

## Aufgabenblatt 2 Termine: 08.04. / 11.04.

Gruppe	
Name(n)	Matrikelnummer(n)

### Aufgabe 2.1 (Punkte 20)

Entwerfen Sie ein Programm zur Steuerung eines Gleichstrommotors. Machen Sie sich mit der Funktionsweise einer H-Brücke vertraut: [en.wikipedia.org/wiki/H\\_bridge](http://en.wikipedia.org/wiki/H_bridge). Der für die Lösung der Aufgabenstellung zu verwendende Baustein TB6612FNG enthält zwei solcher Brücken.

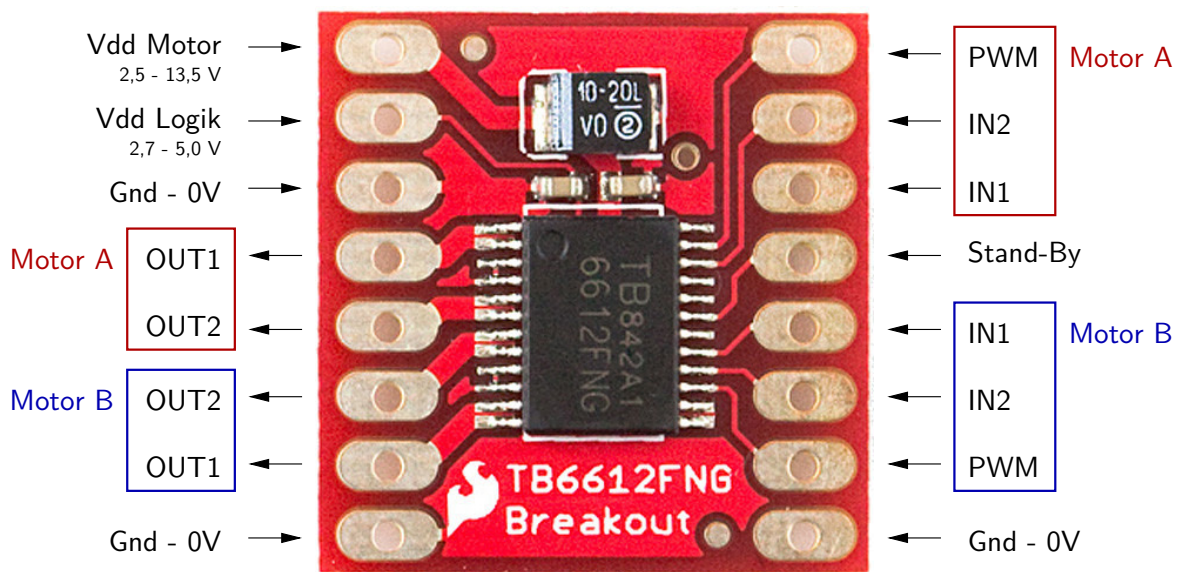


Abbildung 1: Anschlussbelegung des TB6612FNG Breakout-Boards.

Informieren Sie sich über die Anschluss-Belegung (Abb. 1 und Datenblatt: [tams.informatik.uni-hamburg.de/lectures/2014ss/vorlesung/es/doc/tb6612fng.pdf](http://tams.informatik.uni-hamburg.de/lectures/2014ss/vorlesung/es/doc/tb6612fng.pdf)) und verdrahten Sie den ARM-Microcontroller, den TB6612FNG Baustein und den Gleichstrommotor derart, dass das von Ihnen entworfene Programm den Motor nach folgendem Schema ansteuert:

1. Rotieren Sie den Motor im Uhrzeigersinn (CW). Fahren Sie dabei die Leistung innerhalb eines Zeitraums von 5 Sekunden gleichmäßig von 0% auf 100%.
2. Stoppen Sie den Motor indem Sie die Leistung schnell, jedoch gleichmäßig reduzieren.
3. Ändern Sie die Rotationsrichtung und rotieren Sie den Motor entgegen dem Uhrzeigersinn (CCW). Gehen Sie bei der Leistungsregelung analog zu 1. und 2. vor.
  - Fahren Sie mit 1. fort.

Die für die Bearbeitung der Aufgabe notwendigen Funktionen sollten Ihnen bereits vom vorherigen Aufgabenblatt bekannt sein. Beachten Sie bei der Verdrahtung (siehe dazu Abb. 1) folgende Hinweise:

- Stellen Sie zunächst die Stromversorgung des des TB6612FNG Bausteins sicher. „Vdd Motor“ ist der Anschluss für die Versorgungsspannung des Motors (nutzen Sie hier bitte die vom Arduino Due bereitgestellten 5V), „Vdd Logik“ ist der davon getrennte Spannungsanschluss der Logik (hier sind 3.3V ausreichend). Alle Gnd-Anschlüsse sind intern verbunden.
- Da der TB6612FNG Baustein zwei H-Brücken enthält, haben Sie die Wahl zwischen zwei Kanälen: Kanal A (rot gekennzeichnet) und Kanal B (in blau markiert). Benutzen sie alle fünf Anschlüsse (s.u.) eines der beiden Kanäle.
- Verbinden Sie den Kontrollanschluss „PWM“ mit einem PWM-fähigen Pin des ARM-Microcontrollers. Die Eingänge „IN1“ und „IN2“ ermöglichen die Wahl zwischen den im Datenblatt beschriebenen Steuer-Zuständen. Die Anbindung des Gleichstrommotors muss über die Anschlüsse „OUT1“ und „OUT2“ erfolgen.
- Vergessen Sie nicht Stand-By mit einem definierten Pegel zu versehen:  
Stand-By = HIGH schaltet den Baustein ein  
          = LOW schaltet diesen aus.

### Aufgabe 2.2 (Punkte 20)

Erweitern Sie die Lösung der ersten Aufgabe indem Sie zwei Taster hinzufügen um folgende Funktionsweise zu implementieren:

- Ein Taster schaltet (zyklisch) zwischen folgenden Zuständen des Systems um:
  1. Rotation im Uhrzeigersinn
  2. Rotation entgegen dem Uhrzeigersinn
  3. Gleichmäßiger Motorstopp.
- Der zweite Taster soll die Leistung wie folgt regeln:
  1. Steigerung der Leistung von 0% bis 100%
  2. Absenkung der Leistung von 100% bis 0%

Es steht Ihnen frei die Lösung entweder als Schleifen-Konstrukt (Polling) zu implementieren oder Interrupts zu verwenden.

**Aufgabe 2.3** (Punkte 20)

In Anlehnung an den ersten Aufgabenteil sollen Sie ein Programm entwerfen, welches die Steuerung eines Servo-Motors ermöglicht. Verwenden Sie den Futaba BLS451 als Servo-Motor. Zur Ansteuerung wird ein PWM-Signal benötigt - verdrahten Sie den ARM-Microcontroller und den Servo-Motor entsprechend. Verwenden Sie für in der Lösung der Aufgabe die Servo Funktionsbibliothek. Machen Sie sich mit dem Funktionsumfang dieser Bibliothek vertraut: [arduino.cc/en/reference/servo](https://arduino.cc/en/reference/servo).

**Achtung:** Die weiße Anschlussleitung darf nur für den Anschluss eines PWM-Signals verwendet werden, der Servo-Motor könnte sonst beschädigt werden!

Bestimmen Sie den Arbeitsbereich des verwendeten Servo-Motors. Implementieren Sie dazu folgendes Verhalten:

1. Beginnen Sie mit der Winkelstellung in der Mitte des Arbeitsbereichs des Servos und schwingen Sie den Servo-Motor bis an die Grenzen des Arbeitsbereiches auf. Bewegen Sie den Servo-Motor dabei möglichst flüssig.
2. Anschließend dämpfen Sie die Oszillation, bis Sie erneut in der mittleren Winkelstellung angekommen sind.
  - Fahren Sie mit 1. fort.

**Aufgabe 2.4** (Punkte 20)

Entwerfen Sie ein Programm, welches es Ihnen ermöglicht die auf dem Board integrierte LED mittels eines, über ein serielles Terminal abgesetzten Kommandos, ein- und auszuschalten. Es steht Ihnen frei den seriellen Monitor der Arduino IDE als Terminal zu verwenden. Konsultieren Sie dazu die Dokumentation der Serial API: [arduino.cc/en/Reference/serial](https://arduino.cc/en/Reference/serial).

Folgende Funktionen sind hilfreich bei der Lösung der Aufgabenstellung:

* <code>Serial.available()</code>	<a href="https://arduino.cc/en/Serial/Available">arduino.cc/en/Serial/Available</a>
* <code>Serial.read()</code>	<a href="https://arduino.cc/en/Serial/Read">arduino.cc/en/Serial/Read</a>

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/off".

**Aufgabe 2.5** (Punkte 20)

Abschließend sollen die Inhalte der letzten beiden Aufgaben zusammengefasst werden, indem Sie die Steuerung des Servo-Motors durch Kontrollkommandos, die in dem seriellen Terminal eingegeben werden, implementieren. Schreiben Sie einen entsprechenden Parser, der Befehle der Art `moveTo(angle)` verarbeiten kann.