



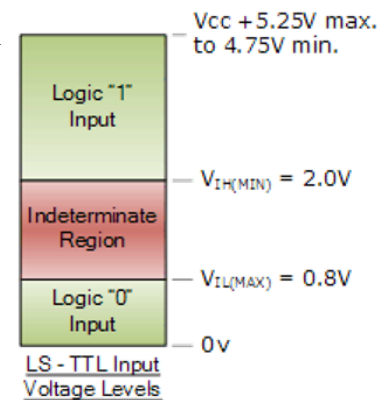
# NNXT - Tutorials

Startseite   Tutorial #01   Tutorial #02   Tutorial #03   Tutorial #04   Tutorial #05   Tutorial #06

## Tutorial #06 - Der I2C-Bus und digitale Inputs/Outputs

### Grundlagen

In der digitalen Welt der Computer arbeiten wir grundsätzlich mit den logischen Werten 1 und 0. Diese sind aber in der elektrischen Welt, die ja als Implementierung digitaler Rechner benutzt wird, nicht existent, sondern hier ist die einzig relevante Größe die elektrische Spannung  $U$ . Um von der elektrischen Spannung  $U$  in die digitale Welt zu kommen, hat man eine Interpretation unterschiedlicher Spannungswerte standardisiert, den sogenannten TTL-Pegel (TTL steht für Transistor-Transistor-Logik, weil digitale Schaltungen immer aus Transistoren aufgebaut werden). Wie diese Pegel für eine 5 Volt Versorgungsspannung aussehen, sehen Sie in nebenstehender Abbildung. Möchte man folglich aus seinem Rechner einen logischen Wert nach außen leiten, um beispielsweise ein Lämpchen an oder aus zu schalten, so muss das digitale Signal in seine entsprechende Spannung übersetzt werden. D.h. wenn man eine **logische 1** über einen Pin an die Außenwelt legt, muss auf dem Pin eine Spannung aus dem **oberen Bereich des TTL-Pegels** genommen werden, legt man hingegen eine **0** nach außen, muss eine Spannung aus dem **unteren Bereich** genommen werden. Um etwaigen Störungen vorzubeugen, wird daher immer der obere bzw. der untere Rand genommen. Heißt also: legen Sie eine 1 nach außen, dann führt der Pin 5V, legen Sie eine 0 nach außen, führt der Pin 0V.



Wie in der Vorlesung besprochen, können wir die digitalen Ausgänge der NNXT-Ports nicht so ohne weiteres verwenden, da über diese beiden Ausgänge der I2C-Bus nach außen geführt wird. Möchten wir also digitale Signale an die Außenwelt übermitteln, benötigen wir einen Baustein, den wir über I2C ansprechen können und der uns digitale Ausgänge (oder auch Eingänge) zur Verfügung stellt. Ein solcher Baustein ist der PCF8574, den wir in Übung 5 verwenden wollen. Dieser verfügt über 8 digitale Ausgänge (P0 bis P7), die wir jeweils auf 1 oder auf 0 setzen können, was dann natürlich bedeutet, dass die zu den Ausgängen gehörigen Pins des ICs jeweils 5V oder 0V zur Verfügung stellen. Genauere Informationen zu diesem Baustein entnehmen Sie bitte dem Datenblatt, welches Sie im Elearning-Kurs finden. Schauen Sie unbedingt in das Datenblatt, da dort auch das Package beschrieben ist, also an welchem Beinchen des IC Sie welches Signal finden. Das ist für den Aufbau wichtig.

Sie haben beim NNXT grundsätzlich 2 Möglichkeiten, den I2C zu benutzen: entweder Sie verwenden einen der Ports 0 bis 3, an dem sonst die Sensoren angeschlossen waren, oder Sie verwenden den Pinheader, den Sie neben dem Display unterhalb der Knöpfe auf dem NNXT finden. Für beide Varianten finden Sie unten die Beschreibung, wie Sie etwas anschließen müssen und wie die Software dafür geschrieben werden muss.

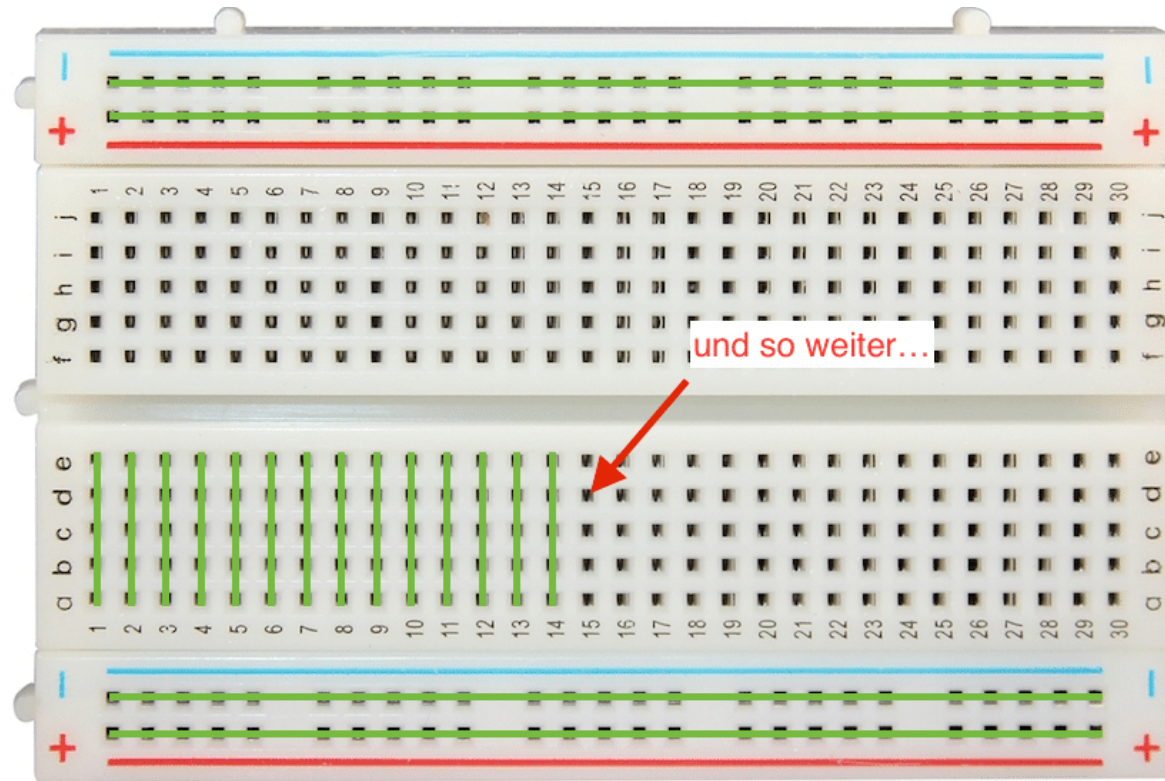
Zunächst aber zum elektronischen Aufbau: Dafür verwenden wir in der Übung ein Steckbrett und das schauen wir uns genauer an. Wer damit schon gearbeitet hat, kann den nächsten Abschnitt überspringen.

### Verwendung eines Steckbrettes

Steckbretter dienen dem schnellen prototypischen Aufbau von Elektronikschaltungen. Man kann damit sehr schön erste Schaltungsentwürfe testen, ohne jeweils eine Platine anfertigen und löten zu müssen. Sie sind sehr einfach in der Handhabung

und man kann relativ wenig falsch machen. Trotzdem sollten Sie vor dem Anschluss an den Roboter Ihre Schaltungen immer auf Kurzschlüsse überprüfen, da dies im Zweifel durchaus den Roboter zerstören kann. Also bitte sehr sorgfältig arbeiten!

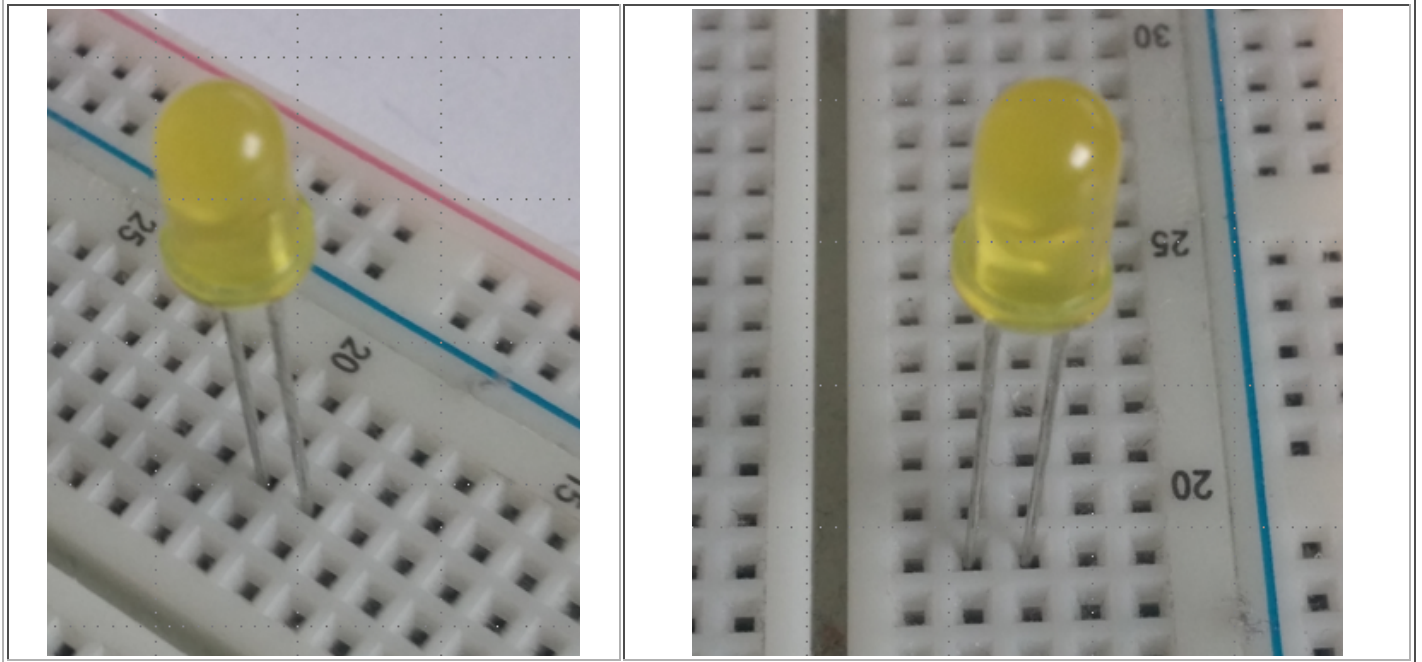
Steckbretter (siehe Abbildung unten) stellen zunächst Einsteckbuchsen zur Verfügung, in die man Kabel oder Beinchen von ICs und anderen Bauteilen einstecken kann und die eine elektrische Verbindung mit anderen Einsteckbuchsen etablieren. Die Einsteckbuchsen sind dabei in Lochreihen aufgeteilt, wobei die Buchsen einer Reihe jeweils miteinander elektrisch verbunden sind, wie in der folgenden Abbildung zu sehen ist:



Es gibt dabei 2 unterschiedliche Bereiche: die oben und unten liegenden mit einem farbigen Balken versehenen Lochreihen sind horizontal, die anderen vertikal bis zur Lücke verbunden. Die grünen Striche oben zeigen, wie die Verbindungen aufgebaut sind.

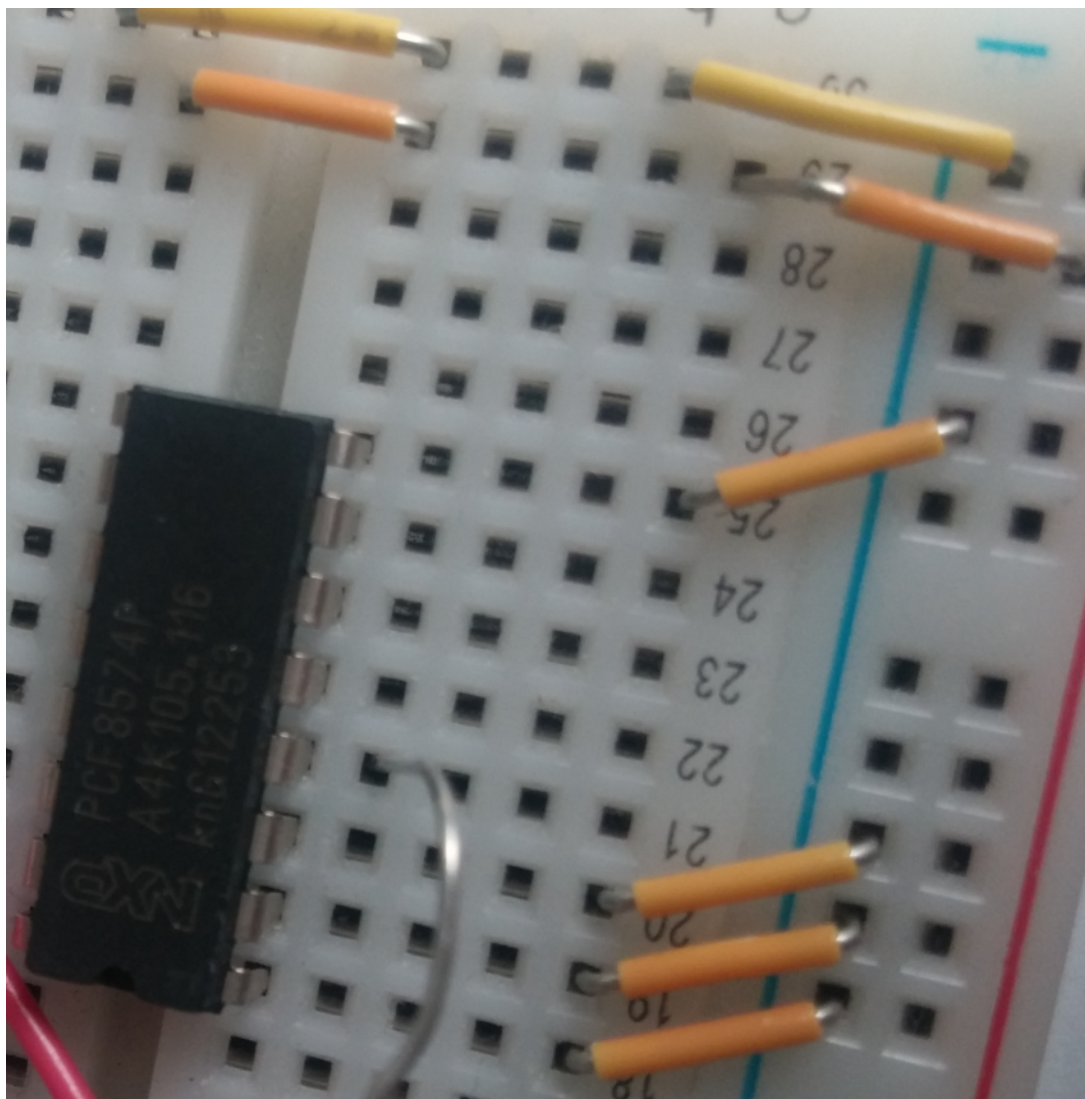
Die kurzen Reihen, die in der Abbildung vertikal verbunden sind, dienen dazu, Bauelemente zu stecken. Passen Sie dabei auf, dass Sie keine Kurzschlüsse erzeugen, also beispielsweise beide Beinchen eines zweibeinigen Bauteils in die gleiche vertikale Reihe stecken. Die folgende Abbildung macht dies deutlich:

So ist es richtig!	So ist es falsch!
Die beiden Beinchen sind in zwei unterschiedlichen Reihen und der Strom muss daher durch das Bauteil fließen	Die beiden Beinchen sind in der gleichen Reihe. Daher gibt es eine Überbrückung des Bauteils durch die Verbindung der Reihe, d.h. der Strom fließt nicht durch das Bauteil



Größere Bauteile, wie beispielsweise ICs werden über die Lücke gesteckt, wie in der unten folgenden Abbildung zu sehen ist. Die in der Reihe liegenden Buchsen sind damit mit dem in der Reihe steckenden Pin des Bauteils elektrisch verbunden und Kabel oder Bauteile können von dort gesteckt werden.

Die horizontalen Reihen dienen der Stromversorgung mit 5V und Masse (0V). Da diese sehr häufig in Schaltungen benötigt werden, sind sie über die gesamte Breite des Steckbrettes geführt. Bitte nutzen Sie diese Reihen unbedingt, da dadurch eine viel übersichtlichere Schaltung aufgebaut werden kann: weniger Kabel heißt weniger Fehlerquellen, weniger Kabel heißt weniger aufwändige Fehlersuche. Verwenden Sie möglichst kurze und passende Kabel, um den Aufbau so übersichtlich wie möglich zu gestalten, wie folgende Abbildung zeigt:



Die Spannungsversorgungsreihen auf den beiden Seiten des Steckbretts sind übrigens nicht miteinander verbunden, so dass Sie eine Verbindung mit Kabeln herstellen müssen (dies ist im obigen Bild an der oberen Kante noch ein wenig zu sehen). Dies ist in jedem Fall zu empfehlen, um eine übersichtliche Schaltung zu bekommen. Verwenden Sie dabei die mit einem roten Strich markierte Reihe immer für die Versorgungsspannung und die mit einem blauen Strich markierte Reihe immer für die Masse.

Versorgungsspannung, Masse und Signalleitungen bekommen wir aus dem NNXT geliefert, entweder über die Sensorports oder über den externen Pinheader. Wie und womit Sie diese Signale auf das Steckbrett bekommen zeigt der nächste Abschnitt.

## Anschluss eigener Sensoren

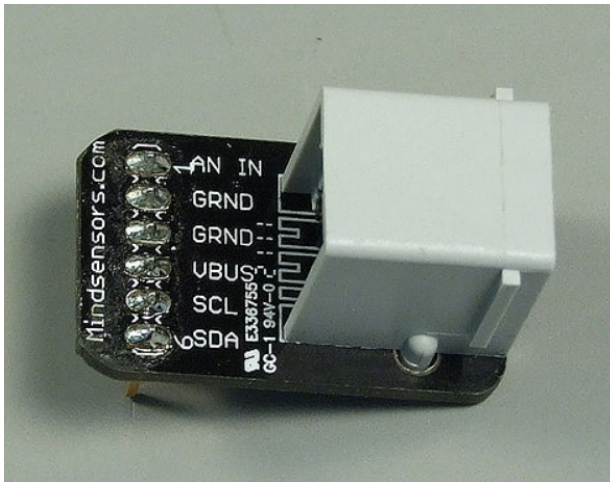
Wie oben schon beschrieben, gibt es 2 Möglichkeiten des Anschlusses: An einem Sensorport oder am Pinheader. Zunächst das Vorgehen am Sensorport. Dies ist die bevorzugte Methode, da Sie dann die langen Kabel aus der Lego-Box benutzen können und der Aufbau etwas einfacher anzustecken ist (so Sie denn nicht die DIY-Kabel benutzen müssen).

### Anschluss an Sensorport

Für den Anschluss an den Sensorport gibt es zwei mögliche Varianten: a) die benutzungsfreundliche Variante über eine spezielle Steckbuchse, die mit einem normalen schwarzen Kabel an einen Sensorport angeschlossen werden kann und b) die fummelige Variante mit einem selbstgelötetem Kabel, das leider sehr kurz ist. Zunächst Variante a):



Hierfür existiert eine Adaptersteckbuchse, die Sie direkt auf ein Steckbrett stecken können. Der Adapter führt alle 6 Leitungen des Sensorports auf seine Beinchen. Wo die Signale liegen ist jeweils beschriftet:



Die Signalbeschriftungen haben die folgende Bedeutung:

1. AN IN : Analoger Eingang; brauchen wir erst in Übung 6, kann also jetzt ignoriert werden
2. GRND : Masseverbindung (0V)
3. GRND : Masseverbindung (0V)
4. VBUS : Versorgungsspannung (5V)
5. SCL : Takt-Signal (SCL) des I2C-Busses
6. SDA : Daten-Signal (SDA) des I2C-Busses

Bitte schließen Sie **beide** Masseleitungen an die blaue Reihe des Steckbrettes an. Bitte schließen Sie die VBUS-Leitung an die rote Reihe des Steckbrettes an (jeweils wie oben beschrieben).

Da wir von diesen Adaptern nur eine begrenzte Menge haben, kann es Ihnen passieren, dass Sie stattdessen ein selbstgelötetes Anschlusskabel verwenden müssen (Variante b) von oben). Dieses verfügt leider nur über ein sehr kurzes Anschlusskabel und es ist daher ein wenig fummelig beim Aufbau. Die folgende Abbildung zeigt dieses Kabel.



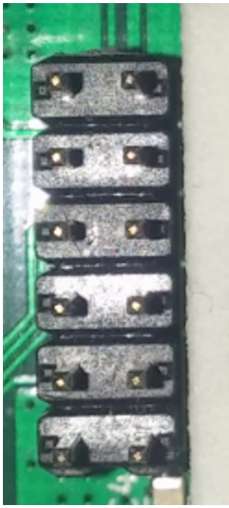
Die Anschlüsse sind in der gleichen Reihenfolge wie oben (also 1. Leitung AN IN, 2. Leitung GRND, ..., 6. Leitung SDA). Die rote Leitung kennzeichnet die 6. Leitung, also SDA. Achten Sie darauf besonders, da Sie ansonsten Kurzschlüsse erzeugen könnten, die den Roboter zerstören.

## Anschluss an Sensorport

Sollten Sie kein Kabel an einem Sensorport verwenden wollen, so können Sie auch einen Anschluss über den Pinheader aufbauen. Den Pinheader finden Sie auf dem NNXT-Baustein neben dem Display:



Auf diesem Pinheader finden Sie unterschiedliche Anschlussmöglichkeiten für Kommunikationsbusse (SPI-Bus, I2C-Bus, UART-Schnittstelle), von denen wir lediglich die I2C-Signale und die Spannungsversorgungssignale (VDD und GND) benötigen. Die Pinbelegung können Sie folgender Abbildung entnehmen (und sollte auf dem NNXT auch als Aufkleber neben dem Pinheader zu finden sein).

<b>GND</b>		<b>VDD (+5V)</b>
<b>SPI_MOSI</b>		<b>UART_TX</b>
<b>SPI_MISO</b>		<b>UART_RX</b>
<b>SPI_SCLK</b>		<b>I2C_SCL</b>
<b>SPI_CS</b>		<b>I2C_SDA</b>
<b>VDD (+5V)</b>		<b>GND</b>

Der Anschluss zum Steckbrett erfolgt über entsprechende Kabel (male/female) und ist ansonsten analog zum Vorgehen beim Anschluss des Adapters. Achtung! Die SW-Ansteuerung ist geringfügig anders, achten Sie also auf die Beschreibung der API weiter unten.

## Die I2C-API

Wie alle Sensoren, müssen die Sensorports auch für die Verwendung des I2C-Protokolles initialisiert werden. Dies erfolgt initial über einen Aufruf der Funktion **SensorPortInit**:

```
void SensorPortInit(sensorport_t port);
```

Für den Parameter **port** setzen Sie wieder den entsprechenden Sensorport ein, an dem Sie die Erweiterung angeschlossen haben (**Port\_0**, **Port\_1**, **Port\_2** oder **Port\_3**). Falls Sie den Pinheader verwendet haben, müssen Sie hier stattdessen den externen Port angeben: **Port\_extern**

Um mit dem angeschlossenen I2C-Device zu arbeiten, existieren ein paar Funktionen, die direkt auf das verwendete Bauteil PCF8574 zugeschnitten sind. Damit Sie diese verwenden können, müssen Sie die dafür notwendige Headerdatei zu Ihrer

Quelldatei hinzufügen:

```
#include "nnxt_pcf8574lb.h"
```

Dort finden Sie einige Funktionen für den Zugriff auf den PCF8574-Baustein, insbesondere die beiden Funktionen **WritePort** und **DeletePort**:

```
void WritePort(sensorport_t port, uint8_t input);
```

```
void DeletePort(sensorport_t port, uint8_t input);
```

**WritePort** dient dazu, in den im Parameter **input** angegebenen digitalen I/O Port des PCF8574 eine logische 1 zu schreiben und **DeletePort** schreibt eine logische 0 in den angegebenen digitalen I/O-Port. Die digitalen I/O-Ports des PCF8574 sind dabei von 0 bis 7 durchnummeriert (so wie auch im Datenblatt des Bausteins mit P0 bis P7 angegeben). Für weitere Funktionen (im Rahmen der Übung 5 aber nicht notwendig) schauen Sie in die Dokumentation der Funktionen.

Die I2C-Adresse des Bausteins ist bereits im oben angegebenen Headerfile festgelegt, so dass Sie diese beim Schreibzugriff nicht angeben müssen. ACHTUNG! Im Labor existieren zwei unterschiedliche Typen von PCF8574-Bausteinen, die einen mit der Endung **AN**, die anderen mit der Endung **P**, siehe folgendes Beispiel für einen Baustein mit der Endung **P**:



Diese Endung in der Bezeichnung des Bauteils ist sehr wichtig, weil diese beiden Bauteile unterschiedliche I2C-Adressen haben: PCF8574P haben die Adresse 0x40, PCF8574AN haben die Adresse 0x70. Kontrollieren Sie bitte diese Endung bei Ihrem Baustein und vergleichen Sie die eingestellte Adresse im **nnxt\_pcf8574lb.h** Headerfile in Zeile 14:

```
7  #ifndef NNXT_PCF8574LB_H_INCLUDED
8  #define NNXT_PCF8574LB_H_INCLUDED
9  /*
10 /*      includes      */
11
12 /*
13 /*      macros      */
14 #define PCF8574_Address      0x70
15 /*
16 /*      extern variables      */
17
```

Ändern Sie ggf. diesen Wert. Wenn Sie nun neu übersetzen, bitte nicht den Menüpunkt "Build", sondern den Menüpunkt "Rebuild" (rechts daneben im Embitz) verwenden, da Headerdateien üblicherweise nicht in den Abhängigkeitslisten eines Projektes stehen und Ihre Änderung andernfalls nicht in Ihr Programm übernommen wird.

Damit haben wir alles notwendige für Übung 5 kennengelernt.

## Viel Spaß beim Basteln auf dem Steckbrett!