

# 64-210 Eingebettete Systeme

[http://tams.informatik.uni-hamburg.de/  
lectures/2013ss/vorlesung/es](http://tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es)

Norman Hendrich / Andreas Mäder



Universität Hamburg  
Fakultät für Mathematik, Informatik und Naturwissenschaften  
Fachbereich Informatik

**Technische Aspekte Multimodaler Systeme**

Sommersemester 2013

# Info zur Organisation

## 1. Organisatorisches

Vorlesung

Übungen

Allgemein

## 2. Literatur



# Vorlesung

[tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es](https://tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es)

- ▶ Mi. 10:15-11:45 F-132  
Do. 10:15-11:45 F-132
- ▶ Vorlesung folgt: „*Embedded System Design*“ von P. Marwedel
- ▶ Originalmaterial: `ls12-www.cs.tu-dortmund.de/daes/de/daes/mitarbeiter/prof-dr-peter-marwedel/embedded-system-text-book/slides/slides-2012.html`
- ▶ diverse gute Lehrbücher — Empfehlungen s.u.
- ▶ Informationen und Downloads auf der Webseite — **aktuell!**
- ▶ eingestreute Hinweise auf aktuelle Themen und Vertiefung

# Übungen

[tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es/uebungen](https://tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es/uebungen)

## KVV

- ▶ zwei Gruppen, jeweils im Anschluß an die Vorlesung
- ▶ Mi. 12:15-13:45 F-132
- Do. 12:15-13:45 F-132

## Planung

- ▶ weniger Übungszettel... mehr Praxis
- ⇒ 6-8 Übungen als Block (mehrstündig)
- ▶ Termin: nach Absprache
- ▶ Vorbereitung (zu Hause) + praktische Arbeit mit Hardware

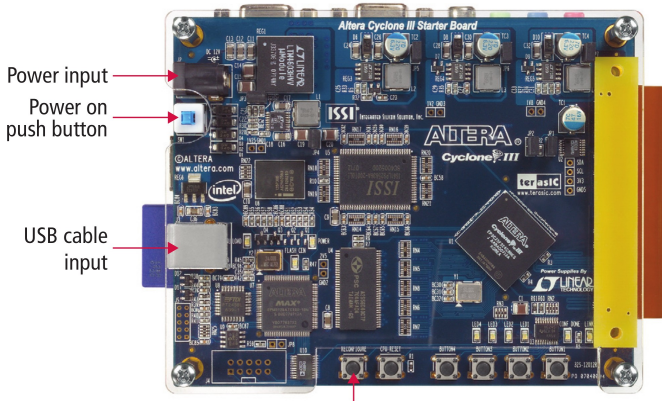
# Übungen: Hardwareplattformen

## $\mu$ Controller System

- ▶ Arduino arduino.cc  
Teensy www.pjrc.com/teensy
- ▶ typische Prototypenboards mit frei belegbaren Ein-/Ausgängen  
Prozessor: ATmega / ARM
- ▶ Aufgaben
  - ▶ Interruptbehandlung
  - ▶ einfache Schnittstellen (seriell, I<sup>2</sup>C...)
  - ▶ Kontrollaufgaben (Steuerung durch endlichen Automaten)
  - ▶ ...

# Übungen: Hardwareplattformen

## FPGA-Prototypensystem



Press RECONFIGURE to return to Application Selector

# Übungen: Hardwareplattformen (cont.)

## FPGA-Prototypensystem

### ▶ programmierbare Hardware: Altera FPGA

[www.altera.com](http://www.altera.com)

- ▶ Cyclone III EP3C25F324
- ▶ 24 624 LEs  $\approx$  290 000 Gatter
- ▶ 594 Kbit (interner) Speicher
- ▶ 66 Hardwaremultiplizierer:  $18 \times 18$  bit
- ▶ 4 PLLs

### ▶ Speicher

- ▶ 1MB SRAM
- ▶ 32MB DDR SDRAM
- ▶ 16MB Flash

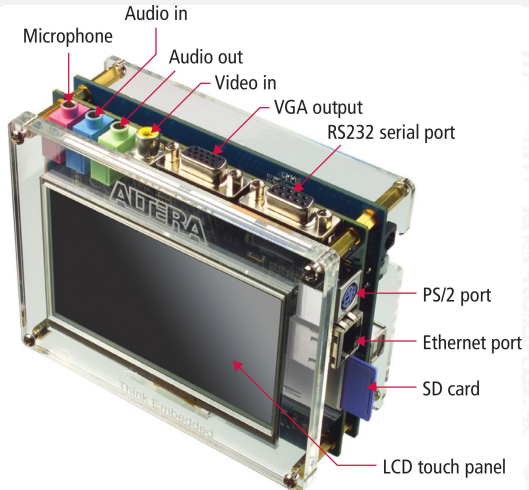
### ▶ Ein-/Ausgabe

- ▶ 4 Taster
- ▶ 4 LEDs

# Übungen: Hardwareplattformen (cont.)

## FPGA-Prototypensystem

### I/O-Erweiterungsplatine





# Übungen: Hardwareplattformen (cont.)

## FPGA-Prototypensystem

- ▶ LCD Touch-Screen,  $800 \times 400$
- ▶ VGA Ausgang
- ▶ Composite TV-Eingang
- ▶ 24-bit Audio I/O
- ▶ 10/100 Mbit Ethernet, PHY+MAC
- ▶ serieller Anschluss
- ▶ SD Karte

# Übungen: Hardwareplattformen (cont.)

## FPGA-Prototypensystem

### Implementationsmöglichkeiten

#### 1. Softwareentwurf

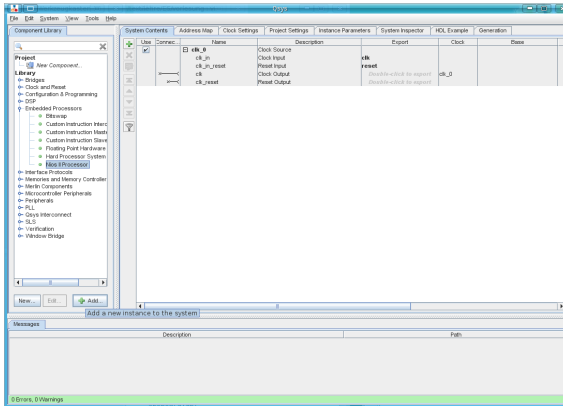
- ▶ IP-Komponenten (*Intellectual Property*)  
Speicher, Busse, I/O-Schnittstellen. . .
- ▶ 32-bit Prozessor (NIOS II)
- ▶ Softwareentwicklung: C Programmierung  
Eclipse + Cross-Compiler

#### 2. Hardwareentwurf: Hardwarebeschreibungssprachen hier: VHDL

#### 3. gemischt: Hardware + Software + IP-Komponenten

# Übungen: Hardwareplattformen (cont.)

## FPGA-Prototypensystem

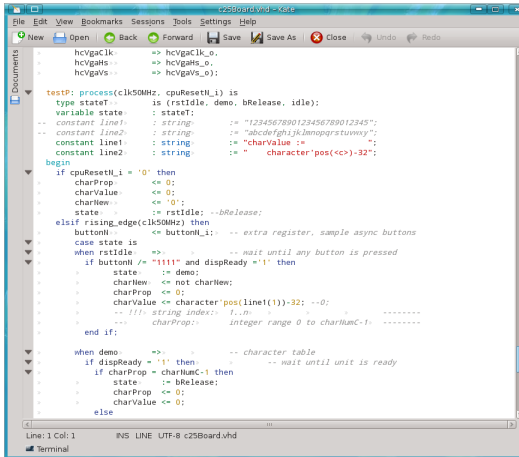


IP-Komponenten



# Übungen: Hardwareplattformen (cont.)

## FPGA-Prototypensystem



```

c25Board.vhd - K&ste
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
hcvgaClk_o => hcvgaClk_o,
hcvgaHS_o => hcvgaHS_o,
hcvgaVS_o => hcvgaVS_o;

testP: process(clk50MHz, cpuResetN_i) is
type stateT is (rstIdle, demo, bRelease, idle);
variable state : stateT;
-- constant line1 : string := "1234567890123456789012345";
-- constant line2 : string := "abcdefghijklmnopqrstuvwxyz";
constant line1 : string := "charValue := ";
constant line2 : string := "character'pos(<>)-32";
begin
if cpuResetN_i = '0' then
charProp <= 0;
charValue <= 0;
charNew <= '0';
state := rstIdle; --bRelease;
elsif rising_edge(clk50MHz) then
buttonN_i <= buttonN_i; -- extra register, sample async buttons
case state is
when rstIdle => -- wait until any button is pressed
if buttonN_i /= "1111" and dispReady = '1' then
state := demo;
charNew <= not charNew;
charProp <= 0;
charValue <= character'pos(line1(1))-32; --0;
-- !!! string index: 1..n
-- charProp: integer range 0 to charNumC-1
end if;
when demo => -- character table
if dispReady = '1' then -- wait until unit is ready
if charProp = charNumC-1 then
state := bRelease;
charProp <= 0;
charValue <= 0;
else

```

VHDL-Code

## Feedback erwünscht

- ▶ Zwischenfragen: bitte Feedback bei Unklarheiten etc.!
- ▶ Fehler und Ungenauigkeiten in den Folien und Materialien bitte melden
- ▶ Vorschläge und Hinweise auf Tools, schöne Lehrmaterialien etc. sind immer willkommen!

Problem: unterschiedliches Vorwissen!

- ▶ Voraussetzungen: insbesondere **Rechnerstrukturen**  
aber auch: Assemblerprogrammierung, Betriebssysteme, Netzwerke. . .
- ▶ generell: Interesse an Hardware

# Kontakt

Dr. Norman Hendrich

[hendrich@informatik.uni-hamburg.de](mailto:hendrich@informatik.uni-hamburg.de)

+49 40 42883 2399

Informatikum, Haus F-314

Dr. Andreas Mäder

[maeder@informatik.uni-hamburg.de](mailto:maeder@informatik.uni-hamburg.de)

+49 40 42883 2502

F-317

# Literatur: empfohlene Lehrbücher

[Mar11] P. Marwedel:

*Embedded System Design –  
Embedded Systems Foundations of Cyber-Physical Systems.*  
2nd edition. Springer-Verlag, 2011. ISBN 978–94–007–0256–1

**Primärliteratur:** Komplette Behandlung des Themas.

Buch und Foliensatz sind Vorlage dieser Vorlesung. Originalmaterial der Vorlesung von Prof. Marwedel:  
ls12-[www.cs.tu-dortmund.de/daes/de/daes/mitarbeiter/prof-dr-peter-marwedel/  
embedded-system-text-book/slides/slides-2012.html](http://www.cs.tu-dortmund.de/daes/de/daes/mitarbeiter/prof-dr-peter-marwedel/embedded-system-text-book/slides/slides-2012.html)

Auch als eBook in der Informatik-Bibliothek vorhanden.



## Literatur: empfohlene Lehrbücher (cont.)

[TH07] J. Teich, C. Haubelt:

*Digitale Hardware/Software-Systeme –  
Synthese und Optimierung.*

2. Auflage. Springer-Verlag, 2007. ISBN 978-3-540-46822-6

Schwerpunkte: Realisierung eingebetteter Systeme, also Synthese, Scheduling, Bindung von Ressourcen etc.

Auch als eBook in der Informatik-Bibliothek vorhanden.

[HT10] C. Haubelt, J. Teich:

*Digitale Hardware/Software-Systeme –  
Spezifikation und Verifikation.*

Springer-Verlag, 2010. ISBN 978-3-642-05355-9

Schwerpunkte: Methoden zur Spezifikation eingebetteter Systeme und der (formalen) Verifikation, Simulation etc.

Auch als eBook in der Informatik-Bibliothek vorhanden.

## Literatur: empfohlene Lehrbücher (cont.)

[VG02] F. Vahid, T. Givargis:

*Embedded System Design –*

*A unified hardware/software introduction.*

John Wiley & Sons, 2002. ISBN 978-0-471-38678-0

Alle Themen im Überblick, etwas älter.

Mehrere Exemplare in der Informatik-Bibliothek vorhanden.

## Literatur: weitere Lehrbücher

[Brä08] T. Bräunl:

*Embedded Robotics – Mobile Robot Design and Applications with Embedded Systems.*

3rd edition. Springer-Verlag, 2008. ISBN 978–3–540–70534–5

Eigentlich ein „Robotik“-Buch, dient aber als Beispiel, dass die Robotik auf „Eingebetteten Systemen“ aufbaut (Teil 1)

Auch als eBook in der Informatik-Bibliothek vorhanden.

[Lew13] D. Lewis:

*Fundamentals of embedded software – with the ARM Cortex-M3.*

2nd edition. Pearson Education, 2013.

ISBN 978–0–13–291654–7

Schwerpunkt: Software, viel praktische Programmierung in C und Assembler.

## Literatur: weitere Lehrbücher (cont.)

[SC12] J. Sanchez, M.P. Canton:

*Embedded systems circuits and programming.*

CRC Press, 2012. ISBN 978-1-439-87904-7

Schwerpunkt auf PIC  $\mu$ Controller, enthält außerdem viele (elektro-) technische Grundlagen.

[Wol12] M. Wolf:

*Computers as components –*

*Principles of embedded computing system design.*

3rd revised ed. Morgan Kaufmann Publishers Inc., 2012.

ISBN 978-0-12-388436-7

[Zöb08] D. Zöbel:

*Echtzeitsysteme – Grundlagen der Planung.*

Springer-Verlag, 2008. ISBN 978-3-540-76395-6

## Literatur: Vertiefung

[BO11] R.E. Bryant, D.R. O'Hallaron:

*Computer systems – A programmers perspective.*

2nd edition. Pearson, 2011. ISBN 978-0-13-713336-9

Rechnerarchitektur mit Schwerpunkt Software und Systeme, leider nicht ganz billig. Viele C-Programme und Systemprogrammierung. Beispiele anhand Intel x86 Architektur.

[TT13] A.S. Tanenbaum, A. Todd:

*Structured Computer Organization.*

6th edition. Pearson International, 2013.

ISBN 978-0-273-76924-8

Guter Überblick zum Thema Rechnerarchitektur, klares didaktisches Konzept. Java VM, Intel x86, SPARC. Mit jeder Auflage komplett überarbeitet und aktualisiert.

## Literatur: Vertiefung (cont.)

[PH12] D.A. Patterson, J.L. Hennessy:

*Computer Organization and Design –  
The Hardware/Software Interface.*

4th revised ed. Morgan Kaufmann Publishers Inc., 2012.

ISBN 978-0-12-374750-1

Schönes Lehrbuch von den Entwicklern der RISC/MIPS Prozessoren.

[HP11] J.L. Hennessy, D.A. Patterson:

*Computer architecture – A quantitative approach.*

5th edition. Morgan Kaufmann Publishers Inc., 2011.

ISBN 978-0-12-383872-8

Die Bibel zum Thema Rechnerarchitektur.