



Der Arduino und Ultraschallsensoren

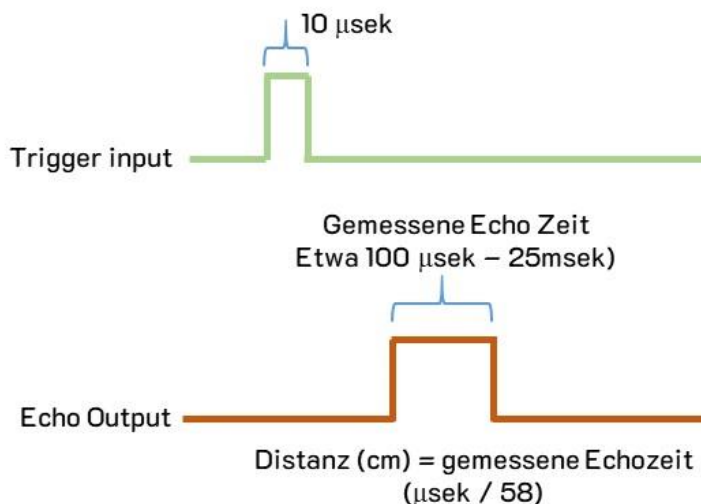
Zu den am häufigsten eingesetzten Sensoren gehören Ultraschallsensoren für die Abstandsbestimmung zu einem Hindernis. Das folgende Beispiel orientiert sich am HC SR-04.



- 1: VCC, Versorgungsspannung 5V
- 2: Triggereingang
- 3: Echo, Ausgang Messergebnis
- 4: GND, 0V

Mit dem Trigger wird eine Messung eingeleitet, auf dem Echo-Ausgang wird das Messergebnis ausgegeben. Dabei entspricht die Dauer des High-Signals der

Entfernung zum Hindernis. Siehe Timing-Diagramm unten: (Quelle: eigene Darstellung nach mbed.com)



Es ist also erforderlich, zur Messung der Echodauer eine Zeitvergleichsbasis zur Verfügung zu haben.

Bevor wir anfangen können, Programmcode zu schreiben, müssen wir erst verstehen, wie die Entstehung von Signalen und von Aktionen funktionieren (die Formeln berücksichtigt dabei bereits, dass das Signal zum Hindernis und zurücklaufen muss). Die Signaldauer entspricht in etwa der Laufzeit des

Echosignals zum Hindernis und zurück.

Die notwendigen Pins für den Sensor müssen ebenfalls wieder festgelegt. Hier:

```
//define HC-SR04 module IO Pin
#define trigPin A5
#define echoPin A4
```

der Triggerpin wird als Output, der Echopin als Input mittels pinMode definiert.

Während der Befehl delay(xxx) eine Verzögerung in Millisekunden umsetzt, erlaubt der Befehl delay-

Microseconds(xxx) eine dies analog für Mikrosekunden.

Von einem Pin lesen können wir mit dem Befehl **pulseIn(pin, wert)** lesen. Beispiel **duration = pulseIn(echoPin, HIGH);** lesen.

Auszug aus der Arduino-Referenz: pulseIn liest einen Wert von einem vorgegebenen Digitalpin ein, entweder HIGH oder LOW. Wenn wert z.B. HIGH ist, wartet pulseIn() darauf, dass der Pin auf den Wert HIGH wechselt, startet einen Timer und wartet anschließend darauf, dass der Pin wieder auf LOW wechselt. Daraufhin stoppt pulseIn() den Timer und gibt die Länge des Impulses in Mikrosekunden zurück. Wird ein Timeout erreicht, kommt 0 zurück.



Glätten und Filtern von Messdaten

Bei der Erfassung von Messwerten (wie z.B. die des Ultraschallsensors) können starke Schwankungen auftreten (z.B. durch Reflektionen). Um die Daten zuverlässig verarbeiten zu können, setzt man Filter- oder Glättungstechniken ein. Besonders bei bewegten Objekten, bei denen sich der tatsächliche Abstand zu einem Hindernis ständig ändert, ist man darauf angewiesen, möglichst sicher zu wissen, wie weit das Hindernis entfernt ist.

Dazu werden z.B. die jeweils 10 letzten Messwerte in einem Array oder einem Stack gespeichert und ihr Mittelwert gebildet. Dieser Mittelwert wird nun mit einem neuen Messwert gewichtet, indem sowohl der Mittelwert als auch der neue Messwert nur mit einem Anteil berücksichtigt werden. Also z.B. 20% der neue Wert und 80% der Mittelwert. Je kleiner der Anteil des neuen Messwerts ist, desto ruhiger ist der Verlauf der zu verarbeitenden Messung, allerdings sinkt die Reaktionsschnelligkeit. Dieser neue Messwert kommt anschließend in die Menge der alten Daten, der älteste fliegt raus. Je größer der Anteil des neuen Messwerts umso stärker wieder die Schwankungen, aber umso schneller kann reagiert werden und umgekehrt.

Glättung von Messdaten

Glättung von Messdaten hat den Zweck, zufällige Anteile in der Messreihe zu reduzieren. Die Glättung ist daher ein Spezialfall der Filterung.

Der aktuelle Datenpunkt wird mit einigen seiner Nachbarwerte verglichen und daraus ein korrigierter Datenpunkt berechnet.

Bei der Glättung ist aber zu berücksichtigen, dass zu Beginn einer Messung (auch am Ende) keine oder nicht genügend Vergleichsdaten zur Verfügung stehen, so dass keine Glättungswerte berechnet werden können.

Glättung durch Gleitenden Mittelwert

Für den gleitenden Mittelwert benötigen wir Messwerte **vor** dem aktuellen. Für jeden Messwert wird ein Mittelwert berechnet, der sich aus dem Messwert und seinen Nachbarwerten ergibt. Je mehr Messwert man zu beiden Seiten des aktuellen Messwertes nimmt, desto stärker ist die Glättung.

Offensichtlich ‚hinkt‘ das Verfahren aber der aktuellen Zeit hinterher, da erst neue Messpunkte vorhanden sein müssen, bevor ein geglätteter Wert berechnet werden kann.

Statt dem Mittelwert kann auch der Median verwendet werden.

Glättungsverfahren dieser Art sollten nur eingesetzt werden, wenn entweder viele Messungen im Verhältnis zur Bewegung aufgenommen werden, oder wenn die Messdaten nicht unmittelbar zur Aktionsbestimmung benötigt werden.

Daher kann das Verfahren so abgewandelt werden, dass es eine Menge alter Messdaten mittelt und daraus den zu erwartenden neuen Messwert bestimmt.



Alpha-Beta-Filter

Der Alpha-Beta-Filter findet sich zur Glättung von Messdaten in einer Reihe von Sensoren wieder.

Wir nehmen an, unser Fahrzeug bewegt sich auf ein Hindernis zu. Dann ändert sich der Abstand nach jeder Messung um den Wert der Geschwindigkeit und der Zeit mit

$d_k = d_{k-1} - \Delta t * v$, also verringert sich der Abstand um das Produkt aus der Fahrgeschwindigkeit v und der vergangenen Zeit Δt .

Nun wird bestimmt, wie gut der angenommene Wert (Schätzwert) zu dem der Messung passt.

$r_k = \text{mess}_k - d_k$, diese Abweichung wird als Residuum bezeichnet. Wir aktualisieren mit Hilfe dieses Residuums unsere neue Schätzung des Abstands mit $d_k = d_k + \alpha * r_k$. Der Faktor Alpha bestimmt damit, wie stark sich die Abweichung auf das Update unserer Schätzung auswirkt. Mit dem Beta-Faktor korrigieren wir nun unsere angenommene Geschwindigkeit (die wir ja nicht messen können): $v_k = v_k + \frac{\beta}{\Delta t} r_k$.

Wenn wir in unserem Fall mit einer angenommenen Geschwindigkeit von 0,1 m/s starten liefern Werte von $\alpha = 0,06$ und $\beta = 0,002$ recht gute Ergebnisse. Die beiden Faktoren hängen davon ab, wie groß die Abweichungen der Geschwindigkeit einerseits und die der Messungen andererseits sein können. Es kann gut angenommen werden, dass die Abweichungen bei der Messung (das Rauschen z.B. durch Reflektionen) deutlich größer sein dürften als die Abweichungen der Geschwindigkeit (z.B. durch Beschleunigung).