-university.de/mediawiki/images/thumb/8/88/Sensor_CH13_LED_aufbau.jpg/800px-Sensor_CH13_LED_aufbau.jpg)

[university.de/mediawiki/images/thumb/8/88/Sensor_CH13_LED_aufbau.jpg/800px-Sensor_CH13_LED_aufbau.jpg](http://letsgoingwiki.reutlingen-university.de/mediawiki/images/thumb/8/88/Sensor_CH13_LED_aufbau.jpg/800px-Sensor_CH13_LED_aufbau.jpg)

⚠ 3. Aufgabenstellung

Sie sollen ein Programm schreiben, welches den ermittelten Abstand vom Ultraschallsensor erfasst und diesen über den seriellen Monitor ausgibt. Verwenden Sie für das Programm die entsprechende Programmvorlage im Repository. Des Weiteren soll eine LED in Abhängigkeit des Abstandes angesteuert werden. Die LED soll dabei mit sinkendem Abstand immer schneller blinken und bei einem sehr kurzen Abstand dauerhaft leuchten. Die genauen Blinkfrequenzen in Abhängigkeit des Abstandes entnehmen Sie den nachfolgenden Bedingungen:

Distanz > 300 cm → 0,5 Hz,
 200 cm < Distanz ≤ 300 cm → 1 Hz,
 150 cm < Distanz ≤ 200 cm → 2 Hz,
 100 cm < Distanz ≤ 150 cm → 5 Hz,
 50 cm < Distanz ≤ 100 cm → 10 Hz,
 30 cm < Distanz ≤ 50 cm → 25 Hz,
 10 cm < Distanz ≤ 30 cm → 50 Hz,
 Distanz ≤ 10 cm → LED leuchtet dauerhaft.

Des Weiteren sollen Sie sich mit der Erstellung einer eigenen Unterfunktion beschäftigen. Diese Unterfunktion soll Ihnen einen Delay-Wert für die Blinkfrequenz der LED in Abhängigkeit des gemessenen Abstandes zurückgeben. Zusammenfassend ergeben sich folgende Aufgaben:

- nutzen Sie die Programmvorlage **V2.Abstandsmessung.ino** für die Erstellung des Programms,

- erfassen Sie die ermittelte Distanz vom Ultraschallsensor **Grove Ultrasonic Ranger**,
- geben Sie den erfassten Abstand über den serielle Monitor aus,
- lassen die eine LED in Abhängigkeit der erfassten Distanz blinken (beachten Sie dabei die Bedingungen für die Blinkfrequenzen),
- legen Sie eine eigene Unterfunktion an, die Ihnen einen Delay-Wert in Abhängigkeit der gemessenen Distanz für die Blinkfrequenz zurück gibt (Übergabeparameter: Distanz, Rückgabewert: Delay),
- Übergabe des fertigen Programms an den Praktikumsleiter.

Hinweis zur Datenerfassung: Damit Sie die Daten vom Ultraschallsensor abrufen können, benötigen Sie die **Grove-Ultrasonic-Ranger-Library**, die Sie hier im Repository finden. Diese Bibliothek muss der Arduino IDE hinzugefügt werden. Eine Anleitung Dazu finden Sie im Repository „**Praktikumsunterlagen und Praktikumsinformationen**“ in der Präsentation „**Einführung in Arduino**“. Die Bibliothek enthält auch ein Beispiel, welches Ihnen die benötigte Klasse „Ultrasonic“ und deren Funktionen erläutert. Schauen Sie sich dieses Beispiel an und übernehmen Sie die benötigten Funktionen in Ihr Programm.

4. Literaturverzeichnis

[1] Atmel

ATmega328 (Mikrocontroller) Datenblatt

http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf

Abfragedatum: 23.10.2017

[2] Arduino

Language Reference

<https://www.arduino.cc/en/Reference/HomePage>

Abfragedatum: 23.10.2017

[3] Seeed

Grove - Ultrasonic Ranger

http://wiki.seeed.cc/Grove-Ultrasonic_Ranger/

Abfragedatum: 04.12.2017

[4] Nongnu

Standard C Library for AVR-GCC

<http://www.nongnu.org/avr-libc/user-manual/modules.html>

Abfragedatum: 23.10.2017

Hochschule Anhalt | Anhalt University of Applied Sciences | Fachbereich 6 EMW
Praktikum Maschinenprogrammierung EIT, MT und BMT 3. Semester

Prof. Dr. Ingo Chmielewski

 Ingo.Chmielewski@HS-Anhalt.de

Tobias Müller, B. Eng.

 Tobias.Mueller@Student.HS-Anhalt.de

