

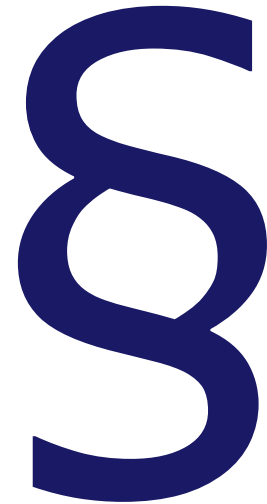
# Software for Embedded Systems

Prof. Dr.-Ing. Bernd-Christian Renner



# Legal Information

- Participants may not record
  - video
  - audio
- Participants give consent that TUHH
  - records video and audio
  - saves transcripts of chats
  - has the right to make all material (including recordings and chats) available online (StudIP, TUHH mediaserver, etc.) to course participants
- Report issues and misconfigurations  
(do not exploit them)





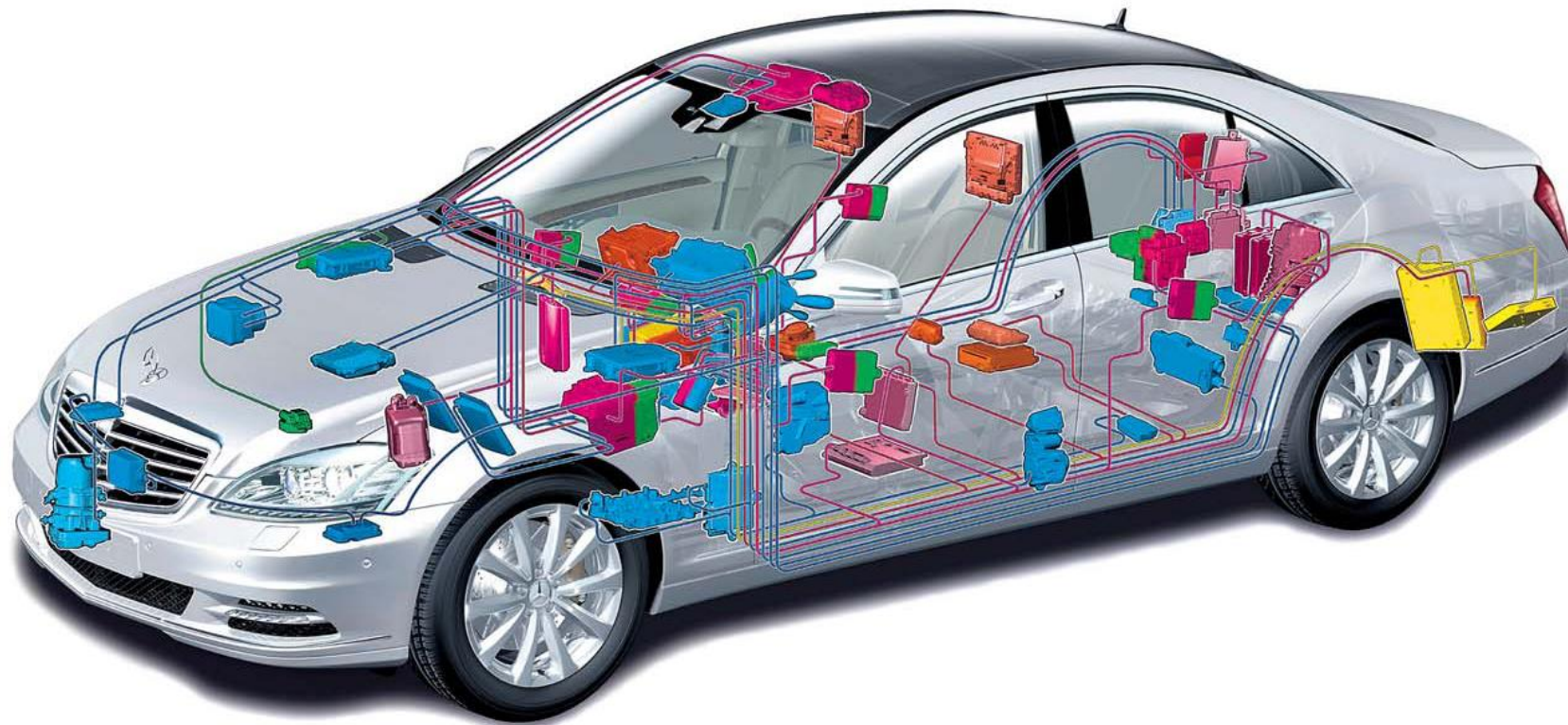


# Motivation

# Goals of the Module

- Give an overview of software for embedded systems
- Communicate the principles of software for
  - microcontrollers and
  - connecting peripherals
- Become acquainted with a real embedded system and be familiar with writing software for it
- Be proficient in state machine models and concurrent process models
- Know schedulers and operating systems for embedded systems

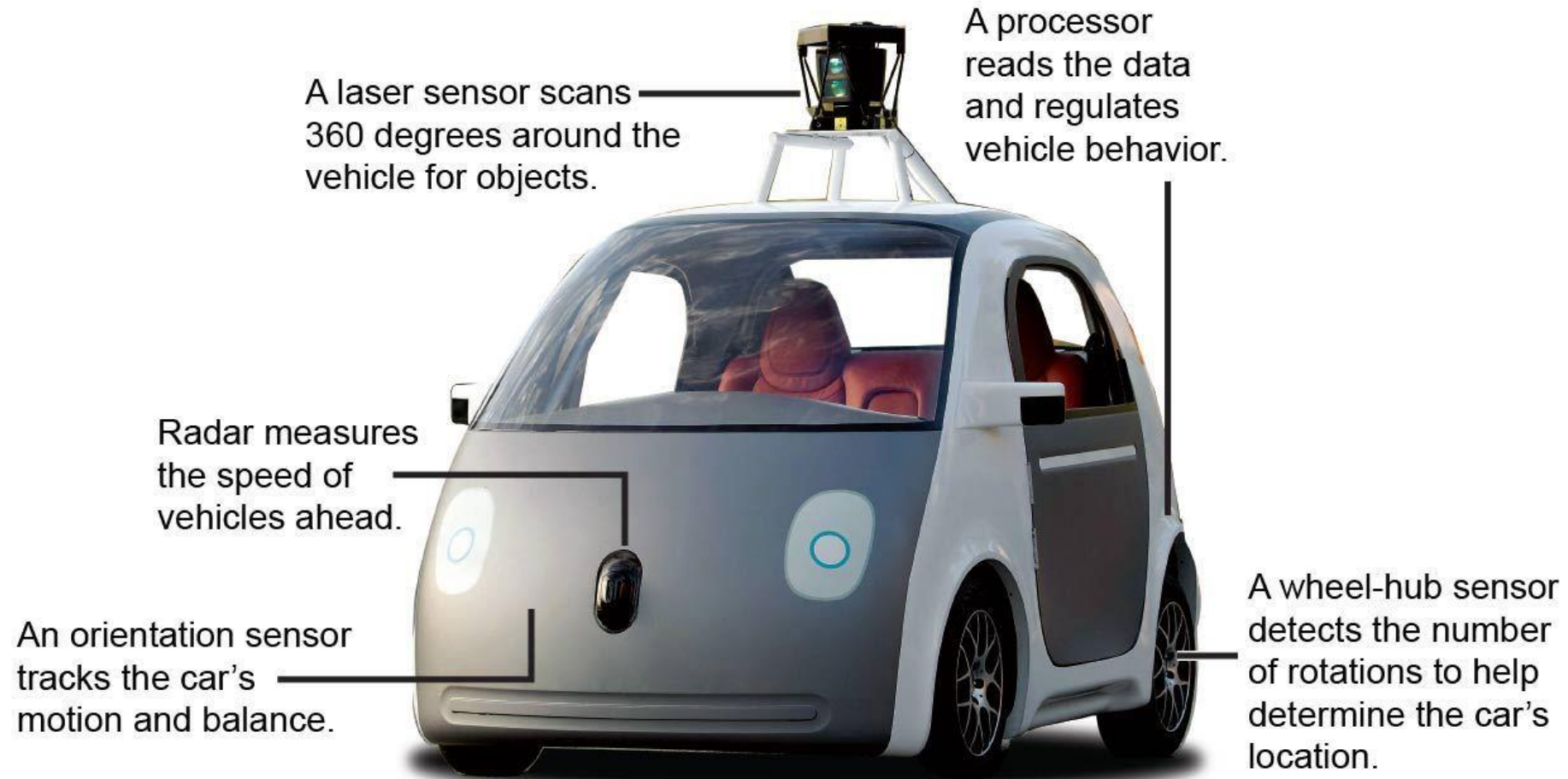
# Automotive Industry



Source: Mercedes-Benz



# Autonomous Cars



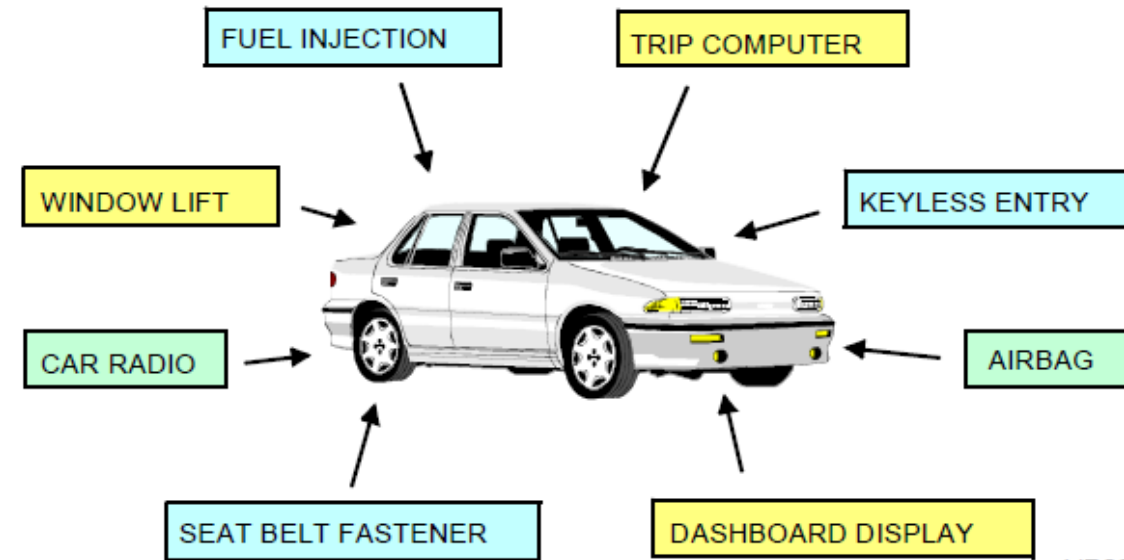
Source: Google

Raoul Rañoa / @latimesgraphics  
Source: Google

# A “short list” of Embedded Systems

And the list goes on...

Anti-lock brakes	Modems
Auto-focus cameras	MPEG decoders
Automatic teller machines	Network cards
Automatic toll systems	Network switches/routers
Automatic transmission	On-board navigation
Avionic systems	Pagers
Battery chargers	Photocopiers
Camcorders	Point-of-sale systems
Cell phones	Portable video games
Cell-phone base stations	Printers
Cordless phones	Satellite phones
Cruise control	Scanners
Curbside check-in systems	Smart ovens/dishwashers
Digital cameras	Speech recognizers
Disk drives	Stereo systems
Electronic card readers	Teleconferencing systems
Electronic instruments	Televisions
Electronic toys/games	Temperature controllers
Factory control	Theft tracking systems
Fax machines	TV set-top boxes
Fingerprint identifiers	VCR's, DVD players
Home security systems	Video game consoles
Life-support systems	Video phones
Medical testing systems	Washers and dryers



Embedded Systems in Automotive



# Organization

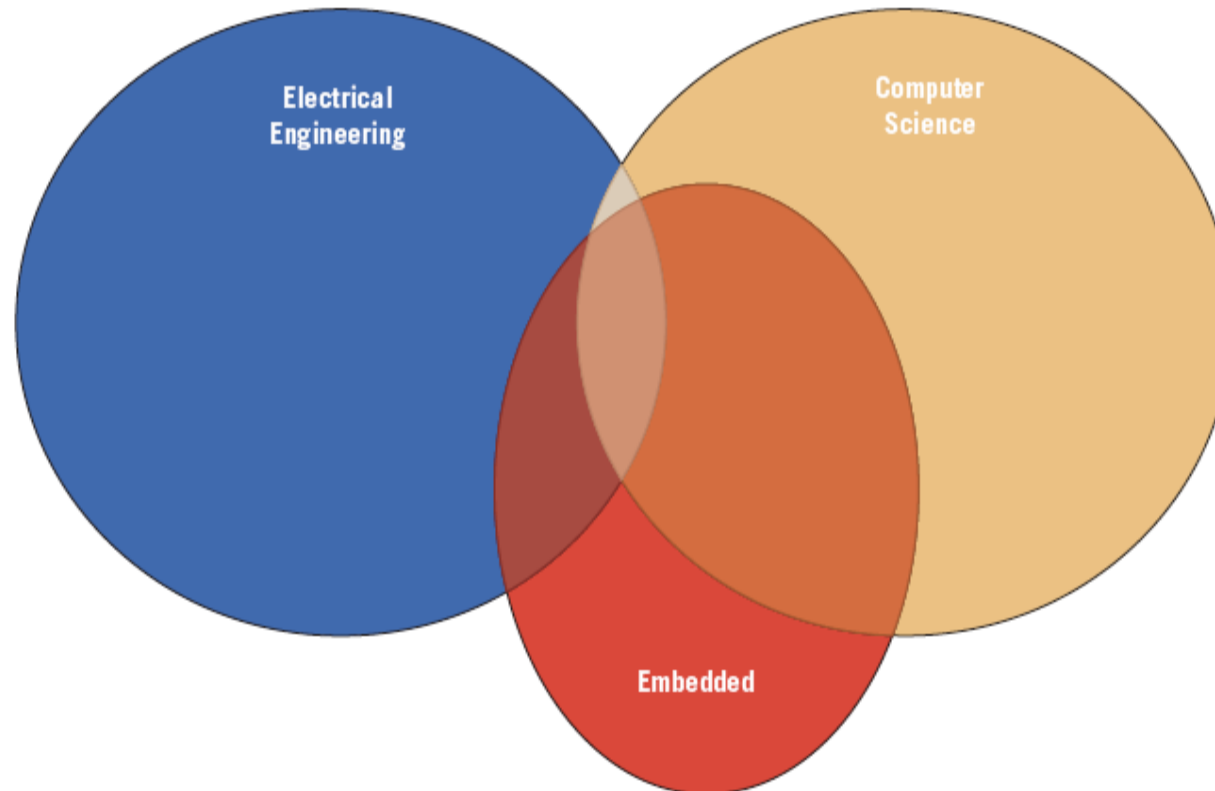


# Content

1. Introduction
2. General-Purpose Processors
3. Programming the Atmel AVR
4. Interrupts
5. Standard Single Purpose Processors: Peripherals
6. Operating Systems for Embedded Systems
7. Memory
8. Finite-State Machines
9. Boot Loader and Power Management
10. Real-Time Embedded Systems
- Appendix: C for Embedded Systems

# Embedded Systems Curricula

- Overlap with Electrical Engineering and Computer Science



# Previous Knowledge & Skills

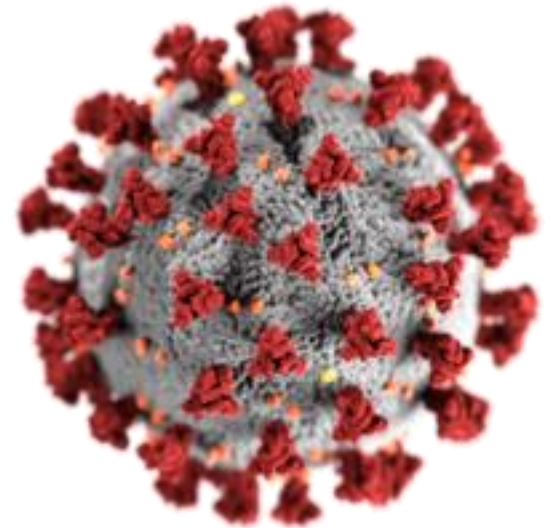
- Required
  - Very good knowledge of and practical experience with the C programming language!
  - Knowledge of OS concepts, interrupts, assembler, development tool chains, etc.
  - Basic understanding of microprocessors
- Recommended
  - Bachelor in computer science or similar degree program
  - Basic knowledge of software engineering





# A Changed World

- Online teaching and learning
- Many changes
- New to all of us
- Patience, respect, and modesty



Source: Wikipedia

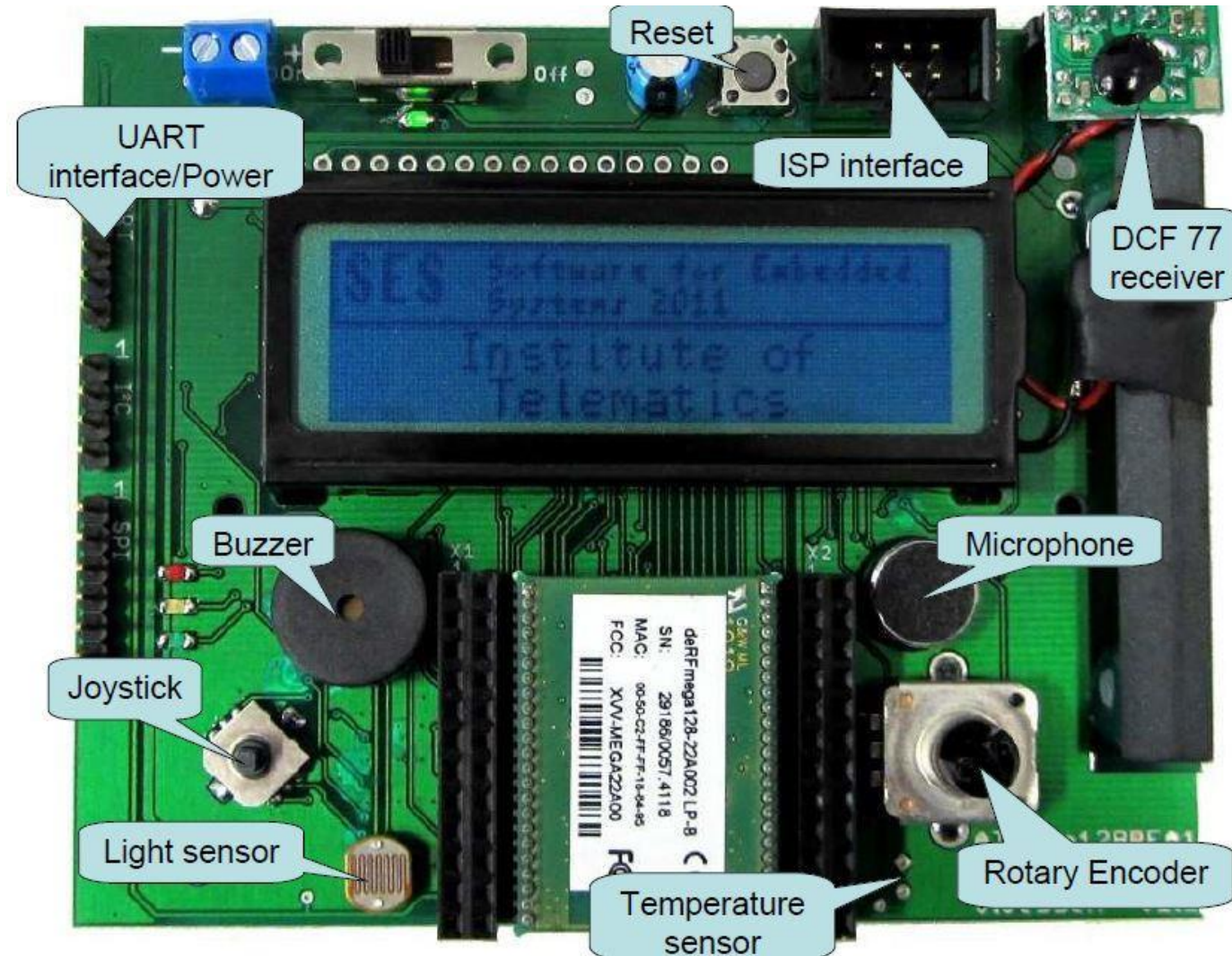
*"Presume not  
that I am the thing I was."*

William Shakespeare, Henry IV, Part 2

# Lecture

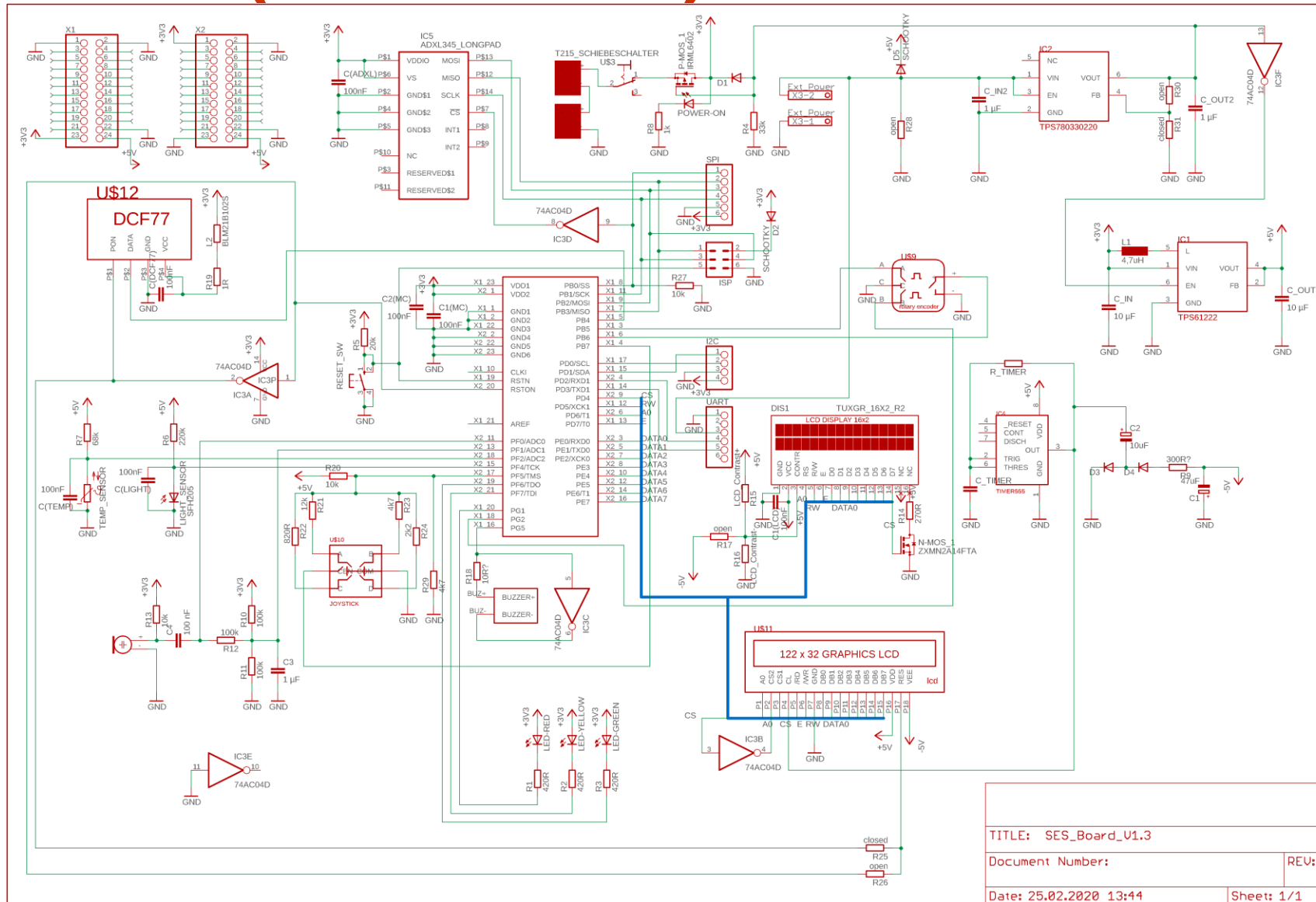
- Thursday, 09:45 – 11:15
- April 23 to July 23 (not on May 21 and June 4)
- Online, live via Zoom (tentative)
  - Chat? Mattermost?
  - Interactive, if possible
- Material online (StudIP)
  - Slides
  - Remarks and annotations made in lecture (planned)
- Video and audio recordings (planned)
  - Available via TUHH servers, restricted audience

# SES Board

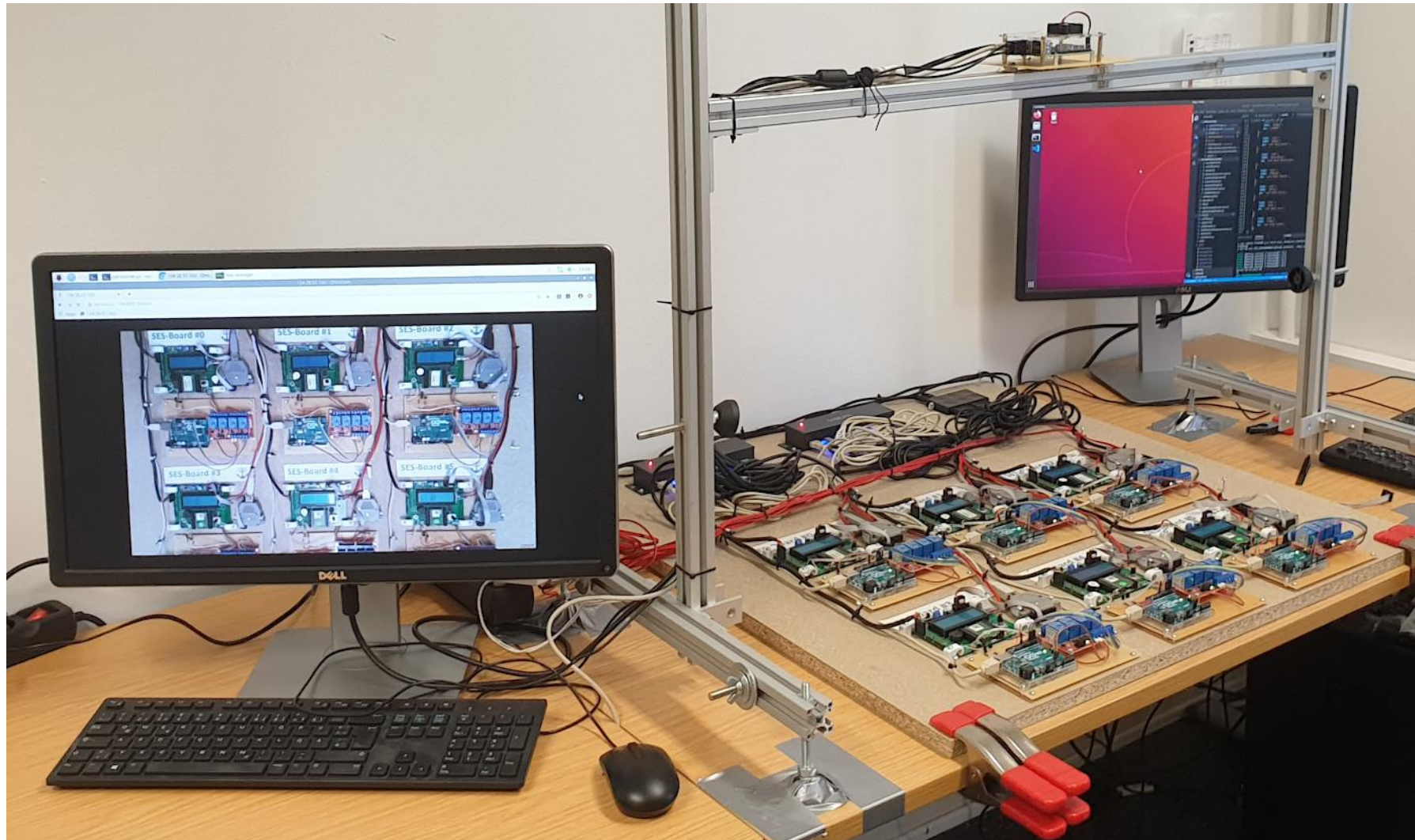




# SES Board (Schematic)



# Area Q1.006 (restricted area)



# Remote Access

Software for Embedded Systems - Remote Builder
You are logged in as christian
You have board 1

### Upload hex file

Pick your .hex file

Upload Browse

### Control Environment


Light Intensity: 42% 42

Joystick Press

Rotary Encoder Left Middle Right

Reset MCU

### See the Boards



### AVR Dude output

```

avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.01s

avrdude: Device signature = 0x1ea701 (probably m128rfa1)
avrdude: erasing chip
avrdude: reading input file "temp/ttyFlash1.hex"
avrdude: input file temp/ttyFlash1.hex auto detected as Intel Hex
avrdude: writing flash (1352 bytes):

Writing | ##### | 100% 0.65s

avrdude: 1352 bytes of flash written
avrdude: verifying flash memory against temp/ttyFlash1.hex:
avrdude: load data flash data from input file temp/ttyFlash1.hex:
avrdude: input file temp/ttyFlash1.hex auto detected as Intel Hex
avrdude: input file temp/ttyFlash1.hex contains 1352 bytes
avrdude: reading on-chip flash data:

```

### Serial Output

```

button2
<left
<left
<left
>right
>right
button

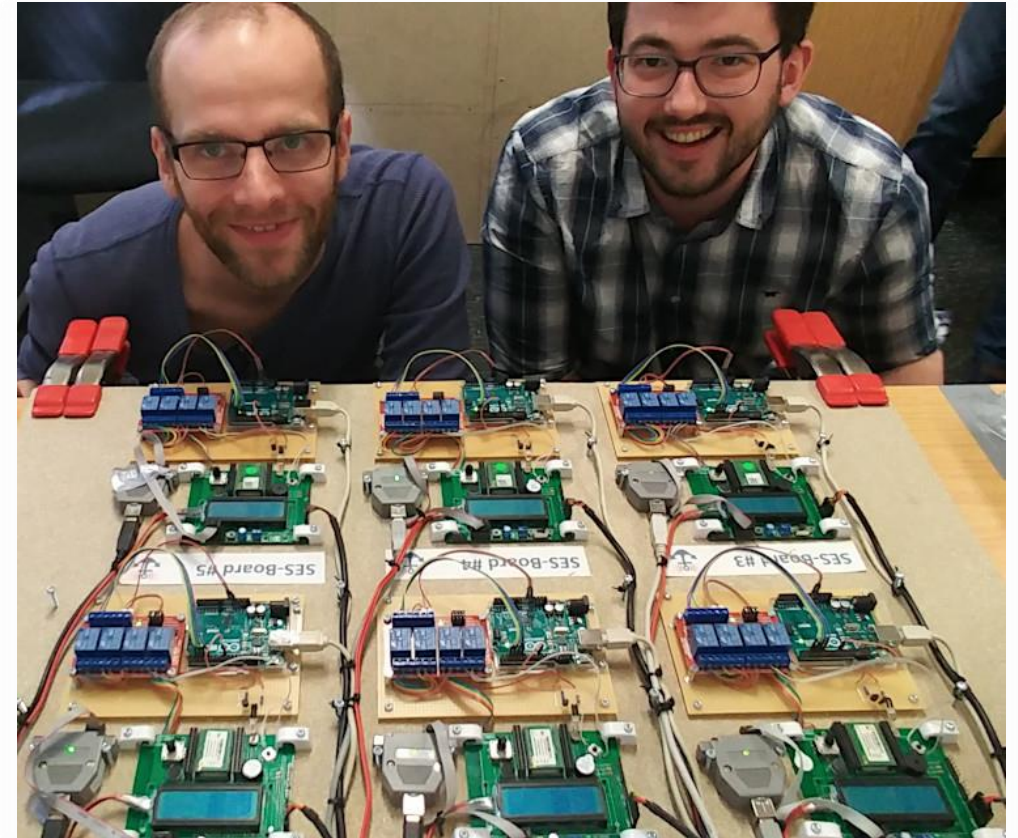
```

Pause Clear



# Lab – Practical Work

- 72 places (FCFS)
- Work in teams of 2 students
- Remote Access & Git usage
- Strict requirements for entry
  - Toolchain installed until 2<sup>nd</sup> lab (see exercise sheet)
  - StudIP registration (for groups)
  - Git registration
- Programming the SES Board  
(Atmel ATmega128RFA + sensors)



# Lab – Remote Access

- `seslab.smp.tuhh.de` (via TUHH VPN)
- Board access
  - Reservation system
  - Random Access
- Restrictions
  - During Q&A sessions
  - Maintenance
- Report issues and problems via StudIP SES forum

# Lab – Q&A Sessions

- Tuesday, 13:15 – 17:30
- 6x 30-minute slots
- **First session: 2020-04-28**
- Registration in StudIP
  - Opens today, 20:00
  - Closes tomorrow, 12:00 (April 24)
- Lab info on Friday afternoon
  - **Prerequisite:** Git login (`collaborating.tuhh.de`)
- Responsible:
  - Fabian Steinmetz, Peter Oppermann, Christian Busse





# Software for Embedded Systems - Online

- Web-Site of lecture at Stud.IP:
  - `https://e-learning.tuhh.de`
  - Updates and Notifications
  - Copies of slides
  - Instructions for the lab
  - Dates, Links
  - ...

# First Exercise

- Read the tutorial (StudIP)
- Read and complete the checklist (StudIP) *before* the first lab!
- Install and set up the environment as described

# Performance Record

- 6 ECTS
- Written Exam
- Lab counts 30% for final grade
- 3 selected exercises have to be submitted
- Bonus points for a challenging task per graded exercise
- Details to be announced

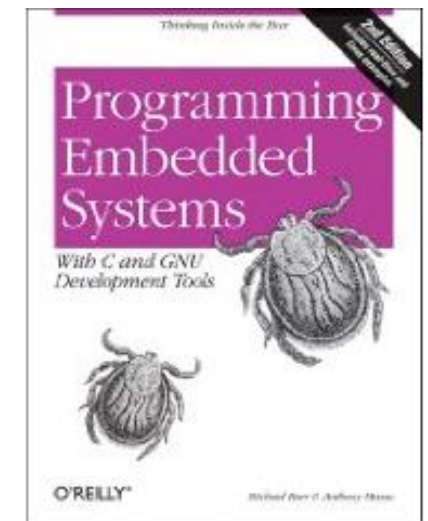
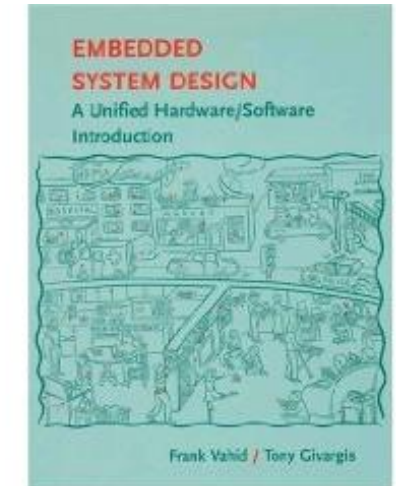




# References

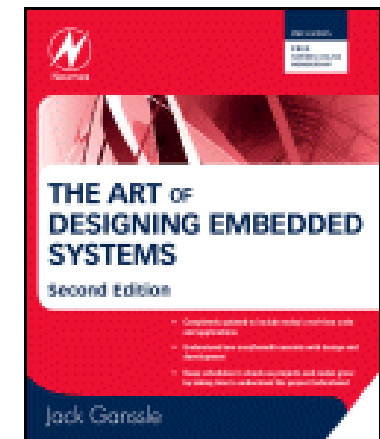
# Literature

- **Embedded System Design**  
F. Vahid and T. Givargis  
John Wiley & Sons, 2002  
ISBN: 9780471386780
- **Programming Embedded Systems: With C and Gnu Development Tools**  
M. Barr and A. Massa  
O'Reilly, 2nd ed. 2006  
ISBN: 978-0596009830



# Literature

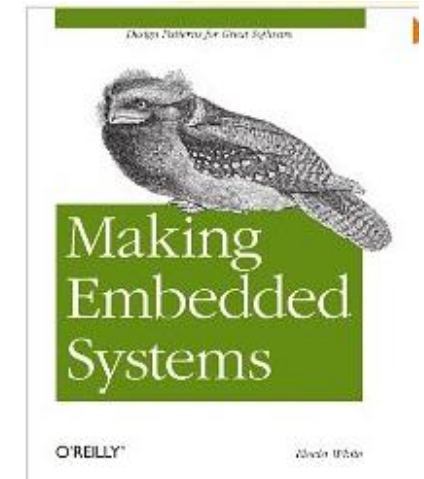
- **C und C++ für Embedded Systems**  
F. Bollow, M. Homann, K. Köhn  
MITP, 2009  
ISBN: 9783826659492
- **The Art of Designing Embedded Systems**  
J. Ganssle  
Newnses, 2nd ed. 2008  
ISBN: 978-0750686440





# Literature

- **Mikrocomputertechnik mit Controllern der Atmel AVR-RISC-Familie**  
G. Schmitt  
Oldenbourg, 2010  
ISBN: 3486589881
- **Making Embedded Systems: Design Patterns for Great Software**  
E. White  
O'Reilly, 2011  
ISBN: 1449302149





# Software for Embedded Systems

Prof. Dr.-Ing. Bernd-Christian Renner

