



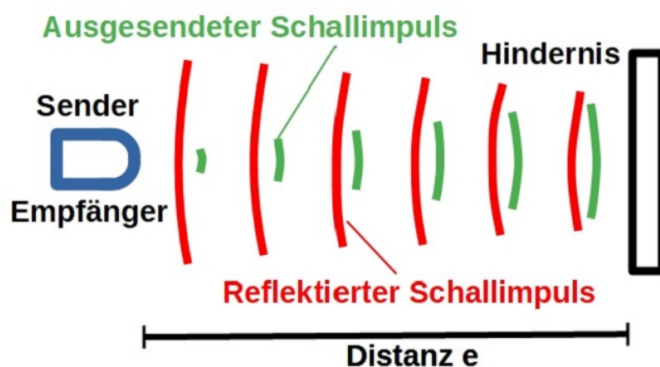
NNXT - Tutorials

[Startseite](#) [Tutorial #01](#) [Tutorial #02](#) [Tutorial #03](#) [Tutorial #04](#) [Tutorial #05](#)
[Tutorial #06](#)

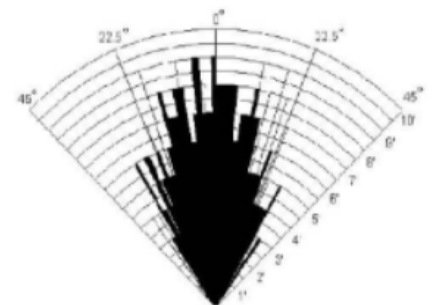
Tutorial #02 - Der Ultraschallsensor

Grundsätzliches

Der Ultraschallsensor ist ein Sensor zur Messung der Entfernung von Objekten. Er verwendet dabei ein aus dem Tierreich bekanntes Verfahren (z.B. bei Fledermäusen), das Aussenden von Schallimpulsen und das anschließende Detektieren von Reflektionen. Die folgende Abbildung macht das Prinzip deutlich:



Sie können folglich nur Objekte detektieren, die a) Schall reflektieren und nicht absorbieren oder streuen und die b) groß genug sind, eine Schallreflektion zu erzeugen. Schallreflektionen werden mit zunehmender Entfernung schwächer, so dass bei der Detektion eine Begrenzung der Reichweite zu erwarten ist. Bei den in diesem Kurs eingesetzten Sensoren liegt die bei etwa 1,20m (manchmal gehen auch höhere Werte; das ist ein statistischer Wert). Beachten Sie auch, dass eine Schallwelle sich nicht zielgerichtet geradeaus ausbreitet, sondern in Form eines Kreisausschnittes vom Sender. Dies bedeutet, dass ein detektiertes Objekt sich irgendwo auf dem Kreisbogen der Entfernung befindet. Genauere Angaben kann man mit einem Ultraschallsensor nicht machen. Nebstehende Abbildung zeigt dies anhand der Daten eines handelsüblichen Ultraschallsensors. Der Kreisausschnitt, auf dem Objekte detektiert werden können, umfasst etwa 30° zu beiden Seiten der zentralen Aussendelinie vom Sensor.



Eingesetzt werden diese Art von Sensoren übrigens bei Abstanzsensoren im Pkw. Wer diese häufig nutzt, kennt auch deren Schwächen, die etwas mit der Reflektionseigenschaften der zu detektierenden Objekte zu tun hat: manchmal genügen einfache, dünne Pflanzenzweige um eine Reflektion zu erzeugen, andere Objekte (wie beispielsweise Maschendrahtzäune mit dünnen Drähten) erzeugen keine Reflektion.

Der in der Übung eingesetzte Ultraschallsensor der Firma Lego besitzt zwei "Augen", eines ist der Sender des Ultraschall-Impulses, das andere ist der Empfänger:



Anschließen müssen Sie den Sensor an einen der Sensor-Ports Port_0 bis Port_3. Die NNXT-SW-Library bietet bei der Programmierung des Sensors Unterstützung an, die im folgenden beschrieben ist.

Die Ultraschallsensor-API

Bevor wir den Ultraschallsensor verwenden können, müssen wir den Port, an dem der Sensor angeschlossen ist, zunächst für diesen Sensor konfigurieren. Dies erfolgt einmalig am Beginn des Programmes durch folgenden Funktionsaufruf, wobei **Port_X** durch **Port_0**, **Port_1**, **Port_2** oder **Port_3** zu ersetzen ist, je nachdem an welchem Port Sie den Sensor angeschlossen haben:

```
SensorConfig(Port_X, SensorUS);
```

Anschließend kann der Sensor verwendet werden. Um Entfernungswerte vom Sensor zu bekommen, können Sie die folgende Funktion benutzen:

```
sensor_error_t US_GetDistance(sensorport_t port, uint8_t *distance);
```

Das Interface für Daten von Sensoren in der NNXT-Library ist genormt, d.h. Sie bekommen den Wert grundsätzlich immer über einen Call-by-Reference Parameter (in diesem Fall **distance**) und der Rückgabewert der Funktion ist ein Fehlercode. Der Fehlercode stellt Ihnen über einen eigenen Datentyp Fehlerursachen, die beim Aufrufen der Funktion aufgetreten sind, zur Verfügung. Sollte ein Fehlercode zurückgegeben werden, so sollten Sie den Sensorwert nicht benutzen, da im allgemeinen dieser dann undefiniert ist. Der Rest des Interfaces der Funktion dürfte selbsterklärend sein. Zur Verdeutlichung folgender Beispiel-Code einer Funktion, die den Ultraschallsensor ausliest und das Ergebnis (oder eine Fehlermeldung) auf dem Display anzeigt:

```
void detektiereObjekt() {  
    sensor_error_t err;  
    uint8_t dist;  
    char distStr[20];
```

```
err = US_GetDistance(Port_0, &dist);  
if (err != sensor_error_NoError) {  
    NNXT_LCD_DisplayStringAtLine(0, "Sensor Error!!!!  ");  
} else {  
    sprintf(distStr, "Entfernung:%d", (int)dist);  
    NNXT_LCD_DisplayStringAtLine(0, distStr);  
}  
}
```

Der Wert, den die Funktion **US_GetDistance** als Entfernung zurück gibt, ist die Entfernung eines Objektes in Zentimeter. Bekommen Sie den Wert 255 zurück, so bedeutet dies, dass kein Objekt in der Reichweite des Sensors detektiert wurde.

Die Funktion **US_GetDistance** liefert einen von drei verschiedenen Fehlercodes:

- **sensor_error_NoError** : es ist kein Fehler aufgetreten
- **sensor_error_SensorNotConfigured** : der Port ist nicht als SensorUS konfiguriert
- **sensor_error_TooFastRead** : die Zeit seit dem letzten Lesevorgang des Sensors ist zu kurz

Insbesondere der letzte Fehler ist interessant und beachtenswert: Das Auslesen des US-Sensors dauert eine gewisse Zeit und kann nicht in einer beliebigen Geschwindigkeit erfolgen. Die minimale Zeitspanne zwischen zwei Lesevorgängen (und damit zwischen 2 Aufrufen von **US_GetDistance**) beträgt 15 Millisekunden. Sie müssen also in Ihrem Code dafür sorgen, dass die Aufrufe dieser Funktion (auch und vor allem in Schleifen) mindestens 15ms zeitliche Differenz haben. Dies erledigen Sie am besten mit der **Delay**-Funktion, d.h. mit der oben angegebenen Funktion und einem Ultraschallsensor an Port 3 könnte Ihre main-Funktion folgendermaßen aussehen:

```
int main() {  
    SensorConfig(Port_3, SensorUS); // Port_3 als Ultraschallsensor  
    konfigurieren  
    while(1) {  
        Delay(20); // 20ms warten bevor der Sensor wieder  
        ausgelesen wird  
        detektiereObjekt(); // Sensor abfragen und Ergebnis anzeigen  
    }  
}
```

