

Aufgabenblatt 5 Termine: 12.06. / 13.06. + 19.06. / 20.06.

Gruppe	
Name(n)	Matrikelnummer(n)

Beginnend mit dieser Aufgabe werden die neuen **Arduino Due** Microcontroller-Boards eingeführt (arduino.cc/en/Main/ArduinoBoardDue). Zumindest für alle nachfolgenden Aufgabenblätter wird ausschließlich die neue Hardware benutzt. Abgesehen von einer vollkommen anderen Microcontroller-Architektur (8-bit AVR vs. 32-bit ARM) ist folgendes Merkmal besonders wichtig: **die maximale, an den I/O-Eingängen anliegende, Spannung darf unter keinen Umständen 3,3 V überschreiten!** Eine höhere Spannung führt in der Regel zur Beschädigung des Microcontrollers.

- Machen Sie sich bitte mit der Pinbelegung der neuen Prototypenplatine vertraut, bevor Sie mit der Lösung der Aufgaben beginnen: tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es/doc/duPinout.pdf.
- Zur Programmierung muss eine andere Software(-version) benutzt werden, die mit „\$tamsSW/arDue/arduino“ gestartet wird, siehe auch tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es/uebung.

Aufgabe 5.1 (Punkte 25)

Für die Lösung dieser Aufgabe werden zwei Microcontroller-Boards benötigt. Arbeiten Sie deshalb gemeinsam mit einer anderen Gruppe zusammen oder benutzen Sie zwei Teensy 2.0 Boards. Wegen der unterschiedlichen Spannungspegel (s.o.), sollten nur "gleiche"Platinen miteinander kommunizieren.

Entwerfen Sie ein Programm zum Austausch von Nachrichten zwischen den beiden Microcontroller-Boards unter Verwendung der seriellen Schnittstelle. Verbinden Sie die beiden Systeme, indem Sie (jeweils gekreuzt) die TX und RX Anschluss-Pins sowie GND verdrahten. Binden Sie an eines der Boards einen Taster an und schliessen Sie an das andere Board eine LED an. Entwerfen Sie zunächst ein einfaches Protokoll, das Ihnen folgenden Funktionsumfang bereitstellt:

1. Betätigung des Tasters regelt mittels PWM die Intensität der LED hoch (bis zum Wert 255) und daraufhin runter (bis zum Wert 0).
2. Ist eine der beiden Grenzen erreicht worden, soll dieses dem Kommunikationspartner mitgeteilt werden, indem die auf dem Board integrierte LED 3 Mal zum Blinken gebracht wird.

Hier noch zwei Hinweise zur Durchführung der Aufgabe:

Arduino Due Das Arduino Due Board bietet insgesamt 4 serielle Anschlüsse an. Es steht Ihnen frei einen auszuwählen. Bei passender Wahl können Sie Punkt 2. auch gerne so lösen, dass eine Nachricht an den angeschlossenen PC geschickt wird anstatt die integrierte LED blinken zu lassen.

Teensy 2.0 Um die UART an den seriellen Port zu binden, muss in Code folgende Anweisung enthalten sein:

```
HardwareSerial Uart = HardwareSerial();

void setup() {
  ...
}
```

Aufgabe 5.2 (Punkte 25)

Bei dieser Aufgabe benötigen Sie mindestens zwei (gleiche!) Microcontroller-Boards. Verwenden Sie für die Lösung dieser Aufgabe den seriellen I²C (Inter-Integrated Circuit) Bus zur Anbindung der Boards. Informieren Sie sich über die Funktionsweise von I²C:

- Wikipedia: en.wikipedia.org/wiki/I2C, bzw. de.wikipedia.org/wiki/I2C
- oder tiefergehend unter: www.i2c-bus.org/de/i2c-bus

Das allgemeine Anschlusschema ist in Abb. 1 skizziert. Lokalisieren Sie die Anschluss-Pins für die I²C Signale SCL und SDA auf dem jeweiligen Board und stellen Sie eine Verbindung durch direkte Verdrahtung her. Achten Sie bitte darauf die Kabellängen möglichst gering zu halten, da sonst die Verwendung von Pull-Up Widerständen für jede Signalleitung notwendig wird. Vergessen Sie bitte darüber hinaus nicht die GND Potentiale beider Boards zu verbinden.

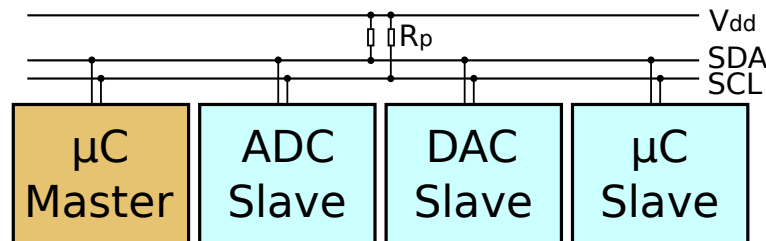


Abbildung 1: Allgemeines I²C Anschlusschema

Verwenden Sie zur Ansteuerung der I²C Schnittstellen die Wire Funktionsbibliothek, die Bestandteil der Arduino Umgebung ist: `#include <Wire.h>`. Machen Sie sich mit dem Funktionsumfang dieser Bibliothek vertraut (www.arduino.cc/en/Reference/Wire). Schauen Sie sich insbesondere die folgenden Funktionen an:

* <code>Wire.requestFrom()</code>	arduino.cc/en/Reference/WireRequestFrom
* <code>Wire.beginTransmission()</code>	arduino.cc/en/Reference/WireBeginTransmission
* <code>Wire.endTransmission()</code>	arduino.cc/en/Reference/WireEndTransmission
* <code>Wire.onReceive()</code>	arduino.cc/en/Reference/WireOnReceive
* <code>Wire.onRequest()</code>	arduino.cc/en/Reference/WireOnRequest

Machen Sie eines der Boards zum *Bus-Master* und entwerfen Sie für die Boards jeweils ein Programm, das ein Protokoll implementiert, welches folgenden Funktionsumfang anbietet:

1. Der Bus-Master schaltet die integrierte LED des Slave-Boards im 2-Sekunden-Takt ein und aus.
2. Das Resultat jeder Ein- bzw. Ausschaltaktion wird vom Bus-Master beim Slave angefragt und das Ergebnis auf der seriellen Schnittstelle (serieller Monitor) ausgegeben.

Aufgabe 5.3 (Punkte 25)

Basierend auf der vorherigen Aufgabe, gilt es eine neue Lösung für die Aufgabenstellung 3.2 zu implementieren, bei der das Slave-Board mit dem Gyroskop-Modul verbunden wird und der Bus-Master mit dem anzusteuernenden Servo.

Aufgabe 5.4 (Punkte 25)

Bei dieser Aufgabe sollen Sie die drahtlose Kommunikation kennenlernen. Verwenden Sie ein mit einem XBee ZB Modul (digi.com/xbee/) ausgestattetes Arduino Wireless SD Shield (arduino.cc/en/Main/ArduinoWirelessShield). Bitte achten Sie darauf, das Shield mit dem Arduino Due der Pin-Bezeichnung entsprechend zu verbinden.

Die von Ihnen verwendeten XBee Module sind so vorkonfiguriert, dass sie sich als Router in ein Mesh-Netzwerk integrieren. Der für die Errichtung des Mesh-Netzwerks notwendige Koordinator wird von uns betrieben.

Sie haben die Möglichkeit die Kommunikation mit einem anderen XBee Modul aufzunehmen, sofern Ihnen die Adresse dieses Moduls bekannt ist. Diese ist 64 Bit lang und setzt sich aus einem höheren und niederen Teil, von jeweils 32 Bit, zusammen.

Sie haben die Möglichkeit mit Ihrem XBee Modul seriell zu kommunizieren indem Sie wie gewohnt die erste serielle Schnittstelle (Serial) verwenden. Da diese Schnittstelle sowohl für den Datentransfer ins Netzwerk als auch für die Konfiguration des Moduls verwendet wird, gilt folgendes zu beachten:

1. Um in den Konfigurationsmodus zu gelangen ist folgende Zeichenkette an das Modul zu senden: „+++“. Beachten Sie bitte, dass diesem Befehl kein „newline“-Zeichen folgen darf.
2. Die Konfiguration des Moduls erfolgt mittels sogenannter AT Kommandos. Durch die Kombination der Zeichenkette AT und des entsprechenden Bezeichners für das gewünschte Register (z.B. ATSH) lässt sich der Wert eines Registers auslesen. Ist das Register schreibbar, so wird der **hexadezimal** notierte Wert dem AT Kommando hinzugefügt.
3. Um die Adresse Ihres eigenen XBee Moduls auszulesen sind die Register SH und SL von Bedeutung. Das erste enthält die höheren 32 Bit der Adresse, das zweite die niederen 32 Bit. Um das Kommando abzusenden fügen Sie bitte ein „newline“-Zeichen an.
4. Um den Kommunikationspartner ansprechen zu können, muss die Zieladresse über die Register DH und DL gesetzt werden. In beiden Fällen ist ein „newline“-Zeichen nach dem Kommando notwendig.

5. Haben Sie die Zieladresse erfolgreich gesetzt, so müssen sie den Konfigurationsmodus verlassen. Senden Sie dazu das Kommando ATCN gefolgt von einem „newline“-Zeichen.
6. Haben Sie den Konfigurationsmodus verlassen, so haben Sie nun die Möglichkeit mit dem Kommunikationspartner wie gewohnt seriell Daten auszutauschen.

Um Kommunikations- und Konfigurationsproblemen vorzubeugen wird empfohlen folgende Timing-Angaben einzuhalten:

- Direkt nach dem Einschalten muss 1,5 Sekunden lang gewartet werden, bis sich das Modul initialisiert hat.
- Nach Absetzen des Sonderkommandos „+++“ sind ebenfalls 1,5 Sekunden abzuwarten, bevor mit der Konfiguration fortgefahren wird.
- Nach jedem normalen AT Kommando wird empfohlen ca. 0,5 Sekunden abzuwarten, bevor das nächste Kommando geschickt wird.

Finden Sie für die Lösung der Aufgabenstellung einen Kommunikationspartner und modifizieren Sie die Aufgabenstellung 5.3 soweit, dass die Steuerung des Servos nun drahtlos erfolgen kann.

Aufgabe 5.5 (Punkte Bonus)

Versuchen Sie auszutesten bei welcher Reichweite eine zuverlässige Steuerung noch gewährleistet werden kann.