Ein Bild, das Tisch enthält.

Automatisch generierte BeschreibungEinheit 4: Neigungssensor

Materialien

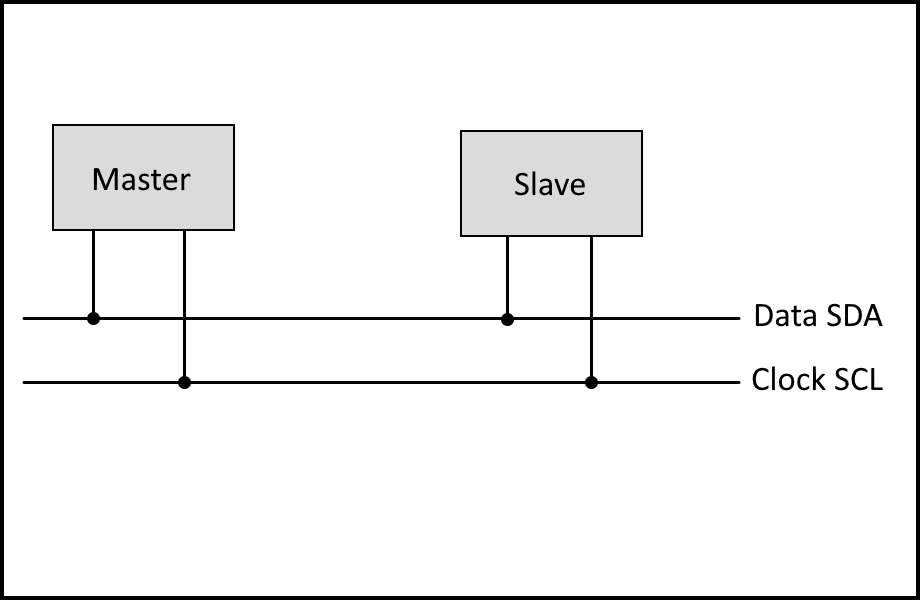
* 1x Neigungssensor (ADXL345)
* 1x Arduino Nano
* 1x Breadboards (mittelgroß)
* 2x Breadboards (klein)
* Jumper Kabel männlich/männlich

Abbildung 1 Materialien

Mit dem Neigungssensor ist es möglich die Informationen zur Lage und Beschleunigung des Sensors auszulesen. Dadurch ist es beispielsweise möglich Bewegungssteuerung in Controllern umsetzen. Der Neigungssensor wird mit dem Arduino Nano durch die I²C-Kommunikationstechnologie verbunden.

Der Neigungssensor misst drei Werte, um zu zeigen, wie er in X-, Y- und Z-Richtung ausgerichtet ist. Diese Werte können zwischen -256 und 255 liegen, abhängig davon, wie der Sensor ausgerichtet ist.

Um den Neigungssensor mit dem Arduino Nano zu verbinden, benötigen wir vier Kabel. Zuerst verbinden wir GND am Neigungssensor mit GND am Arduino, und 5V am Neigungssensor mit dem 5V-Output des Arduinos. Für die SCL- und SDA-Leitung am Arduino benötigen wir die analogen Pins A5 und A4, wobei A5 die SCL-Leitung und A4 die SDA-Leitung ist. Also verbinden wir den SCL des Neigungssensors mit A5 und SDA mit A4.

I²C (Inter integrated circuit)

I²C ist eine Art von Kommunikation zwischen elektrischen Geräten, die zwei Leitungen hat: die SDA-Leitung (Serial Data) und die SCL-Leitung (Serial Clock). Diese sind für die Datenübertragung und Synchronisation verantwortlich. Es gibt auch ein Master-Modul und mehrere Slave-Module mit eindeutigen Nummern. In unserem Fall ist der Master der Arduino Nano und der Slave der Neigungssensor. Der Master und alle Slaves müssen mit der SCL- und SDA-Leitung verbunden sein (siehe Abbildung). Da wir nur einen Slave haben, können wir ihn direkt mit dem Master verbinden und benötigen keine Zwischenleitung.

Abbildung 2 I²C Master-Slave Illustration

**Aufgabe 1**

|  |  |
| --- | --- |
| **1** | **Zeichne Abbildung 2 mit einem weiteren Slave-Modul** |

Neigungssensor anschließen

Wir nehmen uns ein mittelgroßes Breadboard und platzieren den Arduino Nano darauf. Auch den Neigungssensor platzieren wir auf zwei kleine Breadboards, mit jeweils den Pins auf einem der kleinen Breadboards. Jetzt brauchen wir vier männlich/männlich Kabel für die Verbindung. Zuerst verbinden wir GND vom Arduino mit GND vom Neigungssensor und 5V mit 5V. Der SDA-Anschluss beim Arduino ist der Analoge Port 4, also verbinden wir A4 mit SDA, um Daten zu übertragen. Schließlich müssen wir den Analogen Port 5, welcher der SCL-Anschluss beim Arduino ist, mit SCL vom Neigungssensor verbinden, um die Kommunikation zu synchronisieren.

**Aufgabe 2**

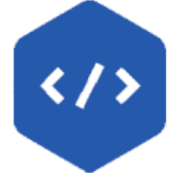
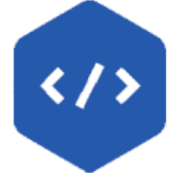
|  |  |
| --- | --- |
| **2** | **Baue die beschriebene Schaltung nach und lasse sie vom Lehrer überprüfen** |

**Achtung überprüfen lassen**

Verbindet eure Schaltung noch nicht mit einer Energiequelle, bevor euer Lehrer nicht eure Schaltung überprüft hat. Bei falscher Schaltung kann es zur Beschädigung von Bauteilen kommen.

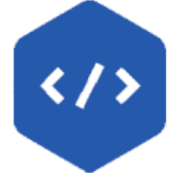
Neigungssensor anlegen

Jetzt wo wir den Neigungssensor richtig verkabelt haben, müssen wir noch die Kommunikation einrichten und die Daten auslesen. Zuerst einmal brauchen wir die Bibliotheken ADXL345.h und Wire.h. Die Wire.h Bibliothek kann direkt eingebunden werden, die ADXL345.h muss vorher aus dem Bibliothekmanager installiert werden. Einfach in dem Bibliothekmanager nach ADXL345 suchen, da sollten einige verschiedene Bibliotheken erscheinen. Wir wählen die Bibliothek „Accelerometer ADXL345“ von Seeed Studio. Zu Beginn müssen wir uns eine Variable des Typen ADXL345 anlegen.



|  |  |
| --- | --- |
| **#include <Wire.h>** |  |
| **Beschreibung** | Fügt dem Sketch die Wire Bibliothek hinzu, welche benötigt wird, um Daten per I²C zu empfangen und zu senden. |

|  |  |
| --- | --- |
| **#include <ADXL345.h>** |  |
| **Beschreibung** | Fügt dem Sketch die ADXL345 Bibliothek hinzu, um den Neigungssensor benutzen zu können. |

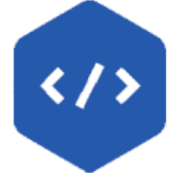


|  |  |  |
| --- | --- | --- |
| **ADXL345 bezeichner;** | |  |
| **Beschreibung** | Legt eine Variable vom Typen ADXL345 an, mit der wir arbeiten können. Der Bezeichner kann frei gewählt werden sollte aber geeignet gewählt werden | |

**Aufgabe 3**

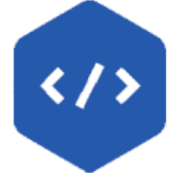
|  |  |
| --- | --- |
| **3** | **Code schreiben**   * **Binde die Bibliothek Wire.h ein** * **Binde die Bibliothek ADXL345.h ein** * **Erstelle eine Variable des Typen ADXL345 mit geeigneter Bezeichnung** |

Sensordaten auslesen

Jetzt müssen wir noch die Daten aus dem Neigungssensor auslesen. Dafür müssen wir zunächst den Sensor starten mit powerOn.

|  |  |
| --- | --- |
| **adxl.powerOn();** |  |
| **Beschreibung** | Startet euren Neigungssensor. Die Funktion müsst ihr über die Variable des Typen ADXL345 aufrufen. |
| **Parameter** | void |
| **Rückgabewert** | void |

Zum Auslesen der tatsächlichen Daten aus dem Neigungssensor gibt es den Befehl adxl.readXYZ(), der direkt alle drei Neigungen ausliest.

******Aufgabe 4**

|  |  |
| --- | --- |
| **adxl.readXYZ(&x, &y, &z);** |  |
| **Beschreibung** | Mit diesem Befehl kann man die Neigung aller drei Achsen des Sensors auslesen und direkt in angegebene Variablen abspeichern. |
| **Parameter** | &x: Die Variable, in die der Neigungswert der x-Achse gespeichert werden soll (das &-Zeichen muss bleiben), der Datentyp ist int  &y: Die Variable, in die der Neigungswert der y-Achse gespeichert werden soll  &z: Die Variable, in die der Neigungswert der z-Achse gespeichert werden soll (das &-Zeichen muss bleiben) |
| **Rückgabewert** | void |

|  |  |
| --- | --- |
| **4a** | **Code schreiben**   * **Starte den Neigungssensor in der Setup Funktion** * **Speichere die Neigungs Werte in Variablen ab** * **Gib die Werte schön formatiert mit einem kleinen Delay auf dem seriellen Monitor wieder** |
| **4b** | **Teste aus, wie du den Neigungssensor Bewegen musst, um einen positiven und negativen Wert auf der X-Achse zu bekommen, ohne dass die anderen Werte sich verändern.** |

Bei Fragen bezüglich der ADXL345 Bibliothek oder bei Interesse an dem Source Code der Bibliothek einfach auf der GitHub Seite vorbeischauen: <https://github.com/Seeed-Studio/Accelerometer_ADXL345>