

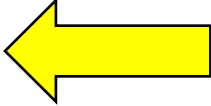
**Folien zur Vorlesung  
Grundlagen systemnahes Programmieren  
Sommersemester 2016  
(Teil 1)**

**Prof. Dr. Franz Korf**

Franz.Korf@haw-hamburg.de

# Kapitel 1: Einführung

## Gliederung

- Steckbriefe 
- Formalien und Kommentare
- Inhalt der Vorlesung
- Zusammenfassung

## Die Folien zu dieser Vorlesung basieren auf Ausarbeitungen von

- Heiner Heitmann
- Reinhard Baran
- Andreas Meisel

## „Steckbrief“ von Franz Korf

### Beruflicher Werdegang

Informatik Studium in Aachen & Promotion in Oldenburg

- HW, SW, Formale Verifikation, Anwendungen

Fujitsu Siemens Computers (Paderborn)

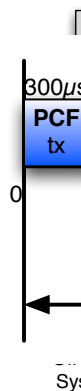
- ASIC Design Prozess, System und RTL Simulation (VHDL, Verilog, C), Synthese, Simulationsumgebungen
- Embedded SW
- OEM / ODM Entwicklung

Hochschule für Angewandte Wissenschaften in Hamburg

- Lehre in der Informatik
- Embedded Systems, RTOS, R-ETH, Time-Triggered Systems, FAUST, Kommunikationssysteme in Auto, CoRE (<http://core.informatik.haw-hamburg.de/en/>)

# CORE (Communication over Realtime Ethernet)

- Kommunikationslösungen für zeitkritische Anwendungen auf Basis der Ethernet-Technologie
- Backbone-Infrastruktur für die Kommunikation in Fahrzeugen
- Simulation von RT Ethernet Netzwerken
- Analyse und Bewertung von Metriken
- RT Ethernet basierte Steuergeräte
- Aufbau von RT Ethernet Netzwerken
- Automotive Anwendungen
- Interdisziplinäre Aufgaben



Werkstatt-  
tester    Augen-  
tester    Fertigungs-  
-tester    ...    kations-  
tools

## **„Steckbrief“ von Ihnen**

- Haben Sie bereits C Erfahrungen?
- Wer hat in den Semesterferien programmiert?
- Wer hat in C programmiert?
- Wer ist im zweiten Semester?
- Wer hat Zugriff auf einen privaten Rechner?
- Werden Sie aufgrund von Erfahrungen aus dem ersten Semester das zweite Semester „anders angehen“?
- Wer arbeitet parallel zum Studium ( >10 Std./Wo)?

## Ziele und Wünsche für diese Veranstaltung

### Von mir

- Mit Freude lernen und Erlerntes behalten
  - Sie arbeiten aus sich heraus
  - Sie haben typische Erfolgsfaktoren der Informatik erlebt, z.B.: Struktur, Disziplin & Ordnung, Vorgehensweise: denken – diskutieren – programmieren
  - Themengebiete, die Sie am Ende des Kurs beherrschen sollten:
    - systemnahe Programmierung
    - prozeduralen Programmiersprachen
    - Transferleistung: Inhalte aus den Programmiervorlesungen des ersten Semesters in die C Welt retten und erfolgreich umsetzen

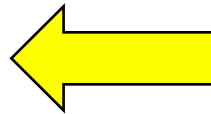
### Von Ihnen

- ???

# Kapitel 1: Einführung

## Gliederung

- Steckbriefe
- Formalien und Kommentare
- Inhalt der Vorlesung
- Zusammenfassung



## Formalien und Kommentare

### Freie Übungszeiten im TI Labor

- s. HomePage des Labors

### Sprechstunde

- nach Vereinbarung
- Sprechen Sie mich direkt an (nach der Vorlesung, im Praktikum, ...)

Bitte nutzen Sie die **Mailing Liste** zu dieser Veranstaltung

### Feedback:

- Feedback ist entscheidend
- Kommentare, Kritik, Verbesserungsvorschläge jeglicher Art sind stets willkommen. Schicken Sie mir eine E-Mail, sprechen Sie mich direkt an, ...



## Formalien und Kommentare

### Tutorium

- **Daniel Sarnow** bietet ein Tutorium zu diese Veranstaltung an
- Sehr gute Ergänzung zur Vorlesung
- Entscheidende Hilfe bei der Vorbereitung des Praktikums

### Unterlagen zur Vorlesung, zum Praktikum und zum Tutorium

- stehen unter EMIL zur Veranstaltung Grundlagen Systemnahen Programmierens (GS/GSP) im Netz
- **Emil Kennwort** ist **GSP**
- Übungen und Ergänzungen an der Tafel

## Formalien und Kommentare

### Praktikum

#### ➤ Anwesenheitspflicht

- Anmeldung sollte schon erledigt sein (sonst Silke Behn ansprechen, 7 Etage)
- Je zwei Personen arbeiten zusammen
- Kein Schnelltest, **aber** Versuche müssen **vorbereitet** werden!
  - Schriftliche Ausarbeitung der Konzepte und des Lösungsansatzes müssen am Anfang des Praktikumstermins zu einer Aufgabe vorliegen.  
**Ohne Konzept kein erfolgreiche Abgabe der Aufgabe**
  - Stichprobenartige Überprüfung der Ausarbeitungen.
  - Versuch darf nicht durchgeführt werden, wenn Ausarbeitung fehlt oder nicht erklärt werden kann. Schlechte Ausarbeitungen müssen im Praktikum verbessert werden.
  - Die Aufgabe fast fertig sein, dann gibt es immer noch genug zu tun.

## Formalien und Kommentare

### Praktikum (Fortsetzung)

- **Erfolgreiche Abgabe der Praktikumsaufgaben**
  - Alle Unterlagen am Tag vor dem Praktikum via e-mail an Silke Behn und mich. Schicken Sie auch eine Kopie an Ihren Praktikumpartner.
  - Alle Quellen müssen zitiert werden (Bücher, Zusammenarbeit, ...)
  - Diskussion der Aufgabe mit den Veranstaltern.
  - Sie müssen die Aufgabe im Detail erklären können.
  - Halten Sie die Abgabetermine ein.

## Formalien und Kommentare

### Praktikum (Fortsetzung)

#### ➤ Struktur des Codes

- Der Code hat einen Datei-**Header**, der u.a. folgende Informationen enthält:
  - **Praktikum** (z.B. GSP)
  - **Semester** (z.B. SS2016)
  - **Namen** der Team-Mitglieder mit Matrikel-Nr.
  - **Aufgaben-Nr.** (z.B. Aufgabe A1)
- Der Code ist gut **kommentiert**. Gerne auch DoxyGen.
- Der Code ist gut **strukturiert**.
- Der Code weist einen einheitlichen und durchgängigen **Coding-Style** auf (insbesondere konsequente Sichtformatierung/Einrückung).  
Ein Coding Style steht unter Emil.
- **Ohne Coding Style kein erfolgreiche Abgabe der Aufgabe**

## Coding Style – generelle Regeln

### **Keep it simple.**

- Diese Regel besagt zum Beispiel: Verwenden Sie keine komplexen Sprachkonstrukte, wenn einfache auch zum Ziel führen.

### **Don't be clever.**

- Diese Regel besagt zum Beispiel: Verwenden Sie nur bei Bedarf die Tricks, die Sie vielleicht gerade erst gelernt haben. Nicht jeder kennt diese Tricks.

### **Be explicit – write your program for people.**

- C Programme werden nicht nur von C Experten gelesen und gewartet. Schreiben Sie Programme so, dass andere Personen dieses leicht lesen und verstehen können – dann können Sie Ihr Programm auch nach einem Jahr noch leicht lesen und verstehen.
- Verständliche Programme beeindrucken, unverständliche Programme schrecken ab.

## Coding Style – weiteres Vorgehen

- Unter Emil steht ein Coding Style, der auf Ihre Situation im Praktikum ausgelegt ist.
- Lesen Sie diese Folien stets vor der Programmierung.
- Gehen Sie Ihr Programm am Ende nochmals mit Blick auf den Coding Style durch.
- **Ergebnis:** Nach der dritten Aufgabe beherrschen Sie die technischen Aspekte des Coding Styles.

## Formalien und Kommentare

### Modulprüfung

- Voraussetzung (PVL): Praktikum bestanden
- Eine benotete Prüfung, die aus zwei Teilen besteht:
  - Fragenteil
  - Prüfung am Rechner, Lösung einer Programmieraufgabe

### Herausforderung

- Selbstständige Lösung einer Programmieraufgabe

### Empfehlungen

- Arbeiten Sie aktiv an der Lösung der Praktikumsaufgabe
- Wer die Aufgaben nicht selbst löst, bekommt in der Klausur und der Programmierprüfung Probleme!
- Schreiben Sie jede Woche ein zusätzliches C Programm!

## Literatur & Software

### ➤ C Literatur Ihrer Wahl, z.B.

- Joachim Groll, Ulrich Bröckl, Manfred Dausmann:  
C als erste Programmiersprache : Vom Einsteiger zum Profi,  
(6. Auflage), Teubner, 2008
- B.W. Kernighan, D.M. Ritchie: Programmieren in C, (2. Auflage), Hanser, 1990
- J. Wolf: C von A bis Z, Galileo Press
- Carsten Vogt: C für Java-Programmierer, Hanser, 2007.

### ➤ Grundlagenbücher zu Programmiertechniken und Algorithmen

### ➤ Unterlagen für Mikrocontroller STM32F4-Series (STM32F417ZGT6)

- Reference manual [http://www.st.com/st-web-ui/static/active/en/resource/technical/document/reference\\_manual/DM00031020.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/reference_manual/DM00031020.pdf)
- ARM Cortex-M4 Processor: Technical Reference Manual  
[http://infocenter.arm.com/help/topic/com.arm.doc.ddi0439d/DDI0439D\\_cortex\\_m4\\_processor\\_r0p1\\_trm.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ddi0439d/DDI0439D_cortex_m4_processor_r0p1_trm.pdf)



## Ein Zitat

**Goethe:** „*Denn wir behalten von unseren Studien am Ende nur das, was wir praktisch anwenden.*“

Daraus ergibt sich

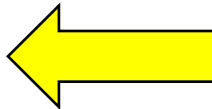
- Nehmen Sie an der Vorlesung und am Praktikum aktiv teil.
- Arbeiten Sie die Vorlesung sofort nach.
- Selbststudium ist mehr als wichtig.
- Bereiten Sie sich intensiv auf das Praktikum vor.
- Erstellen Sie Ihre persönliche Mitschrift.
- Rechnen / programmieren Sie Beispiele durch.

**Tipp:** Praktikum, Klausur und Vorlesung sind eng miteinander verbunden.

# Kapitel 1: Einführung

## Gliederung

- Steckbriefe
- Formalien und Kommentare
- Inhalt der Vorlesung
- Zusammenfassung



# Inhalt der Vorlesung

## Voraussetzungen:

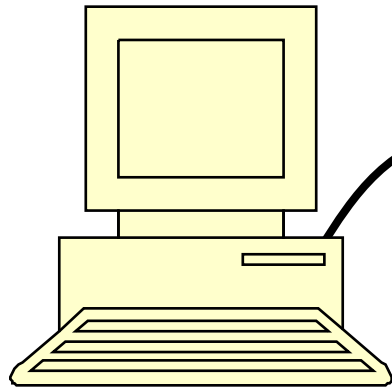
- Inhalte der Veranstaltungen: Programmieren I und Programmiertechnik
- Inhalte der Veranstaltung: Grundkurs Technische Informatik
  - Grundlegende Funktionsweise von Computern.
  - Welchen Zweck haben Speicher, CPU, Busse, IO-Einheiten?
  - Adressierungsarten
  - ...

## Inhalt der Vorlesung (Fortsetzung)

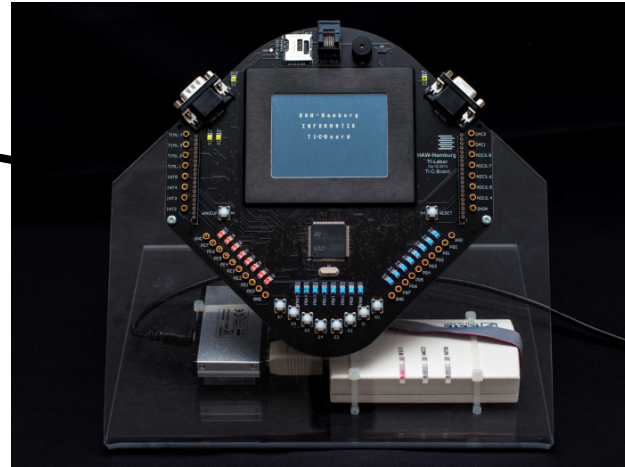
### Themengebiete:

- Methoden und Techniken zur prozeduralen - und maschinennahen Programmierung
- Relevante Konzepte der hardwarenahen Programmiersprache C, die für systemnahe Programmierung relevant sind.
- Speicherverwaltung auf Hochsprachen- und Maschinenebene
- C Projekte: Verwaltung, Modultechnik, Bibliotheken, Standardbibliotheken
- Interfaces zur Verzahnung von Hochsprachen und Assembler
- elementare Zeitmessungen

## Das Laborsystem



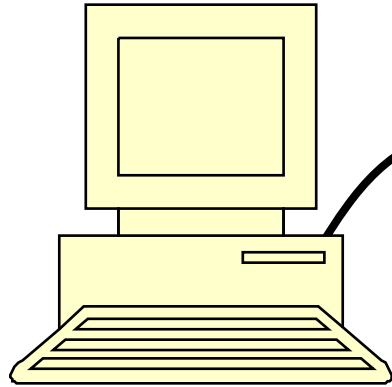
Entwicklungsumgebung (Host)



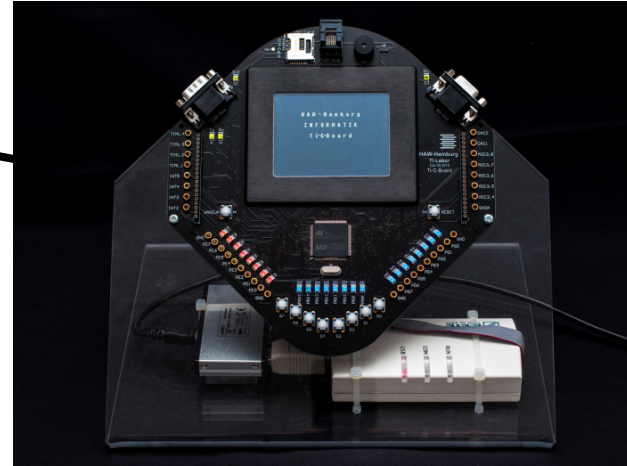
Zielsystem (Target)

- Crossentwicklung
- Target:
  - Experimentierboard, Kein Betriebssystem, Kein Massenspeicher
- Host:
  - PC mit Editor, Compiler und Linker für ARM-CPU

## Crossentwicklung



Entwicklungsumgebung (Host)



Zielsystem (Target)

- Editor, Compiler, Linker, Debugger, Bibliotheken .. liegen auf dem Host
- Software wird auf dem Host erzeugt, dann auf das Target geladen und dort ausgeführt
- Debugging: Die Software läuft auf dem Target, der Debugger läuft auf dem Host. Interface zur Steuerung der SW auf dem Target durch den Debugger auf dem Host.

# Kapitel 1: Einführung

## Zusammenfassung

