

Systemnahe Programmierung in C

2 Organisation der Veranstaltung

J. Kleinöder, D. Lohmann, V. Sieh, P. Wägemann

Lehrstuhl für Informatik 4
Systemsoftware

Friedrich-Alexander-Universität
Erlangen-Nürnberg (FAU)

Sommersemester 2026

<http://sys.cs.fau.de/lehre/ss26>



- Inhalt und Themen
 - Grundlegende Konzepte der systemnahen Programmierung
 - Einführung in die Programmiersprache C
 - Unterschiede zu Java
 - Modulkonzept
 - Zeiger und Zeigerarithmetik
 - Softwareentwicklung auf „der nackten Hardware“ (ATmega- μ C)
 - Abbildung Speicher \leftrightarrow Sprachkonstrukte
 - Unterbrechungen (*interrupts*) und Nebenläufigkeit
 - Softwareentwicklung auf „einem Betriebssystem“ (Linux)
 - Betriebssystem als Ausführungsumgebung für Programme
 - Abstraktionen und Dienste eines Betriebssystems



- 36 Themenabschnitte
 - Foliensätze auf WWW-Server
 - Reihenfolge/Termin aus dem Semesterplan ersichtlich
 - → Voraussetzung für erfolgreiche Bearbeitung der Übungsaufgaben
- Fragen zur Vorlesung
 - am besten **sofort** stellen
 - im StudOn Thread *Fragen zur Vorlesung*
 - Beantwortung im Forum oder während der wöchentlichen Vorlesung/Tafelübung
- Ende des Semesters Klausurfragestunde
- **Tafel- und Rechnerübungen sind keine Ersatzvorlesungen!**



- Screencasts aus SS20
- Vorlesungsaufzeichnung aus SS19 existieren...

siehe

- WWW-Seite
- Video-Portal Uni

...sind aber z.T. (etwas) veraltet!

■ Teil A: Konzept und Organisation 1.2 - Kapitel 2:
Organisation [SS2020] 🔒



Dr. Volkmar Sieh
2020-04-20
IdM-login

Referenz ist die aktuelle Präsenzvorlesung!



- Tafelübung und Rechnerübung
 - Tafelübungen
 - Ausgabe und Erläuterung der Programmieraufgaben
 - Gemeinsame Entwicklung einer Lösungsskizze
 - Besprechung der Lösungen



- Tafelübung und Rechnerübung
 - Tafelübungen
 - Ausgabe und Erläuterung der Programmieraufgaben
 - Gemeinsame Entwicklung einer Lösungsskizze
 - Besprechung der Lösungen
 - Rechnerübungen
 - selbstständige Programmierung
 - Umgang mit Entwicklungswerkzeug
 - Betreuung durch Übungsbetreuer
 - Abgabegespräche für Bonuspunkte



- Tafelübung und Rechnerübung
 - Tafelübungen
 - Ausgabe und Erläuterung der Programmieraufgaben
 - Gemeinsame Entwicklung einer Lösungsskizze
 - Besprechung der Lösungen
 - Rechnerübungen
 - selbstständige Programmierung
 - Umgang mit Entwicklungswerkzeug
 - Betreuung durch Übungsbetreuer
 - Abgabegespräche für Bonuspunkte
- Termin:
 - Anmeldung über Waffel ab Donnerstag 18.04.2024 um 18:00 Uhr (siehe Webseite)
 - Bitte für T01 anmelden
 - Anmeldung freiwillig, nur für Bonuspunkte nötig
 - **Separate** Übung für SLP

Zur Übungsteilnahme wird ein gültiger Login im Linux-CIP gebraucht!



WARNING!

Im WS wird es **keine** Übungen für Wiederholer mehr geben!

WARNING!



Programmieraufgaben

- Praktische Umsetzung des Vorlesungsstoffs
 - Acht Programmieraufgaben ↪ 2-14
 - Teilweise Gruppenabgaben
- Lösungen per SPiC-IDE abgeben
 - Lösung wird durch Skripte überprüft
 - Ein Tutor wird die Abgabe mit Euch besprechen
 - LLMs sind zur Bearbeitung **nicht** gestattet (ChatGPT, Perplexity, etc.)



Programmieraufgaben

- Praktische Umsetzung des Vorlesungsstoffs
 - Acht Programmieraufgaben ↪ 2-14
 - Teilweise Gruppenabgaben
- Lösungen per SPiC-IDE abgeben
 - Lösung wird durch Skripte überprüft
 - Ein Tutor wird die Abgabe mit Euch besprechen
 - LLMs sind zur Bearbeitung **nicht** gestattet (ChatGPT, Perplexity, etc.)
- ★ Abgabe der Übungsaufgaben ist **freiwillig**; ↪ 2-13
es können jedoch bis zu **10% Bonuspunkte**
für die Prüfungsklausur erarbeitet werden!



Programmieraufgaben

- Praktische Umsetzung des Vorlesungsstoffs
 - Acht Programmieraufgaben ↪ 2-14
 - Teilweise Gruppenabgaben
 - Lösungen per SPiC-IDE abgeben
 - Lösung wird durch Skripte überprüft
 - Ein Tutor wird die Abgabe mit Euch besprechen
 - LLMs sind zur Bearbeitung **nicht** gestattet (ChatGPT, Perplexity, etc.)
 - ★ Abgabe der Übungsaufgaben ist **freiwillig**; ↪ 2-13
es können jedoch bis zu **10% Bonuspunkte**
für die Prüfungsklausur erarbeitet werden!
- Plagiate können zum **Verlust aller Bonuspunkte** führen.



Programmieraufgaben

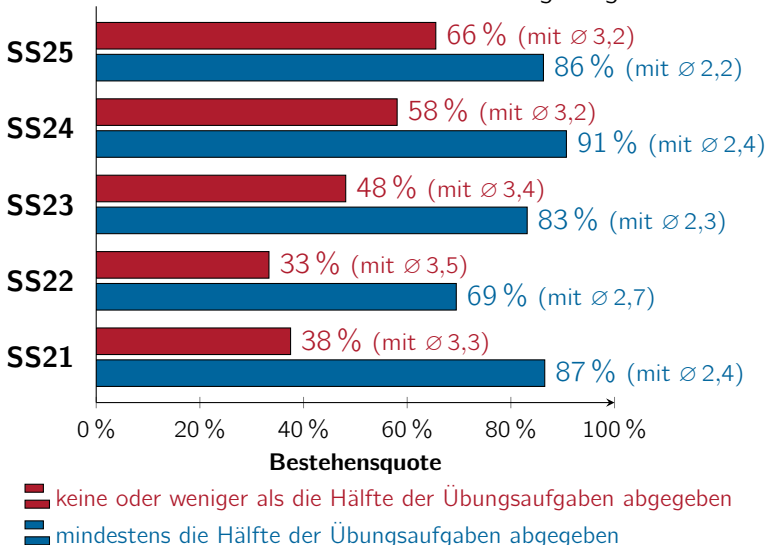
- Praktische Umsetzung des Vorlesungsstoffs
 - Acht Programmieraufgaben ↪ 2-14
 - Teilweise Gruppenabgaben
 - Lösungen per SPiC-IDE abgeben
 - Lösung wird durch Skripte überprüft
 - Ein Tutor wird die Abgabe mit Euch besprechen
 - LLMs sind zur Bearbeitung **nicht** gestattet (ChatGPT, Perplexity, etc.)
 - ★ Abgabe der Übungsaufgaben ist **freiwillig**; ↪ 2-13
es können jedoch bis zu **10% Bonuspunkte**
für die Prüfungsklausur erarbeitet werden!
- Plagiate können zum **Verlust aller Bonuspunkte** führen.

Unabhängig davon ist die Bearbeitung der
Übungen **dringend empfohlen!**



Bestehensquote der Klausur

nach Aktivität der Teilnehmer bei den Übungsaufgaben



Programmieren mit KI

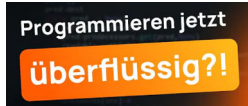
Programmieren jetzt

überflüssig?!

Quelle: YouTube



Programmieren mit KI



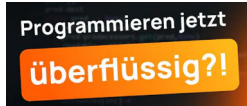
Quelle: YouTube

unsicherer Code:

KI-generierter Code kann versteckte Sicherheitslücken enthalten



Programmieren mit KI



Quelle: YouTube

unsicherer Code:

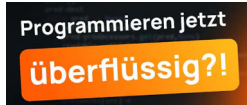
KI-generierter Code kann versteckte Sicherheitslücken enthalten

Verlust Fähigkeiten:

Entwickler verlassen sich zunehmend auf KI



Programmieren mit KI



Quelle: YouTube

unsicherer Code:

KI-generierter Code kann versteckte Sicherheitslücken enthalten

Verlust Fähigkeiten:

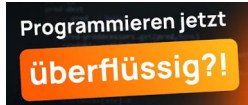
Entwickler verlassen sich zunehmend auf KI

Wartungsaufwand:

KI-generierter Code führt zu schwer wartbarer Codebasis



Programmieren mit KI



Quelle: YouTube

unsicherer Code:

KI-generierter Code kann versteckte Sicherheitslücken enthalten

Verlust Fähigkeiten:

Entwickler verlassen sich zunehmend auf KI

Wartungsaufwand:

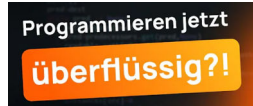
KI-generierter Code führt zu schwer wartbarer Codebasis

Over-Reliance:

blindes Vertrauen auf KI statt Probleme zu verstehen



Programmieren mit KI



Quelle: YouTube

unsicherer Code:

KI-generierter Code kann versteckte Sicherheitslücken enthalten

Verlust Fähigkeiten:

Entwickler verlassen sich zunehmend auf KI

Wartungsaufwand:

KI-generierter Code führt zu schwer wartbarer Codebasis

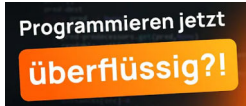
Over-Reliance:

blindes Vertrauen auf KI statt Probleme zu verstehen

Fazit: KI ist ein mächtiges Werkzeug, ersetzt aber (noch) nicht die Notwendigkeit von Expertenwissen und sorgfältigem Code-Review.



Programmieren mit KI



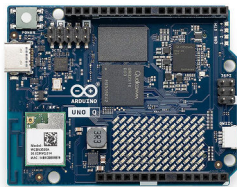
Quelle: YouTube

Fazit: KI ist ein mächtiges Werkzeug, ersetzt aber (noch) nicht die Notwendigkeit von Expertenwissen und sorgfältigem Code-Review.

Wer Code von anderen (oder einer KI) reviewen will, muss selber programmieren können!



„Arduino kenn ich schon...“

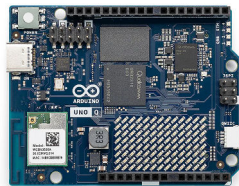


Quelle: arduino.cc

SPiC:



„Arduino kenn ich schon...“



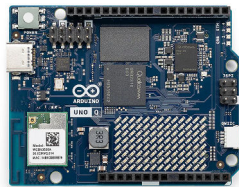
Quelle: arduino.cc

- Board ähnlich dem SPiC-Board mit vielen Erweiterungsmöglichkeiten

SPiC:



„Arduino kenn ich schon...“



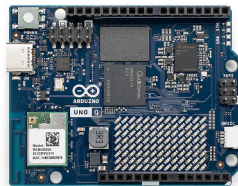
Quelle: arduino.cc

- Board ähnlich dem SPiC-Board mit vielen Erweiterungsmöglichkeiten
- Programmierung mit **C++** (Sub-Set)

SPiC:



„Arduino kenn ich schon...“



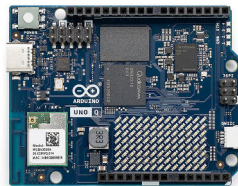
Quelle: arduino.cc

- Board ähnlich dem SPiC-Board mit vielen Erweiterungsmöglichkeiten
- Programmierung mit **C++** (Sub-Set)
- Programmierung mit **externen** Klassen/Bibliotheken

SPiC:



„Arduino kenn ich schon...“



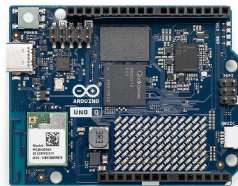
Quelle: arduino.cc

- Board ähnlich dem SPiC-Board mit vielen Erweiterungsmöglichkeiten
- Programmierung mit **C++** (Sub-Set)
- Programmierung mit **externen** Klassen/Bibliotheken
- viele Tutorials im Netz

SPiC:



„Arduino kenn ich schon...“



Quelle: arduino.cc

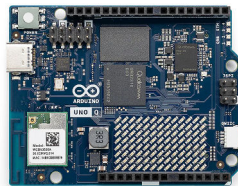
- Board ähnlich dem SPiC-Board mit vielen Erweiterungsmöglichkeiten
- Programmierung mit **C++** (Sub-Set)
- Programmierung mit **externen** Klassen/Bibliotheken
- viele Tutorials im Netz

SPiC:

- Programmierung in **C**



„Arduino kenn ich schon...“



Quelle: arduino.cc

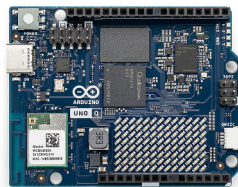
- Board ähnlich dem SPiC-Board mit vielen Erweiterungsmöglichkeiten
- Programmierung mit **C++** (Sub-Set)
- Programmierung mit **externen** Klassen/Bibliotheken
- viele Tutorials im Netz

SPiC:

- Programmierung in **C**
- **ohne** externe Bibliotheken



„Arduino kenn ich schon...“



Quelle: arduino.cc

- Board ähnlich dem SPiC-Board mit vielen Erweiterungsmöglichkeiten
- Programmierung mit **C++** (Sub-Set)
- Programmierung mit **externen** Klassen/Bibliotheken
- viele Tutorials im Netz

SPiC:

- Programmierung in **C**
- **ohne** externe Bibliotheken
- Entwicklung **eigener** Bibliotheken

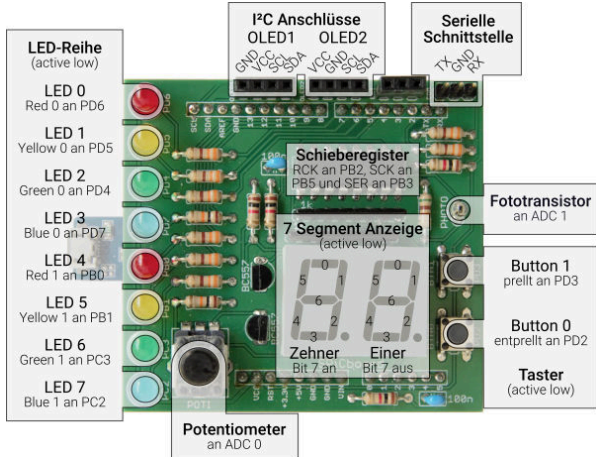


Übungsplattform: Das SPiCboard

- ATmega328- μ C
- USB-Anschluss
- 8 LEDs
- 2 7-Seg-Elemente
- 2 Taster
- 1 Potentiometer
- 1 Fotosensor
- OLED Display

optional:

- OLED Display



- Ausleihe während Rechnerübung
- Oder noch besser \leftrightarrow selber Löten
- Alternativ: Entwicklung im Simulator



- Die FSI EEI, FSI ME sowie das FabLab bieten einen „Lötabend“ für die Teilnehmer der Veranstaltung an
 - Teilnahme ist freiwillig
 - (Erste) Löterfahrung sammeln beim Löten eines eigenen SPiCboards
 - Insgesamt 3 Termine (in KW 17)
- Anmeldung über Waffel **notwendig**, da begrenzte Plätze:
ab Donnerstag 16.04.2026 um 18:00 Uhr (siehe Webseite)
- Kostenlos für Teilnehmer dieser Veranstaltung
(finanziert aus Studienzuschüssen)

Der bei der Anmeldung gewählte Termin ist verbindlich!

■ Prüfung (Klausur)

- Termin: voraussichtlich Ende Juli / Anfang August
- Dauer: 60 min (GSPiC) bzw. 90 min (SPiC und InfoEEI)
- Inhalt: Fragen zum Vorlesungsstoff + Programmieraufgabe(n)

■ Klausurnote \mapsto Modulnote

- Bestehensgrenze (in der Regel): 50% der möglichen Klausurpunkte (KP)
- Falls **bestanden** ist eine Notenverbesserung möglich durch Bonuspunkte aus den Programmieraufgaben
 - Basis (Minimum): 20% der möglichen Übungspunkte (ÜP)
 - Bonuspunkte werden gleichmäßig auf Intervall [20%;80%] der möglichen ÜP aufgeteilt
- ~> 80%-100% der möglichen ÜP \mapsto +10% der möglichen KP



Semesterplanung

KW	Mo	Di	Mi	Do	Fr	Themen
	13.04.	14.04.	15.04.	16.04.	17.04.	
15			VL1	VL2		Einführung, Organisation, Java versus C Softwarearchitekturen und Abstraktion, Sprachüberblick, Einfache Datentypen
	20.04.	21.04.	22.04.	23.04.	24.04.	
16		A1		VL3		Operatoren und Ausdrücke, Kontrollstrukturen, Funktionen, Variablen A1 (blink)
	27.04.	28.04.	29.04.	30.04.	01.05.	
17		A2		VL4	Tag der Arbeit	Präprozessor, Programmstruktur und Module, Zeiger und Felder A2 (snake)
	04.05.	05.05.	06.05.	07.05.	08.05.	
18		A3		VL5		Verbundtypen, Mikrocontroller Vorbemerkungen, Mikrocontroller Prozessor, Mikrocontroller Peripherie A3 (led-modul)
	11.05.	12.05.	13.05.	14.05.	15.05.	
19				Christi Himmelfahrt		
	18.05.	19.05.	20.05.	21.05.	22.05.	
20		A4		VL6		Unterbrechungen, Unterbrechungen Beispiel, Unterbrechungen Nebenläufigkeit A4 (spiel)
	25.05.	26.05.	27.05.	28.05.	29.05.	
21	Pfingstmontag			VL7		Dynamische Speicherallokation, Speicherorganisation, Speicherorganisation Stack, Speicherorganisation Zusammenfassung
	01.06.	02.06.	03.06.	04.06.	05.06.	
22		A5		Fronleichnam		A5 (ampel)
	08.06.	09.06.	10.06.	11.06.	12.06.	
23			Pfingstdienstag	VL8		Ergänzungen Zeiger, Ergänzungen Ein-/Ausgabe, Ergänzungen Fehlerbehandlung, Betriebssysteme
	15.06.	16.06.	17.06.	18.06.	19.06.	
24		A6		VL9		Datensysteme Einleitung, Datensysteme Unix A6 (concat)
	22.06.	23.06.	24.06.	25.06.	26.06.	
25		A7		VL10		Programme und Prozesse, Programme und Prozesse Unix, Signale A7 (printv)
	29.06.	30.06.	01.07.	02.07.	03.07.	
26		A8		VL11		Multiprozessoren, Nebenläufige Fäden, Nebenläufige Fäden Praxis A8 (mesh)
	06.07.	07.07.	08.07.	09.07.	10.07.	
27				VL12		Klausurvorbereitung
	13.07.	14.07.	15.07.	16.07.	17.07.	
28				Q&A		Fragestunde

Details: <http://sys.cs.fau.de/lehre/ss26>

Dozent Vorlesung



Dr. Volkmar Sieh

Organisatoren des Übungsbetriebs



Arne Vogel



Maxim Ritter
von Onciul



Übungsleiter



Quirin Gebert



Janick Martin



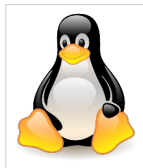
Lea Klein



Simeon Zobel



Matthias
Schmidbauer



Benedikt Klein





Conner Siebert



Bei Fragen oder Problemen

- Vorlesungs- und Übungsfolien konsultieren
- Häufig gestellte Fragen (FAQ) und Antworten siehe Webseite
- Rechnerübungen
- StudOn Forum
→ https://www.studon.fau.de/studon/goto.php?target=frm_5700999
- Bei speziellen Fragen Mail an Mailingliste
→ alle Übungsleiter i4spic@i4.cs.fau.de (inhaltlich)
→ wiss. Mitarbeiter i4spic-orga@i4.cs.fau.de (organisatorisch)

Chat für euren Austausch:

<https://to.chat.fau.de/#/room/#spic:fau.de>

