

# 7

## Übung zur Kommunikation über elektrische Schnittstellen

- 1 Ziele [7|1](#)
- 2 Allgemeine Hinweise [7|2](#)
- 3 B15 Beschaltung und Programmierung [7|3](#)
  - 3.1 Programmierung von PortA [7|3](#)
  - 3.2 Schaltung [7|4](#)
- 4 Aufgabenstellung [7|6](#)

Die Laboranleitung Nr. **7** entstand im November 2020, dem zweiten Semester unter den Auswirkungen der weltweiten COVID19-Pandemie<sup>[6]</sup>. Beim aktuellen Dokument handelt es sich um die überarbeitete Version vom **14. Oktober 2024**. Wir danken TRISTAN KRAUSE für seine wundervolle Firmware für das B15-Board.<sup>1</sup> Die Software entstand im letzten Semester seines Masterstudiums an der HTWK-Leipzig. Das B15 ist ein hochwertiges Datenerfassungsboard (DAQ), das speziell für den Einsatz an Hochschulen entworfen wurde. Es unterscheidet sich von anderen Boards<sup>2</sup> zur Datenerfassung durch seine einfache Wartbarkeit, leicht zu erlernende Programmierung und natürlich seinen Preis. Weiterhin ist die komplette Dokumentation für unsere Studentinnen und Studenten offen. Der Name leitet sich daraus ab, dass die Idee zu diesem Board 2015 entstand. Seit 1991 sind solche Boards im Hardwarepraktikum in Benutzung.

Der vorliegende Text ist durch das Urheberrecht<sup>[1]</sup> privilegiert. In Semester- und Abschlussarbeiten dürfen Sie nicht wie in Unterrichtsmitteln verfahren.



Wegen der im Dokument übernommenen Abbildungen dürfen Sie, auf Grund des Urheberrechts, diese Laboranleitung nur privat oder im Rahmen des Unterrichts nutzen und in keiner Weise weiterverbreiten.

### 1 Ziele

Wichtigstes Ziel ist das Erfolgserlebnis selbst ein Programm geschrieben zu haben, dass Daten über ein Patchkabel transportiert hat. Dabei sollen Sie selbst OSI Layer 1 und in Ansätzen Layer 2 und Layer 3 definieren und implementieren<sup>[3, 5]</sup>, also erste Kenntnisse für den Transport von Daten über elektrische Leitungen erwerben.

Außerdem werden Sie in diesem Versuch Ihre Kenntnisse im Umgang mit elektrischer Messtechnik vertiefen und die Fernwartung von Firmware über Internet üben. Diese Fähigkeiten werden zunehmend im Automobilbau oder auch Gebäudeautomatisierung gefragt. Wir geben hier einen Einstieg.

<sup>1</sup>Siehe <https://github.com/devfix/b15f>

<sup>2</sup>Vgl.: <https://www.ni.com/de-de/shop/data-acquisition.html>

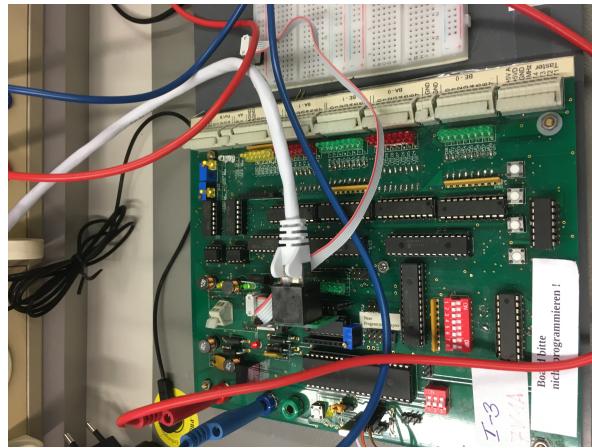


Abbildung 1: B15 Board mit Patch-Kabel an PortA

## 2 Allgemeine Hinweise

Unser Ziel war es, die Versuche für das Selbststudium und die Arbeit mit elektronischen Medien zu optimieren, die PDF sind **interaktiv** und wir erwarten die Abgabe ebenfalls in elektronischer Form. Diese dürfen Sie gern auf einem Tablett erstellen oder handschriftlich auf Papier schreiben und danach einscannen.

Führen Sie ein Labortagebuch [2, 4]. Ein Journal zu führen ist in vielen Ingenieur- und Naturwissenschaften Pflicht, z.B. um die Ergebnisse zertifizieren zu können. Die ToDos dieser Aufgabenstellen gehören teilweise zu einem vorbereitenden Theorieteil, teilweise zum Praxisteil oder zur Auswertung eines Experimentes.

Alle Rechner im Hardwarelabor haben eine statische IP-Adresse. Für externe Experimente halten wir das gesamte Subnetz offen. Das gibt den Studentinnen und Studenten alle Möglichkeiten, stellt aber gleichzeitig auch ein hohes Risiko dar. Wir können aber die Tür nicht gleichzeitig abschließen und offen lassen. Das Subnetz liegt hinter einer einfachen Firewall. Von der Hochschule aus kann aber ohne Einschränkung auf das Netz zugegriffen werden. Für einen Zugriff von zu Hause müssen Sie das VPN der Hochschule nutzen und von dort auf die **adas**.<sup>3</sup> Dem Hardwarelabor gehört das gesamte Subnetz 141.57.6.x. An der Nummerierung können Sie leicht erkennen, dass die HTWK dieses Labor sehr früh ans Internet angeschlossen hat. Heute wäre diese Exklusivität nicht mehr möglich. Gehen Sie verantwortungsvoll damit um und erhalten Sie diese unbürokratische Möglichkeit für die Generationen nach Ihnen!

Die PCs für studentische Experimente haben die IP-Adressen:

**ada1** 141.57.6.151  
**ada2** 141.57.6.152  
**ada3** 141.57.6.153  
 ...

---

<sup>3</sup>Ja, Ihr Professor ist ein Fan, auch wenn an vielen Stellen zu viel Mystik in diese Figur gelegt wird: [https://de.wikipedia.org/wiki/Ada\\_Lovelace](https://de.wikipedia.org/wiki/Ada_Lovelace) (sic!), es gibt aber auch Literatur, die den Geist erkennt: <http://sydneypadua.com/2dgoggles/lovelace-the-origin-2/> (sic!)

**ada11 141.57.6.161**

Es gibt noch weitere Rechner (Router Einplatinenrechner, virtuelle Maschinen, Steuerrechner) für studentische Experimente, die Sie aber für den aktuellen Versuch nicht benötigen.

Die Rechner {ada1, ada2}, {ada3, ada4}, {ada5, ada6} {ada7, ada8} und {ada11, ada12} bilden jeweils ein Paar. Die Datenerfassungsboards sind wie im Versuch gefordert miteinander verkabelt. Am Besten greifen Sie über ein Terminalfenster auf die Rechner zu. Unter Linux oder MacOS ist leicht mit dem Befehl `ssh famulus@141.57.6.a` möglich (ergänzen Sie bitte die IP-Adresse). Das funktioniert auch unter MS-Windows. Sollte Ihr Betriebssystem keine *secure shell* unterstützen, könnte Ihnen das Programm *putty* weiterhelfen.<sup>4</sup>

Wir haben keine automatische Verwaltung der exklusiven Ressource des Datenerfassungsboards. D.h. Sie müssen sich absprechen. Seien Sie fair! Wenn zwei Gruppen sich in je zwei Rechner teilen, müssen Sie sich absprechen, ihre Software sollte nach einer gewissen Zeit terminieren und die serielle Schnittstelle wieder freigeben. Die Ressource `/dev/ttyUSBx` ist atomar und kann nur von einem Prozess gleichzeitig genutzt werden. Daher kann immer nur eine Gruppe ihre Software auf beiden Rechnern starten. Auch wenn Sie hier vermutlich mit anderen Gruppen interagieren werden erwarten wir pro Zweiergruppe eine eigenständige Lösung!

### 3 B15 Beschaltung und Programmierung

#### 3.1 Programmierung von PortA

Das hier gezeigte Beispiel ist nur ein erster Schritt. Es hat natürlich mit der gezeigten Lösung wenig zu tun. Die Rechner sind mit dem PortA des Datenerfassungsboards verbunden. Wir zeigen Ihnen nur wie Sie auf einem Board etwas ausgeben und auf dem zugehörigen Board des anderen Rechners Daten lesen können.

Das folgende Listing überträgt abwechselnd 0 oder 1:

```
#include <iostream>
#include <b15f/b15f.h>

using namespace std;

int main ()
{
    B15F& drv = B15F::getInstance(); //drv wird ein Objekt einer Klasse
    drv.setRegister(&DDRA, 0x01); //Bit 7-1 Eingabe, Bit 0 Ausgabe

    while (1)
    {
        // Ausgabe
        cout << "Schreibe 0" << endl;
        drv.setRegister(&PORTA, 0); // ungeschicktes Alternieren
        cout << "Schreibe 1" << endl;
        drv.setRegister(&PORTA, 1);
    }
}
```

Das folgende Listing liest asynchron vom PortA und gibt die Zahl auf der Console aus:

<sup>4</sup><https://www.putty.org/>

```
#include <iostream>
#include <b15f/b15f.h>

using namespace std;

int main ()
{
    B15F& drv = B15F::getInstance(); //drv wird ein Objekt einer Klasse
    drv.setRegister(&DDRA, 0x00); //Bit 7-1 Eingabe, Bit 0 Ausgabe

    while (1)
    {
        // Lesen
        while (1){
            cout << ((int)drv.getRegister (&PINB)) << endl;
            drv.delay_ms(10);
        }
    }
}
```

Sie erkennen sicherlich die Ähnlichkeit zum Fach Systemnahe Programmierung, auch das B15 Board wird von einem Atmel Mikrocontroller gesteuert. Übersetzen Sie die beiden Listings z.B. mit dem Befehl g++ -std=c++17 -O3 ihrfile.cpp -lb15fdrv und starten Sie die Binarys auf jeweils einem der verbundenen Rechnern. Sie können hier auch wieder die Makefile aus V1 benutzen. Da es sich bei dieser Aufgabe um ein komplexes Programmierprojekt handelt, empfehlen wir dringend die Nutzung eines Versionsverwaltungssystems (z.b. über <https://git.imn.htwk-leipzig.de>).

### 3.2 Schaltung

Abbildung 2 Zeigt eine Seite der Kommunikationsstrecke. Über die Adapterplatine kann an das Patchkabel zusätzlich eine Logicanalyser angeschlossen werden um die Zustände auf der Leitung zu Dokumentieren. Wir werden die Adapterplatine nutzen um während Ihrer Datenübertragung Störungen auf der Leitung zu Provozieren.



Die Logicanalyser vertragen keine negativen Spannungen, auch Spannungen größer als 5V zerstören die Geräte, achten Sie also unbedingt das Sie am Signalgenerator einen CMOS Pegel erzeugen und verbinden Sie die Masseleitungen aller Komponenten.

Abbildung 3 zeigt alle benötigten Komponenten in einem Blockschaltbild. Bitte beachten Sie das das Patchkabel lediglich 8 Leitungen für Daten enthält, verbinden Sie zusätzlich die GND Leitung des Arduino und des B15 um ein gemeinsames Bezugspotential zu schaffen.



Bitte beachten Sie das sowohl das Arduino und das B15 serielle Verbindungen zur Kommunikation mit dem PC nutzen. Das B15 muss zwingend als /dev/ttyUSB0 verbunden werden. Entfernen Sie ggf. die USB-Kabel und stecken Sie diese in der richtigen Reihenfolge wieder ein.

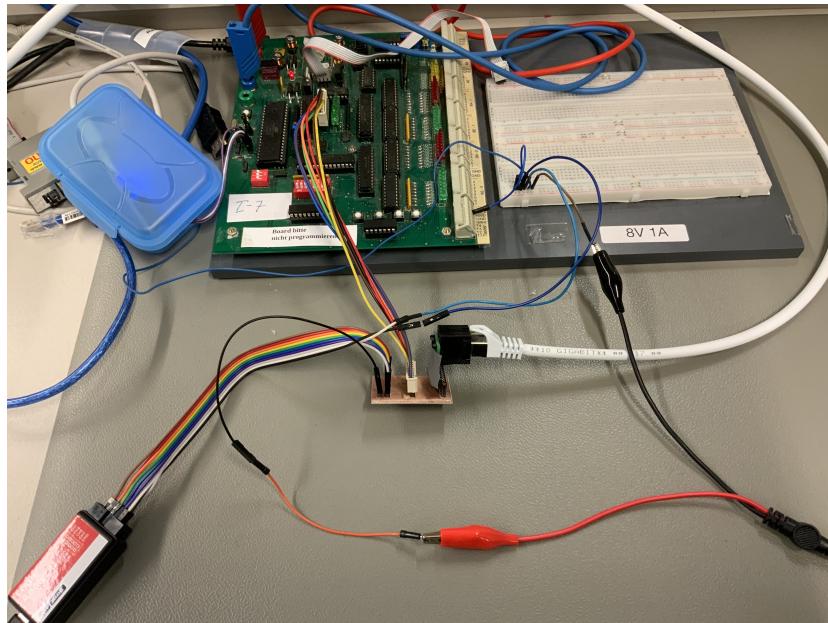


Abbildung 2: B15 Board mit Patch-Kabel an PortA, Adapterplatine, Logicanalyser und gemeinsamer Masse.



Genau wie eine echte Netzwerkkarte hat auch das B15 und der Arduino in seiner Firmware Beschränkungen die die Übertragungsgeschwindigkeit limitieren. Überfordern Sie beide Geräte nicht, mit unkontrollierten Burst-Zugriffen. Das B15 neigt hier zum Absturz, implementieren Sie lieber eine stabile und langsame Lösung als eine unzuverlässige. Geschwindigkeit spielt für die Bewertung der Aufgabe keine Rolle. (Die 1kByte Datei sollte allerdings in angemessener Zeit übertragen werden.)

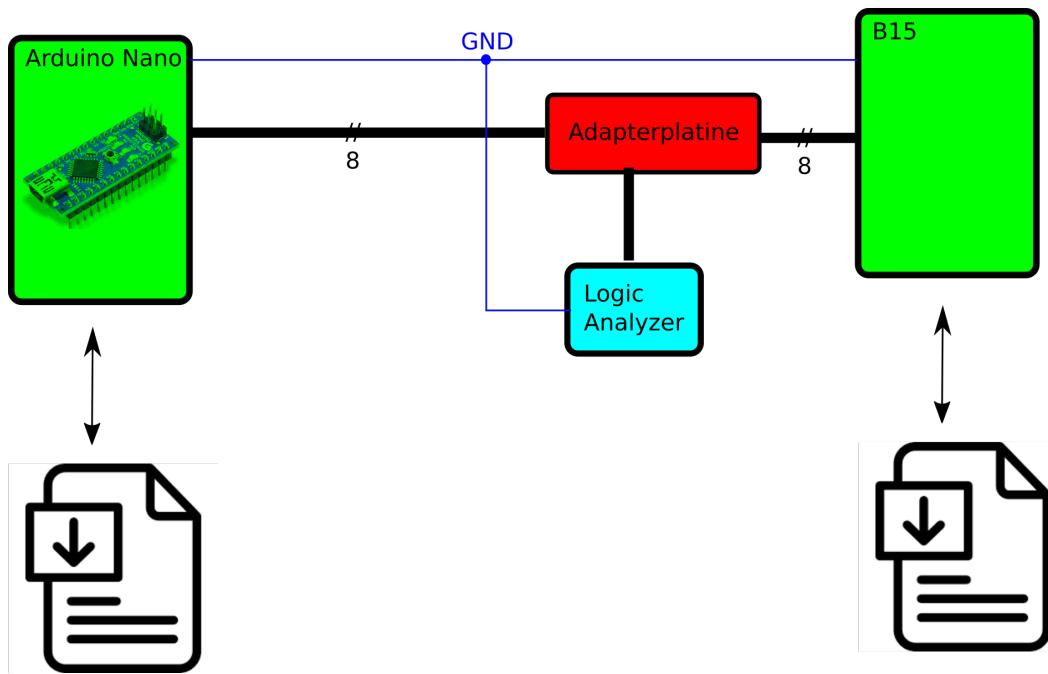


Abbildung 3: Blockschaltbild zweier Tische und aller benötigten Komponenten.

#### 4 Aufgabenstellung

Zusammengefasst müsste die Aufgabe Lauten "Implementieren Sie mit einem B15, einem Arduino und einem Patchkabel eine fullduplex Datenübertragung inklusive Automatic Repeat reQuest (ARQ) und testen Sie diese.". Zur Arbeitserleichterung haben wir für Sie Teilaufgaben formuliert. Die Aufgaben innerhalb der Blöcke sind für die Bewertung wichtig, außerhalb der roten Boxen finden Sie nützliche Hinweise.

Sie müssen keine umfassende Benutzerschnittstelle Programmieren, eine Funktionalität ähnlich wie netcat deckt alle Usecases ab.

Für die folgenden Aufgaben soll ein Full-Duplex Verfahren implementiert werden. Jeder der beiden Rechner soll zeitgleich senden und empfangen können. Sie können die Leitung z.B. aufteilen um vier Bit senden und vier Bit empfangen zu können. Wählen Sie eine oder mehrere dedizierte Takteleitungen. Ein Pegelwechsel auf dieser Leitung soll ein neues Symbol auf der Leitung signalisieren.

Schreiben Sie ein Programm, dass Zahlen empfängt und sendet. Da sie zunächst nur 3 Bit zur Verfügung haben, können Sie nur die Zahlen 0 bis 7 übertragen. Fassen Sie nun drei dieser Zeichen zusammen um ein Byte zu übertragen. Da Sie damit insgesamt 9 Bit zur Verfügung haben, reservieren Sie 8 Bit für die Nutzdaten und das 9. Bit zur Rahmensynchronisation zu Beginn eines neuen Zeichens.

**Nº 4.1**

10 Punkte

Schreiben Sie Software für den Linux-PC, und den Arduino so dass, fullduplex Zeichenketten (8bit) übertragen werden können! (Das Programm erwartet gleichzeitig Zeichen zum Senden von STD:IN und zum Empfang über Kabel.) Schreiben Sie Ihr Programm so, dass Sie eine ganze Textzeile eingeben und übertragen können. Nutzen Sie hier Linux-Pipes, STD:IN und STD:OUT. echo "das ist ein Test" | ihrbinary. Lösungen ohne Linuxpipes oder mit dem seriellen Monitor des Arduino werden nicht akzeptiert, informieren Sie sich z.B. über das Programm netcat+cat/echo, dieses ist der geforderten Lösung sehr nahe. Weisen Sie mit dem Logicanalyzer nach, dass die Übertragung wirklich fullduplex stattfindet.

**Nº 4.2**

10 Punkte

Bauen Sie eine Fehlererkennung ein.

**Nº 4.3**

10 Punkte

Ermöglichen Sie die Übertragung großer Dateien ( 1kByte bis 1GByte) im Vollduplex. Lesen Sie hier von STD:IN und geben Sie gleichzeitig auf STD:OUT aus, sodass Sie Ihr Programm über Linux-Pipes ansprechen können. Teilen Sie die Übertragung in Blöcke auf.



Unter /mnt/datadisk/\*random.bin finden Sie eine Testdateien für Ihre Übertragung.



Sollten Sie in Ihrem Programm mit Threads arbeiten, achten Sie darauf nicht gleichzeitig auf die B15-Funktionen zuzugreifen, ansonsten kann es zu Programmbstürzen kommen. Nutzen Sie Semaphoren etc.

**Nº 4.4**

10 Punkte

Erkennen Sie Fehler und übertragen Sie fehlerhafte Blöcke erneut.

**Nº 4.5**

6 Punkte

Vergleichen Sie die Geschwindigkeit Ihrer Lösung mit dem Unix-Kommandos `scp`, `netcat` und `rsync`.



Sie können für die Messung der Übertragungsgeschwindigkeit die Software `time` nutzen. Um nicht die Zeit für Ihre Passworteingabe zu messen müssen Sie ggf. zunächst ein SSH-Schlüsselpaar erstellen und mit `ssh-copy-id` auf den anderen PC übertragen, hierdurch vermeiden sie die Passwortabfrage bei SSH.

Gesamtpunktzahl: 46

### Abkürzungen

**ARQ** Automatic Repeat reQuest