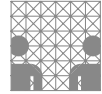


64-211 Übung Eingebettete Systeme



Aufgabenblatt 3 Termine: 26.04./29.04 + 03.05./05.05.

Gruppe	
Name(n)	Matrikelnummer(n)

Häufig sind die in eingebetteten Systemen verarbeiteten Signale analoger, kontinuierlicher Natur. Besonders häufig ist dieses dann der Fall, wenn es sich um Signale externer Komponenten (z. B. Sensoren) handelt. Um analoge Signale innerhalb eines digitalen Systems verarbeiten zu können ist eine Analog-Digital-Wandlung (de.wikipedia.org/wiki/Analog-Digital-Umsetzer) durch Abtastung/Quantisierung notwendig. Ihre Aufgabe ist es einen analog "kommunizierenden" Sensor (**Gyroskop**) in Ihr System einzugliedern und die akquirierten Daten sinnvoll aufzubereiten, um diese dann in den anschließenden Aufgaben zu verwenden.

Aufgabe 3.1

Entwerfen Sie ein Programm zur Auswertung der Daten eines 2-Achsen (**X, Z**) Gyroskops. Machen Sie sich zunächst mit der Funktionsweise eines Gyroskops vertraut (de.wikipedia.org/wiki/Drehratensensor / de.wikipedia.org/wiki/Gyroskop). Verwenden Sie für Ihren Versuchsaufbau den Sensorbaustein **IXZ-500** von **Invensense**.

Informieren Sie sich über die Anschlussbelegung (Abb. 1) und betrachten Sie das entsprechende Datenblatt: tams.informatik.uni-hamburg.de/lectures/2016ss/vorlesung/es/uebungen/doc/ixz-500.pdf). Verdrahten Sie den Mikrocontroller und den IXZ-500 Sensorbaustein derart, dass das von Ihnen entworfene Programm folgendes leistet:

1. Tasten Sie die Ausgangssignale des Gyroskops für die X- und die Z-Achse mit 10 Hz ab.
2. Bestimmen Sie die Abweichung vom Referenzsignal.
3. Errechnen Sie, der von Ihnen gewählten Sensitivität entsprechend, den korrekten Wert für die Drehrate der jeweiligen Achse (die Formel befindet sich im Datenblatt auf der Seite 13).
4. Geben Sie die einzelnen Werte (sinnvoll formatiert) über den seriellen Monitor aus.
 - Fahren Sie mit 1. fort.

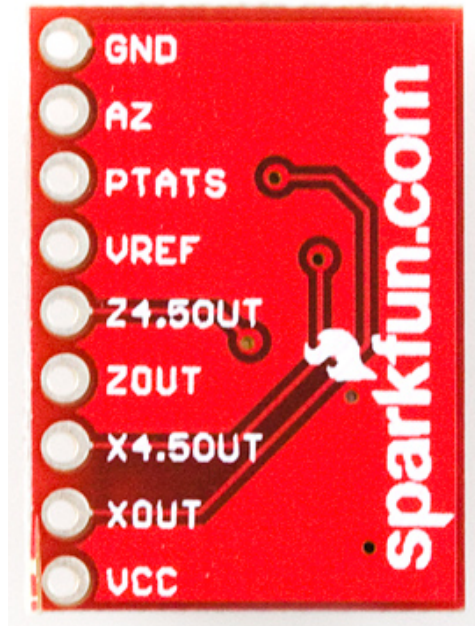


Abbildung 1: Anschlussbelegung des IXZ-500 Breakout-Boards.

Achtung: Die Ausgänge des IXZ-500 Sensorbausteins sind **analog!** Benutzen Sie daher die analogen Eingänge des Mikrocontrollers (**beginnend mit dem Buchstaben "A"**) für den korrekten Anschluss des Sensors. Verwenden Sie folgende Funktion zur Abtastung der analogen Eingänge:

```
* analogRead(<pin>) arduino.cc/en/Reference/AnalogRead
```

Die restlichen für die Bearbeitung der Aufgabe notwendigen Funktionen sollten Ihnen bereits bekannt sein. Beachten Sie bei der Verdrahtung Ihres Versuchsaufbaus folgende Hinweise:

- Stellen Sie zunächst die Stromversorgung des IXZ-500 Sensorbausteins sicher. VCC und GND sind die entsprechenden Anschlüsse. Der Baustein benötigt 5 V Versorgungsspannung.
- Sie haben zwei Möglichkeiten die Ausgangssignale für die beiden Achsen abzugreifen. Die Ausgänge X0UT und Z0UT ermöglichen das Messen einer höheren Rotationsgeschwindigkeit, die Ausgänge X4.50UT und Z4.50UT liefern eine höhere Präzision. Die notwendigen Berechnungsfaktoren finden Sie im Datenblatt des IXZ-500 (Seite 13).
- Verbinden Sie den Eingang AZ (IXZ-500) mit einem beliebigen digitalen Anschluss-Pin des Arduino Due. Konfigurieren Sie den Eingang anhand der Information aus dem Datenblatt (Seite 14).
- Das Referenzsignal erhalten Sie am Ausgang VREF des IXZ-500 Bausteins.

Aufgabe 3.2

Ein im Bereich der Robotik häufig eingesetzter Aktor ist der **Servo-Motor**. Entwerfen Sie im Rahmen dieser Aufgabe ein Programm, welches die Steuerung eines Servo-Motors ermöglicht. Verwenden Sie hierzu den zur Verfügung gestellten **Datan S1213** Servo mit analogem Feedback-Signal. Zur Ansteuerung wird ein pulsweitenmoduliertes Signal benötigt - verdrahten Sie den Mikrocontroller und den Servo-Motor entsprechend. Verwenden Sie für die Lösung der Aufgabe die Servo Funktionsbibliothek des Arduino-Projekts. Machen Sie sich mit dem Funktionsumfang dieser Bibliothek vertraut: arduino.cc/en/reference/servo.

Achtung: Die gelbe Anschlussleitung darf **nur** für den Anschluss eines pulsweitenmodulierten Signals verwendet werden, der Servo-Motor könnte sonst beschädigt werden! Zusätzlich gilt es zu beachten, dass der Funktionsumfang der Servo Bibliothek explizit über `#include <Servo.h>` in Ihrem Quellcode eingebunden werden muss.

Entwerfen Sie ein Programm, um den Arbeitsbereich des verwendeten Servo-Motors abfahren zu können. Verwenden Sie das analoge Feedback-Signal (weisse Leitung) um festzustellen, ob bestimmte Winkel an den Grenzen des Arbeitsbereichs nicht mehr angefahren werden können. Implementieren Sie dazu folgendes Verhalten:

1. Beginnen Sie mit der Winkelstellung in der Mitte des Arbeitsbereichs des Servos (90 Grad) und schwingen Sie den Servo-Motor bis an die Grenzen des von der Servo Bibliothek verwendeten Arbeitsbereiches auf. Bewegen Sie den Servo-Motor dabei möglichst flüssig.
2. Dämpfen Sie anschließend die Oszillation langsam ein, bis Sie erneut in der mittleren Winkelstellung angekommen sind.
 - Fahren Sie mit 1. fort.

Aufgabe 3.3

Verwenden Sie die Lösung der Aufgabe 3.1 und erweitern Sie diese um die Möglichkeit der direkten Steuerung des Servo-Motors mittels der Rotation des Gyroskops um eine von Ihnen gewählte Achse. Beachten Sie dabei die Grenzen des Servo-Motors. Signalisieren Sie das Erreichen einer Grenze durch dreimaliges Blinken der auf dem Arduino Due Board integrierten LED. Die integrierte LED kann über Anschluss 13 angesteuert werden.

Aufgabe 3.4

Die serielle Schnittstelle kam bisher primär als Kanal für die Ausgabe von "Debugging"-Information zum Einsatz. Entwerfen Sie im Rahmen dieser Aufgabe ein Programm, welches es Ihnen ermöglicht die auf dem Board integrierte LED mittels Kommandos über ein serielles Terminal, ein- und auszuschalten. Es steht Ihnen frei den seriellen Monitor der Arduino IDE als Terminal zu verwenden.

Betrachten Sie erneut die Dokumentation der Serial Funktionsbibliothek: arduino.cc/en/Reference/serial. Folgende Funktionen sind hilfreich bei der Lösung der Aufgabenstellung:

* <code>Serial.available()</code>	arduino.cc/en/Serial/Available
* <code>Serial.read()</code>	arduino.cc/en/Serial/Read

Entwerfen Sie zunächst eine Version, die einzelne Zeichen als Steuerkommandos interpretiert. Erweitern Sie die Lösung, indem Sie die Verarbeitung von Strings als Steuerkommandos implementieren, z.B. `LED_on` und `LED_off`.

Aufgabe 3.5

Abschließend sollen die Inhalte der Aufgaben 3.2 und 3.4 zusammengefasst werden, indem Sie die Steuerung des Servo-Motors durch Kontrollkommandos implementieren. Implementieren Sie einen "Parser", der Befehle der Art `moveServoTo(angle)` verarbeiten kann. Vergessen Sie nicht die Benutzereingaben zu validieren - kontrollieren Sie z. B. ob es sich um Werte ausserhalb des Arbeitsbereichs handelt.