

Vermittlung systemnaher Programmierung während Corona

Einsatz von Entwicklungsboards bei präsenzloser Lehre

Sommersemester 2020

Bernhard Heinloth, Benedict Herzog, Tim Rheinfels und Volkmar Sieh

Lehrstuhl für Informatik 4 – Verteilte Systeme und Betriebssysteme
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

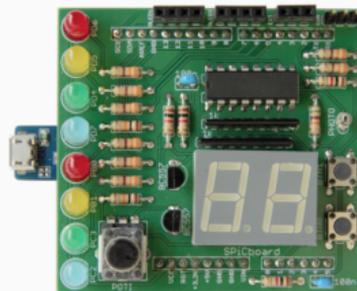
Systemnahe Programmierung in C

Das Fach **Systemnahe Programmierung in C (SPiC)** vermittelt insbesondere Studierenden der Fächer **Mechatronik** und **Elektrotechnik-Elektronik-Informationstechnik (EEI)** als Grundlagenveranstaltung Kenntnisse in der **Mikrocontrollerprogrammierung** sowie **systemnaher Softwareentwicklung unter Linux**.

Um eine praxisnahe Ausbildung zu ermöglichen, wird im Übungsbetrieb die populäre **AVR ATmega** Familie detaillierter betrachtet und ein speziell für die Veranstaltung entworfenes und mit einem **ATmega328PB** Mikrocontroller versehenes Entwicklungsboard verwendet: Das **SPiCboard** (inzwischen in der 3. Revision)



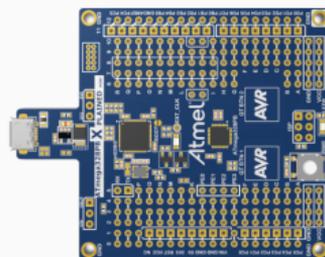
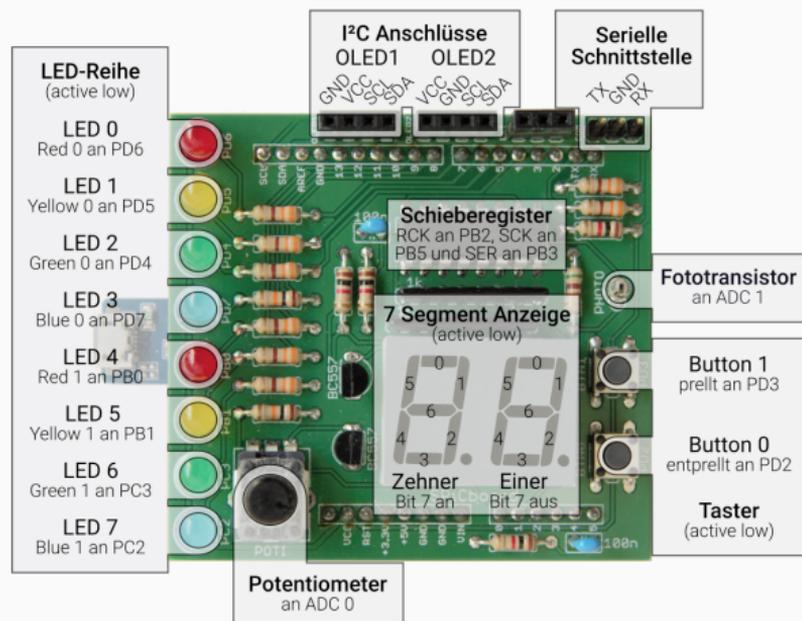
ATmega328PB



SPiCboard (v3)

Entwicklungsboard

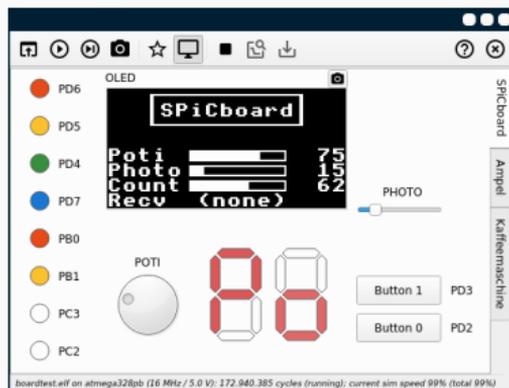
Das **SPiCboard** ist mit verschiedener Peripherie bestückt (acht LEDs, zwei Taster, ein Drehpotentiometer, ein Photodetektor sowie eine zweistelligen 7-Segment-Anzeige), welche im **Übungsbetrieb** Anwendung findet.



Die Verwendung des **Xplained Mini** Evaluierungskit (unterhalb des *Shields*) erlaubt dabei direktes Programmieren und Debuggen des SPiCboards **via USB** – ohne weitere teure Zusatzhardware.

Entwicklungsboard (simuliert)

Sofern, wie zu Beginn der Coronasituation, kein Hardware-SPiCboard zur Verfügung gestellt werden kann, steht ein Simulator für das Board – der **SPiCsim** – bereit.

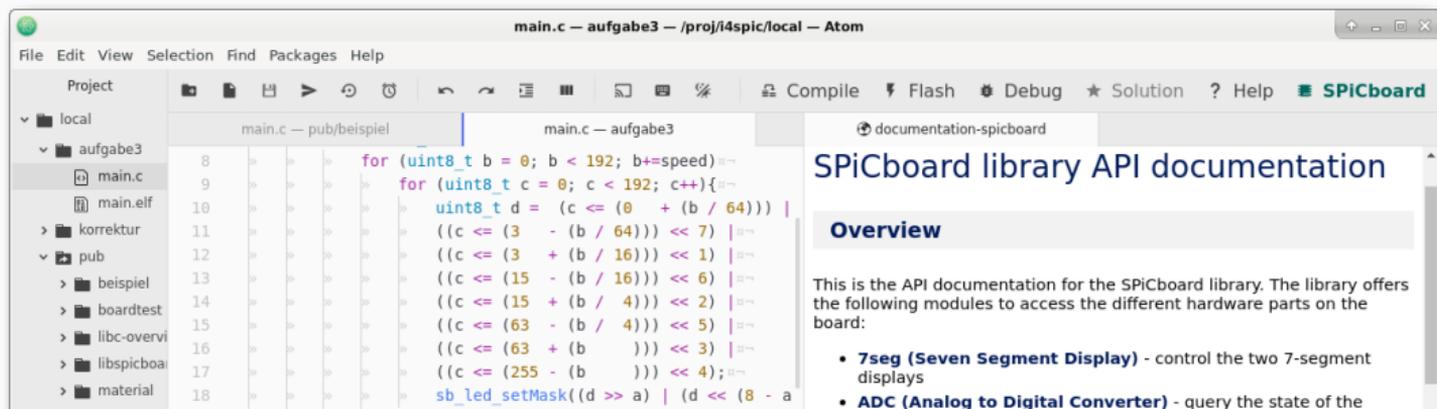


Dieser ist nicht nur in der Lage, die Peripherie **physikalisch korrekt** (bis hin zur scheinbaren Helligkeit der LEDs) wiederzugeben, sondern unterstützt auch die im SPiCboard vorhandenen **Erweiterungen** (wie OLED Display) und kann verschiedene, für die Aufgaben **angepasste Oberflächen** anzeigen. Zudem sind ein **Debugger** sowie ein **Logikanalysator** integriert.

Entwicklungsumgebung

Um eine fokussierte Bearbeitung der Übungsaufgaben zu ermöglichen, wird eine eigene **integrierte Entwicklungsumgebung (IDE)** mit einer reduzierten aber funktionalen Oberfläche eingesetzt. Diese ermöglicht direkten Zugriff auf alle relevanten Funktionen, bietet einen Überblick über das eigene Aufgabenverzeichnis und erlaubt sowohl das Übersetzen, Ausführen und (graphische) Debuggen der eigenen Anwendung auf einem **SPiCboard** als auch im **SPiCsim**.

Aufgabenstellung, Folien zur Vorlesung und Übung sowie Dokumentation der Bibliotheken können dabei direkt in der IDE aufgerufen und angezeigt werden, ebenso ist bequem die Abgabe einer Übungsaufgabe zur Korrektur möglich.



The screenshot displays the SPiCboard IDE interface. The main window shows a C code editor with the following code:

```
main.c — pub/beispiel | main.c — aufgabe3
8 | for (uint8_t b = 0; b < 192; b+=speed){
9 |     for (uint8_t c = 0; c < 192; c++){
10 |         uint8_t d = (c <= (0 + (b / 64))) |
11 |         ((c <= (3 - (b / 64))) << 7) |
12 |         ((c <= (3 + (b / 16))) << 1) |
13 |         ((c <= (15 - (b / 16))) << 6) |
14 |         ((c <= (15 + (b / 4))) << 2) |
15 |         ((c <= (63 - (b / 4))) << 5) |
16 |         ((c <= (63 + (b / 4))) << 3) |
17 |         ((c <= (255 - (b / 4))) << 4) |
18 |         sb_led_setMask((d >> a) | (d << (8 - a
```

The right-hand pane displays the **SPiCboard library API documentation** with an **Overview** section. The text states: "This is the API documentation for the SPiCboard library. The library offers the following modules to access the different hardware parts on the board:"

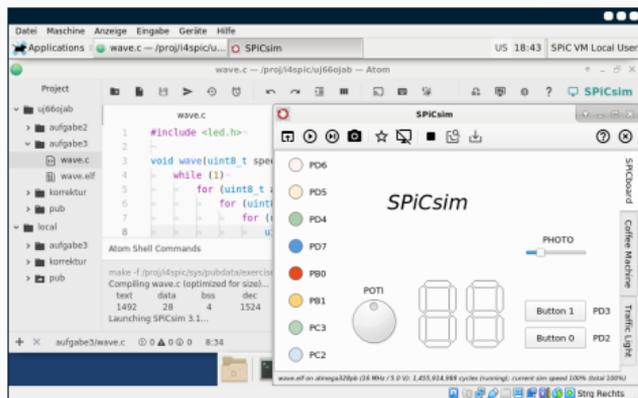
- **7seg (Seven Segment Display)** - control the two 7-segment displays
- **ADC (Analog to Digital Converter)** - query the state of the

Entwicklungsumgebung zu Hause verwenden

Für den Einsatz der **IDE zu Hause** werden neben der Oberfläche auch die beteiligten Komponenten wie **Übersetzer und Simulator** benötigt, welche den Studierenden alle gemeinsam als ein **Gesamtpaket** zur Verfügung gestellt werden.

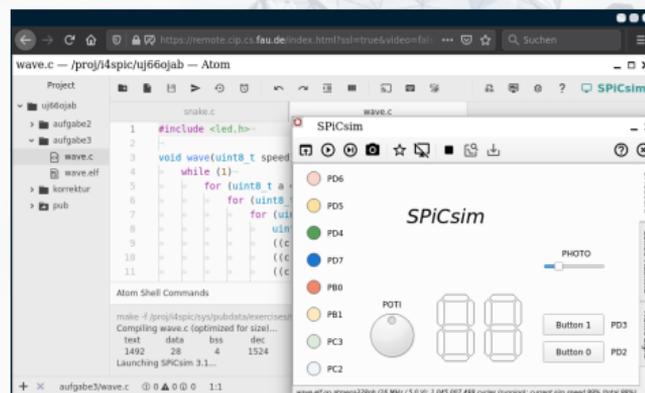
Dabei sind sowohl lokale Ausführung auf dem eigenen PC als auch Fernzugriff möglich:

Virtuelle Maschine



Abbild mit Linux und vorkonfigurierter Entwicklungsumgebung zum Download, lauffähig unter Windows, MacOS und Linux

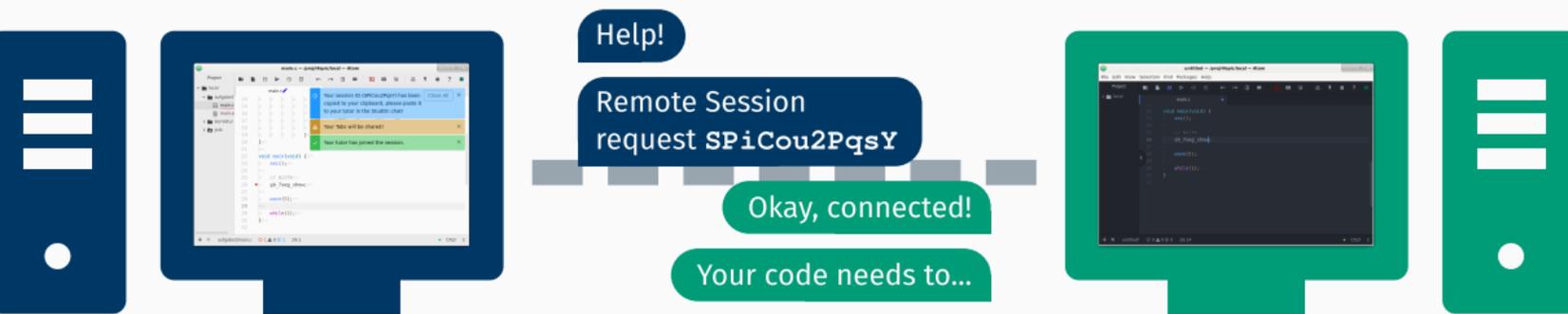
Webzugriff



Unterstützt von alle modernen Webbrowsern (Chrome, Firefox, Safari sowie Edge) durch Verwendung von HTML5 und JavaScript

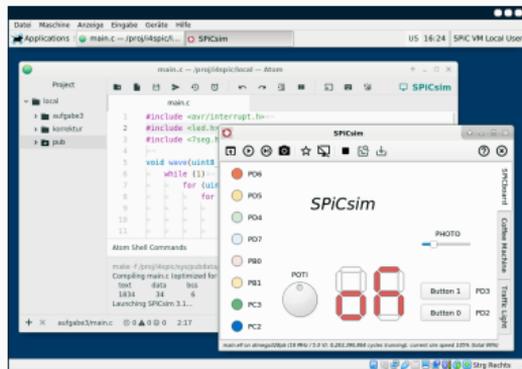
Für die Lehre wird ein **asynchrones Modell** bevorzugt: Die Inhalte werden als Webvideo abrufbare **Bildschirmaufzeichnungen** vermittelt, die **Präsentationsfolien** sind mit weiteren Informationen über die **Veranstaltungsw Webseite** abrufbar, welche auch eine **ausführliche Dokumentation** der beteiligten Komponenten beinhaltet.

Für Rückfragen stehen neben **Mailingliste und Forum** aber auch täglich (im Durchschnitt 6 Stunden lang) Tutoren über einen **Videokonferenzdienst** zur Verfügung, welche bei Problemen über die in der IDE **integrierte Bildschirmübermittlung** die Studierenden interaktiv unterstützen können.



Konsequenter Einsatz, Erweiterung und Entwicklung von primär¹ **freier Software**, wodurch die Software lizenzfrei von den Studierenden verwendet werden kann:

VirtualBox 6.1 als Virtualisierungssoftware



Debian 10 als Gastbetriebssystem in der VM

Editor basierend auf **Atom 1.45.0**

GCC zum Übersetzen des Quelltextes und
AVRDUDE zur Programmierung der Hardware

Debuggen mit **GDB** und **AVaRICE**

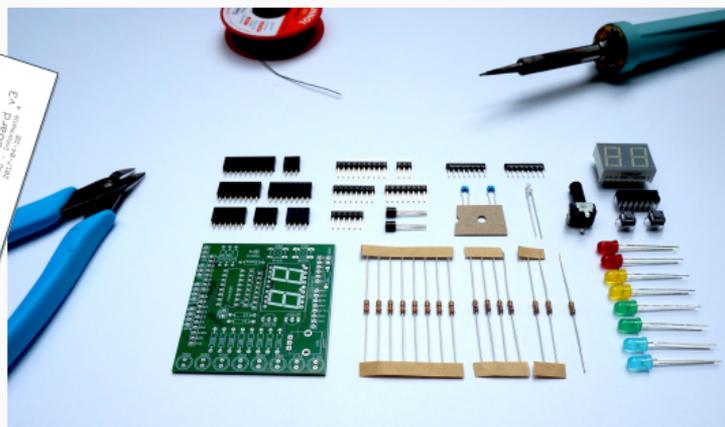
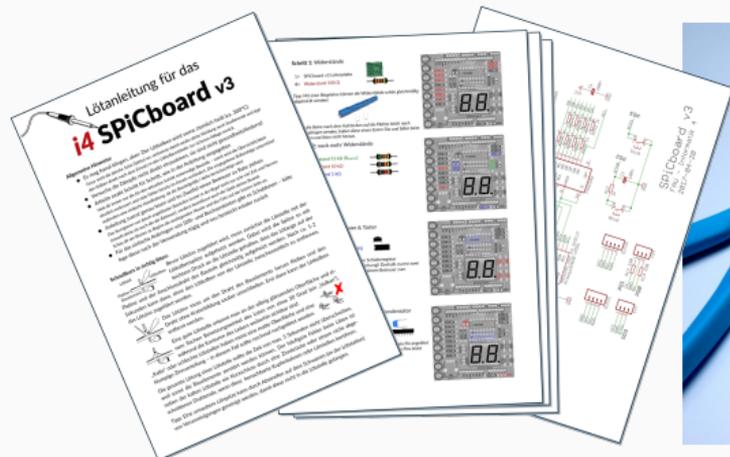
SPICboard AVR Simulation mittels **simavr**

Xpra Protokoll für den Webzugriff

¹Ausnahme: *VirtualBox Extension Pack* (zur Ansteuerung der SPICboard Hardware in der VM via USB), welches jedoch für den privaten und universitären Einsatz frei verwendbar ist.

Entwicklungsboard selber löten (optional)

- Für Anfänger **optimiertes Design** mit Hilfestellungen zum Nachbau
- Verwendung von günstigen und gut verfügbaren **Standardkomponenten** (Kostenpunkt für alle Bauteile im Einzelhandel ca. 20 €)
- Postalischer **Verleih** von SPiCboards und (bei Präsenz) **Lötpraktikum** für Anfänger
- Programmierung über virtuelle Maschine (aufgrund USB Weiterreichung keine Installation von Treibern im Host notwendig)



Weitere Informationen

sowie die in dieser Präsentation referenzierten Softwarelösungen, Videos zu Vorlesung & Übungen und Details zur Hardware sind über die Webseite der Lehrveranstaltung – ohne Zugangsbeschränkung – abrufbar:

www4.cs.fau.de/Lehre/SS20/V_SPIC/