

HotSys Praktikum

Vorbesprechung

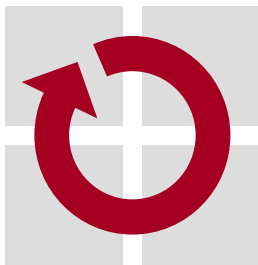
27.10.2010

Peter Ulbrich

Martin Hoffmann

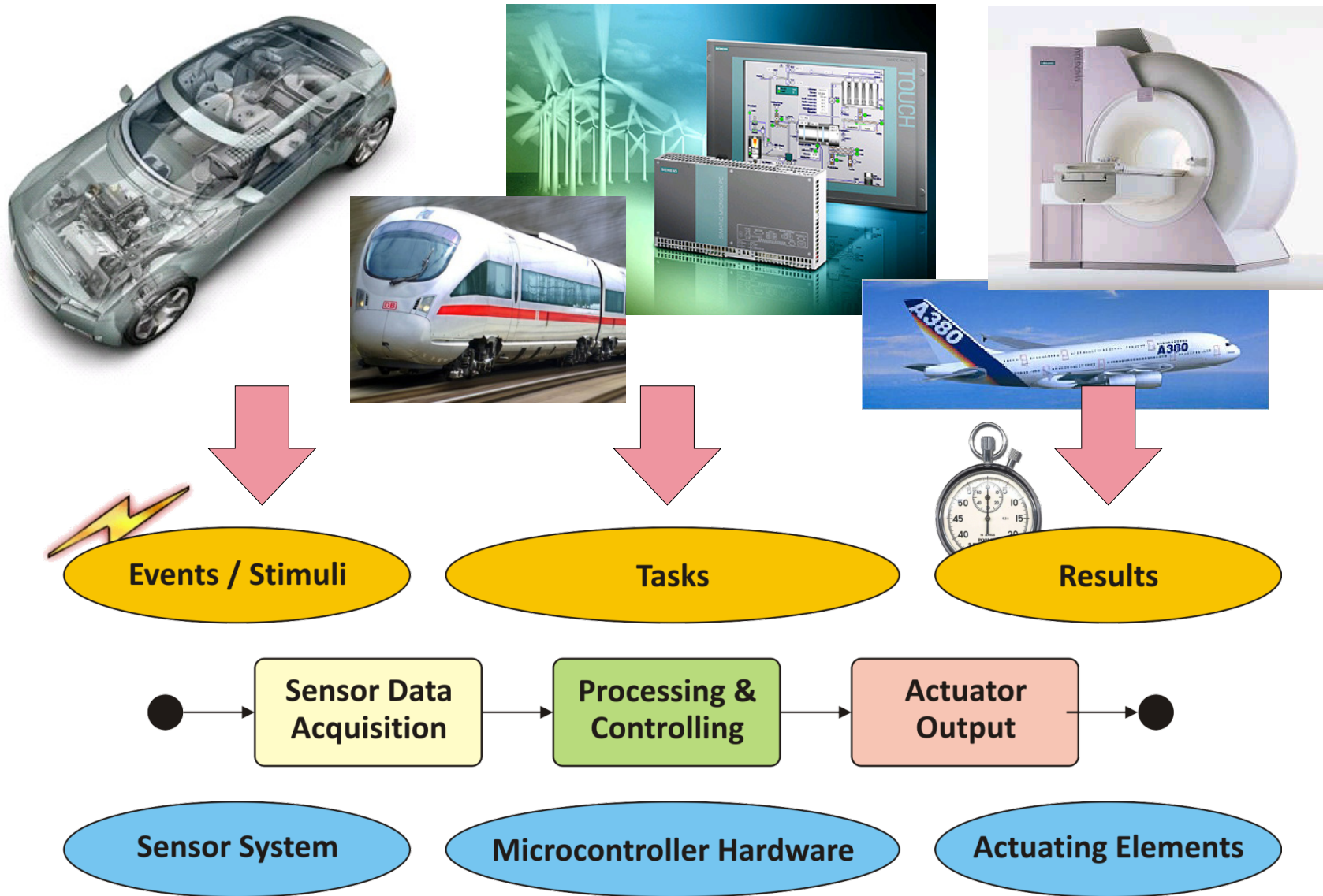
Jürgen Kleinöder

<http://www4.cs.fau.de/~{ulbrich, hoffmann, jklein}>
{ulbrich, hoffmann, jklein}@cs.fau.de

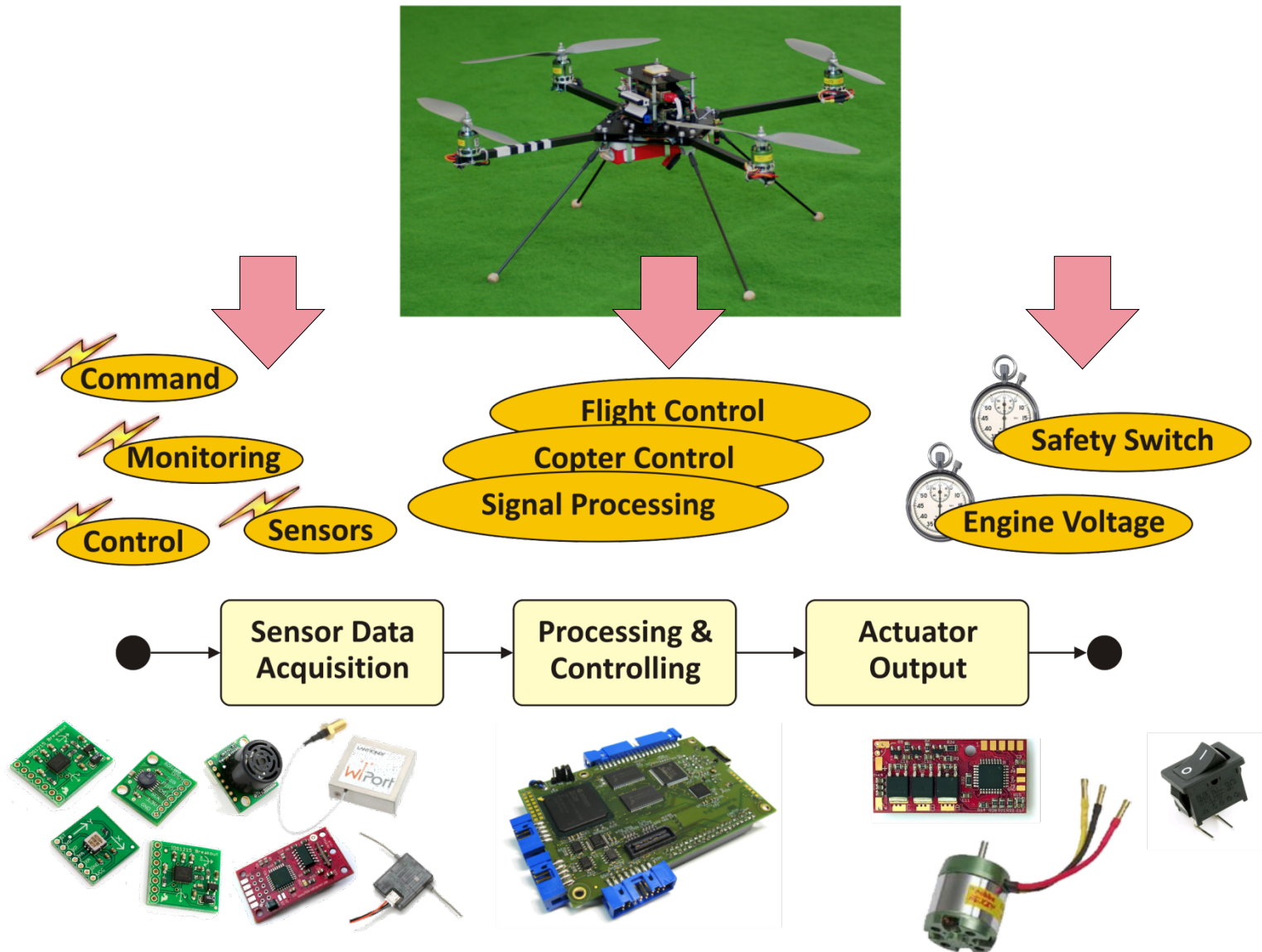


Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme
Friedrich-Alexander-Universität Erlangen-Nürnberg

Sicherheitskritische Systeme



I4Copter – Ein sicherheitskritisches System



I4Copter Timeline (1)

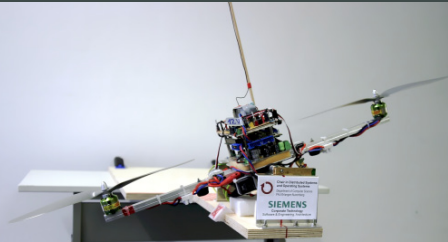
Late 2007

- A bagful of hardware
- First clumsy copter
- Incapable of flying



Early 2008

- Back to drawing-board
- 1-axis test rig
- Engine test rig



Mid 2008

- I4Copter Prototype v1
- First flight (Late 2008)
- I4Copter Protoype v1.1



Early 2009

- I4Copter Prototype v2
- Acceptable flight behaviour
- I4Copter → SCR



I4Copter Timeline (2)

Early 2009

- Multi-layer sensor periphery board
- Pressure sensor
- Software fully based on CoSa design



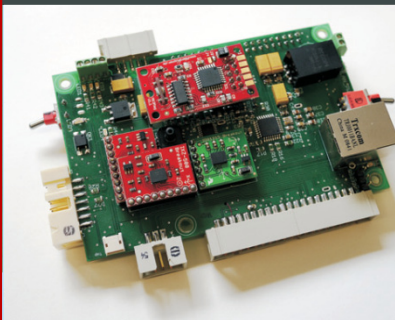
Mid 2009

- All new attitude controller
- Detail improvements



Late 2009

- Significantly improved sensor board
- TMR gyroscopes
- Digital radio control



Early 2010

- CoRed Software Redundancy
- Power electronics board
- I4Copter → SCR



„Systemsoftware für sicherheitskritische Systeme“



■ Forschungs- und praxisorientiert

- Im Kontext aktueller Forschungsprojekte: I4Copter, Keso, DanceOS
- Anhand konkreter Systeme: z.B. I4Copter

■ Lehrziele

- System- und hardwarenahe Softwareentwicklung in C++ und Java
- Mikrocontroller- und Peripherie-Programmierung
- Einsatz von Analyse- und Debugging-Werkzeugen
- Umgang mit Echtzeiteigenschaften in der Praxis
- Methoden und Techniken für *Software-based Dependability*



Lehrveranstaltungskonzept (1)

■ Bausteine der Veranstaltung:

Semester		Vorlesungsfrei
Wöchentliche Treffen mit Seminarcharakter	(2,5 ECTS)	Blockpraktikum (2 Wochen) (2,5 ECTS)
Bearbeitung von Übungsaufgaben	(2,5 ECTS)	
Besuch einer Vertiefungsveranstaltung	(2,5 ECTS)	

■ Seminar und Übungen (5 ECTS)

- **Propädeutikum**: Einführung in die Grundlagen
- **Übungsaufgaben**: Kennenlernen von Techniken und Werkzeugen
- **Seminar**: Vorstellen der Übungen und Diskussion
- Bearbeitung in kleinen Gruppen (2-3 Personen)

→ **Was und womit bringe ich etwas zum „fliegen“?**



Seminarthemen

■ Vorläufiger Themenüberblick (Seminar / Übungen)

1	Entwicklung eingebetteter Systeme in C/C++	PU
2	Peripherie und Debugging	PU
3	Entwicklung von Echtzeitanwendungen in eCos	MH
4	I4Copter: Architektur und Aufbau	PU
5	Aspektorientierte Programmierung	DL/IT
6	Konfiguration mit Kconfig / eCos CDL	RT
7	Embedded Java mit KESO	MS
8	Software-basierte Redundanz für sicherheitskritische Systeme	MH



Martin Hoffmann



Daniel Lohmann



Michael Stilkerich



Reinhard Tartler



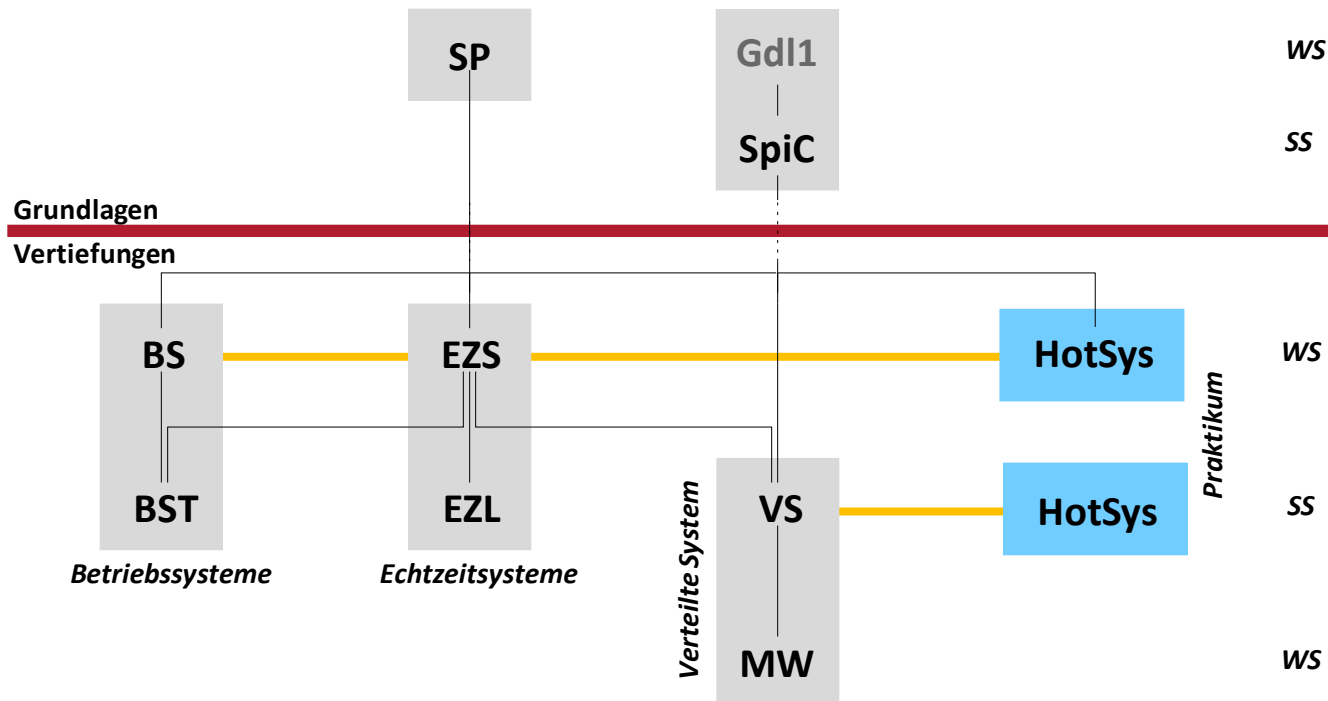
Isabella Thomm



Peter Ulbrich



Lehrveranstaltungs-konzept (2)



■ Besuch einer Vertiefungsveranstaltung (2,5 ECTS)

- **Detailwissen:** Durch eine Vertiefung aus Lehre@I4
- Im WS: BS oder EZS (2,5 ECTS aus Vorlesung, Übung, **erweiterte Übung**)

→ **Wie bringe ich etwas zum „fliegen“?**



Lehrveranstaltungs-konzept (3)

■ Blockpraktikum (ca. 2,5 ECTS)

- Im Anschluss an das Semester → 2 Wochen intensives Blockpraktikum
- Konkretes (Implementierungs-) Projekt
- 2-3 Gruppen

■ Themenübersicht

Projekt 1:
I4Copter Anwendung

Projekt 2:
eCos Multicore

Projekt 3:
Java for I4Copter

Projekt 4:
Software Redundanz

→ ***Wir bringen es zum „fliegen“?***



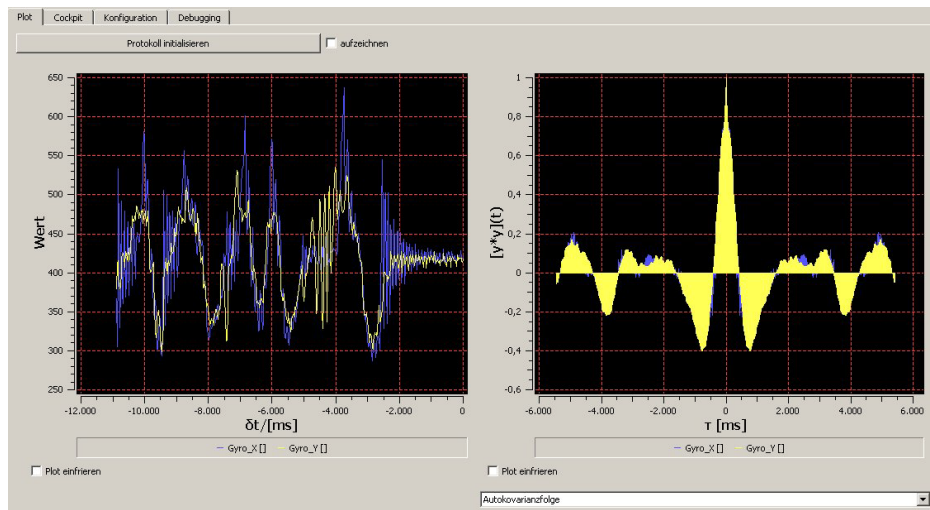
I4Copter Anwendung

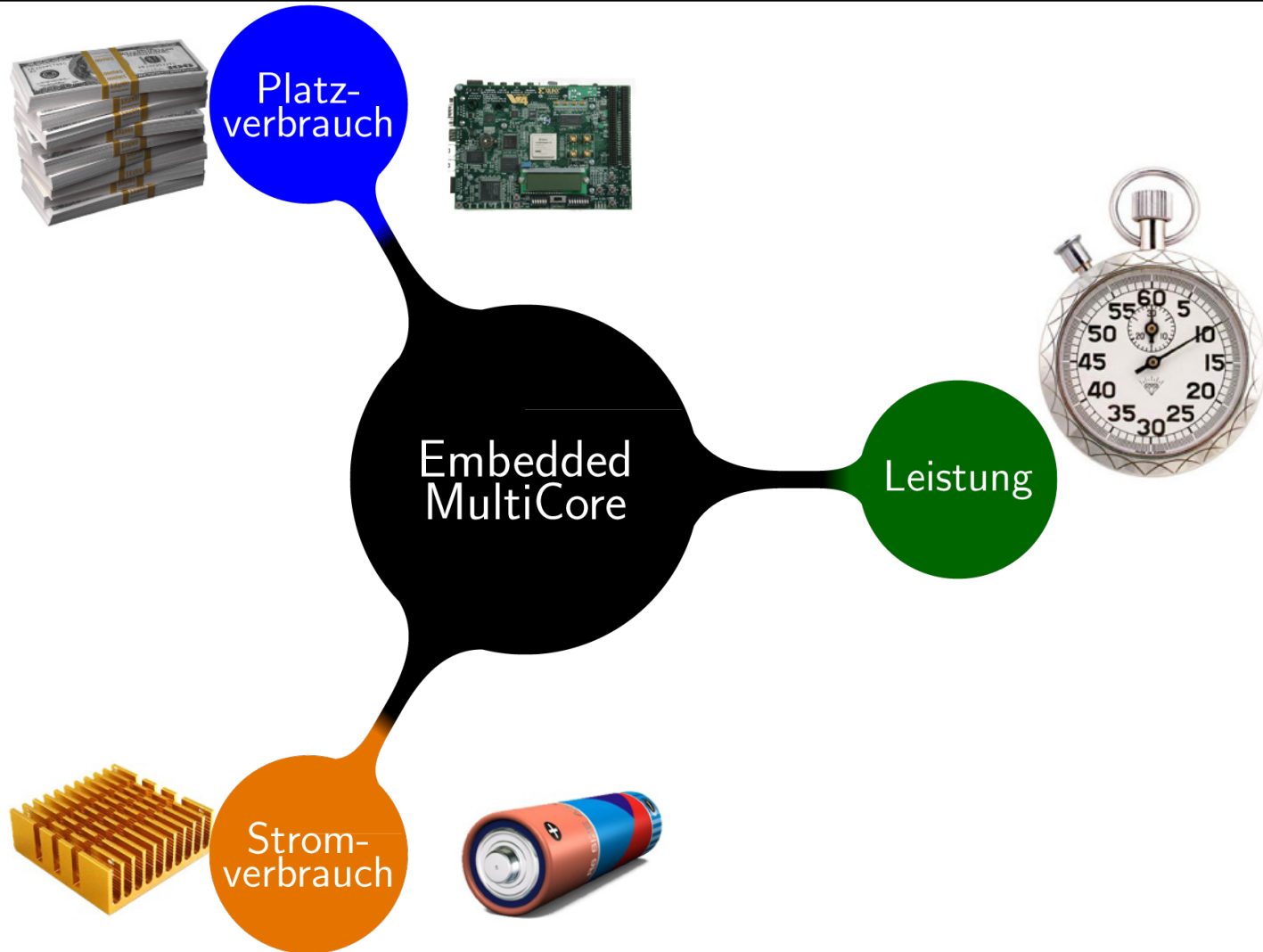
■ Neuentwicklung

- Vollkompensierte (Roll/Pitch) Kamerasteuerung
- Verhaltenssteuerung für autonomen Flug
- Kollisionsvermeidung durch Nahbereichsüberwachung

■ Verfeinerung bestehender Anwendungen

- Wegpunktnavigation mittels GPS
- Sensorsystem für Lage und Höhe





- Professionell eingesetztes eingebettetes Echtzeit-BS
- Stark konfigurierbar für verschiedene Plattformen
- Aktuell nur grundlegende SMP Unterstützung (x86)



Praktikumsziel: Erweiterung von eCos

- MultiCore Unterstützung (→ MPStuBS)
- Feingranular konfigurierbar (→ AspectC++)
- Evtl. Soft-Router Implementierung

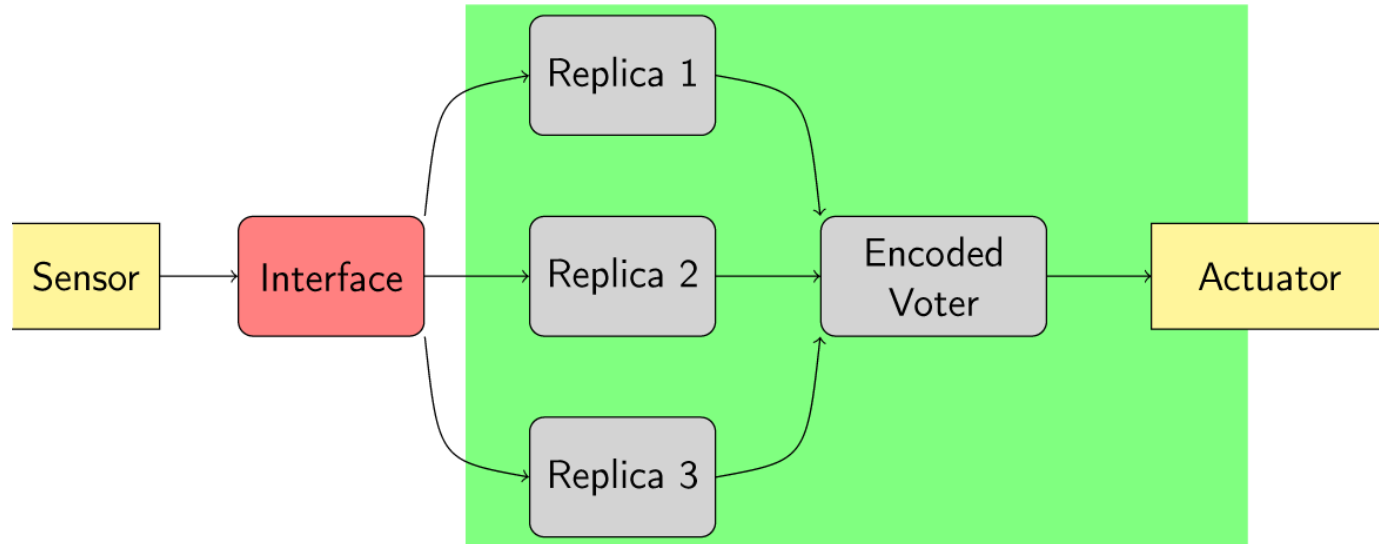
Java-Implementierung einer I4Copter-Komponente

- I4Copter-Anwendung in Komponenten untergliedert
- Datenaustausch über saubere Schnittstellen
 - einzelne Komponenten können ausgetauscht werden
 - mit Java-zu-C-Übersetzer KESO auch in Java
- Fluglageregelung ist bereits portiert und kann als Vorlage dienen
- Einbringung des erworbenen Toolwissens
 - Vergleich der eigenen Implementierung mit der Originalvariante bzgl.
 - Laufzeit
 - Codegröße
 - RAM-Verbrauch



Software-basierte Redundanz (I)

- Moderne Hardware anfällig für „transiente“ Fehler
- Effiziente Lösung: Software-basierte Redundanz



Aktuelle Umsetzung im I4Copter:

- Dreifach redundante Ausführung
- Kodierter Mehrheitsentscheid

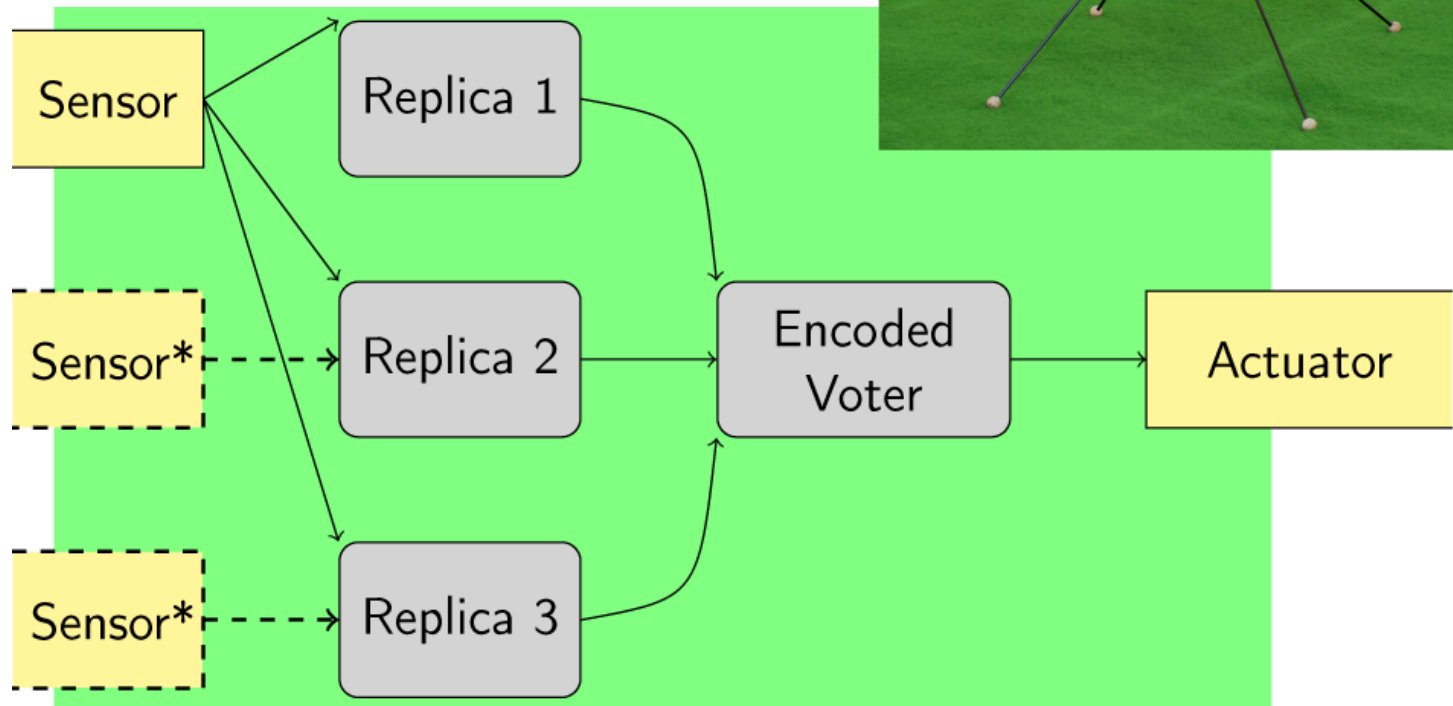
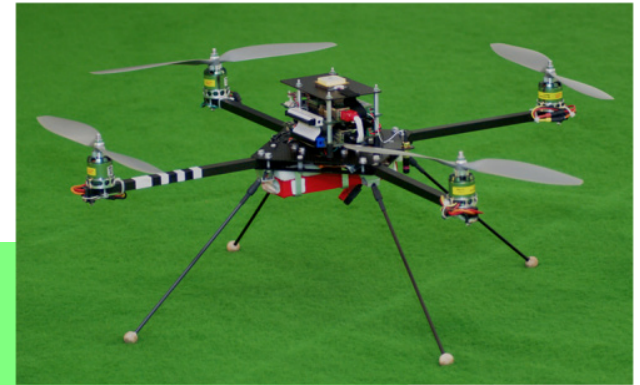
→ Sensordatenerfassung noch nicht abgesichert!



Software-basierte Redundanz (II)

Praktikumsziele: Erweiterung des Redundanzbereichs

- Datendiversität am Eingang
- Toleranzbereichsvoting
- Copter **sicher(er)** fliegen lassen



Organisation

■ Materialien

- www4.informatik.uni-erlangen.de/Lehre/WS10/P_HOTSYS
- kein Skript – Folien zu Vorlesung/Übung

■ Termine

- Seminar: Mittwoch 10 Uhr c.t., Raum 0.035
- Laborarbeit zur freien Einteilung (im Umfang von 2,5 ECTS)
- Vorlesung/Übung/Erweiterte Übung aus einem Vertiefungsfach
- Blockpraktikum 1-3 Wochen nach Semesterende (nach Vereinbarung)

■ Prüfungsmöglichkeiten

- **Präsentation** der Übungen und des Projekts
 - **Vertiefung** in der jeweiligen Veranstaltung – alternativ Kolloquium
- ➔ **Benoteter Schein**



Ende

ENDE

Fragen?

