Real-Time & Embedded Systems

Uwe R. Zimmer - The Australian National University

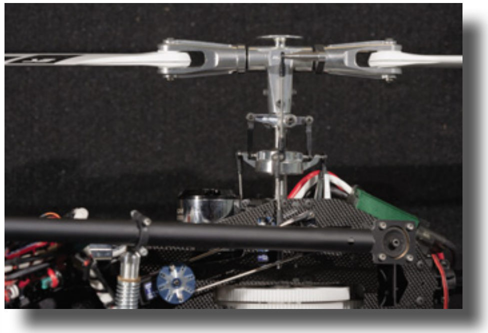
Real-Time & Embedded Systems 2019

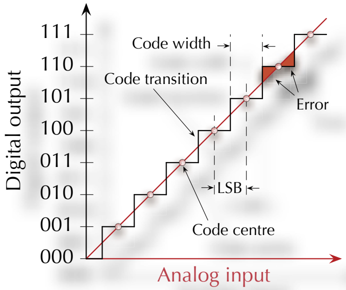
Organization & ToC

Uwe R. Zimmer - The Australian National University

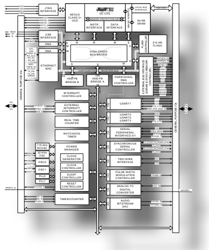
Organization & ToC

what is offered here?

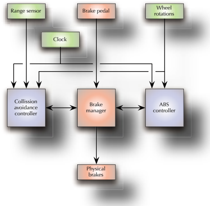




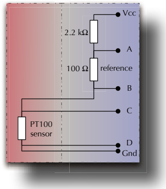
Overview, Perspectives, Paths, Methods, Implementations

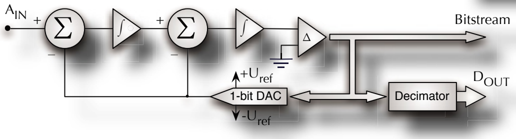


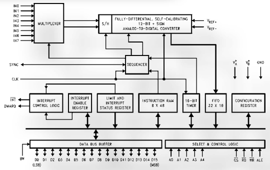
and open questions

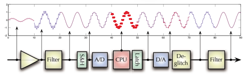


of/into/for/about

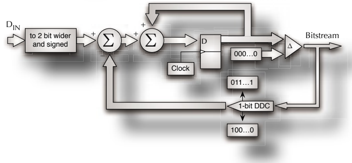


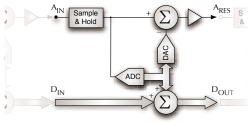


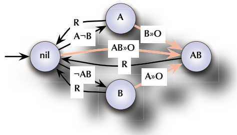


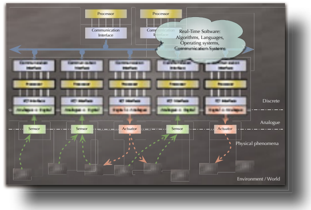


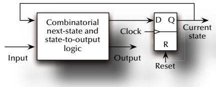
Real-Time & Embedded Systems

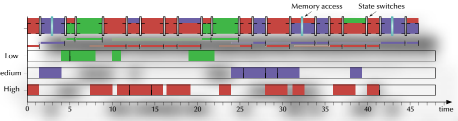


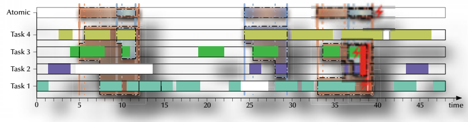












Organization & ToC

who could be interested in this?

anybody who …

… *would like to see* **immediate real-world involvement** *in his/her work.*

… *would like to learn how to create*
predictable and fault-tolerant, complex systems.

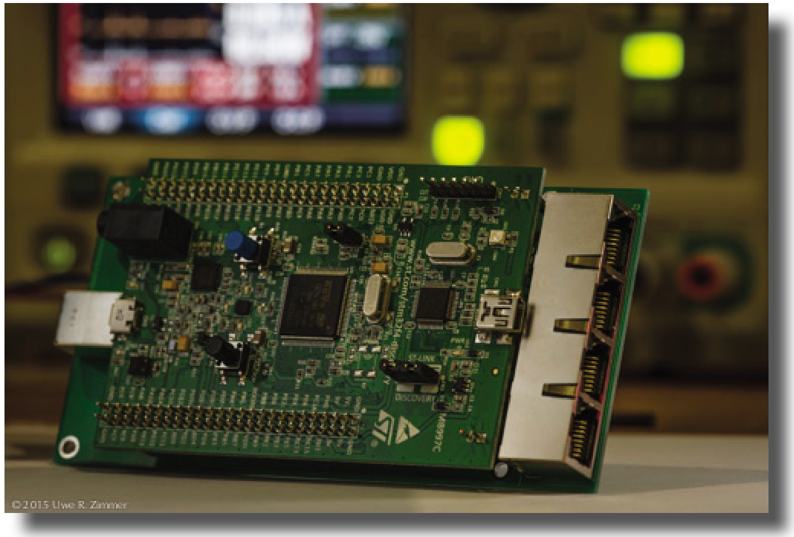
… *would like to know more about the usage of >95% of all processors.*

Organization & ToC

who are these people? – introduction

This course will be given by

Uwe R. Zimmer



Tutoring and labs by

Calum Snowdon &
Michael Bennett

Electronics design by

Mark Turner

Organization & ToC

***how* will this all be done?**

☞ Lectures:

- 2 x 1.5 h lectures per week … all the nice stuff
  Monday, 15:00 (Engineering Theatre) and Thursday 09:00 (Forestry Theatre)

☞ Laboratories:

- 2 hours per week … all the rough stuff
  time slots: on our web-site – all in CSIT laboratories
  -enrolment: https://cs.anu.edu.au/streams/

☞ Resources:

- Introduced in the lectures and collected on the course page:
  https://cs.anu.edu.au/courses/comp4330/ … as well as schedules,
  slides, sources, link to forums, etc. pp. … keep an eye on this page!

☞ Assessment:

- Exam at the end of the course (70%) plus one assignments (30%)
  – both are tested in oral exams (unless enrolment numbers require otherwise).

Organization & ToC

Textbooks (sort of …)

[Burns2009]
Alan Burns and Andy Wellings
Real-Time Systems and Programming Languages
Addison Wesley, fourth edition, 2009

[Burns2007]
Alan Burns & Andy Wellings
Concurrent and Real-Time Programming in Ada
Cambridge University Press, 2007

[McCormick11]
McCormick, J. W., Singhoff, F., & Hugues, J.
Building Parallel, Embedded, and Real-Time Applications with Ada.
Cambridge University Press, 2011.

… plus specific references for each topic (all on the course site).

Organization & ToC

Topics

1. *Introduction & Real-time languages*
2. *Physical coupling*
3. *Interfaces*
4. *Time & Embodiment*
5. *Asynchronism*
6. *Synchronisation*
7. *Scheduling*
8. *Resource control*
9. *Reliability & Fault-tolerance*

Organization & ToC

Topics

1. *Introduction & Real-time languages*

1.1. Staking out the field

1.2. Features (and non-features) of a real-time system

1.3. Components of a real-time system

1.4. Real-time languages

- Ada
- Esterel
- Pearl
- VHDL
- Timed CSP
- Real-time JAVA
- POSIX

2. *Physical coupling*
3. *Interfaces*
4. *Time & Embodiment*
5. *Asynchronism*
6. *Synchronisation*
7. *Scheduling*
8. *Resource control*
9. *Reliability & Fault-tolerance*

Organization & ToC

Topics

1. *Introduction & Real-time languages*
2. *Physical coupling*

2.1. Physical values

2.2. Introduction to sensors

2.3. Frequently employed sensors

3. *Interfaces*
4. *Time & Embodiment*
5. *Asynchronism*
6. *Synchronisation*
7. *Scheduling*
8. *Resource control*
9. *Reliability & Fault-tolerance*

Organization & ToC

Topics

1. Introduction & Real-time languages
2. Physical coupling
3. Interfaces

3.1. Analogue signal chain in a digital system

3.2. Analog-Digital converters

3.3. Interface devices

3.4. µ-controllers

4. Time & Embodiment
5. Asynchronism
6. Synchronisation
7. Scheduling
8. Resource control
9. Reliability & Fault-tolerance

Organization & ToC

Topics

1. *Introduction & Real-time languages*
2. *Physical coupling*
3. *Interfaces*
4. *Time & Embodiment*

4.1. What is time? / What is embodiment?

4.2. Time: notion, delays, time-out

4.3. Interfacing with time

4.4. Specifying timing requirements

4.5. Satisfying timing requirements

5. *Asynchronism*
6. *Synchronisation*
7. *Scheduling*
8. *Resource control*
9. *Reliability & Fault-tolerance*

Organization & ToC

Topics

1. *Introduction & Real-time languages*
2. *Physical coupling*
3. *Interfaces*
4. *Time & Embodiment*
5. *Asynchronism*

5.1. Interrupts, signals, exceptions

5.2. Atomic Actions

5.3. Asynchronous transfer of control

6. *Synchronisation*
7. *Scheduling*
8. *Resource control*
9. *Reliability & Fault-tolerance*

Organization & ToC

Topics

1. *Introduction & Real-time languages*
2. *Physical coupling*
3. *Interfaces*
4. *Time & Embodiment*
5. *Asynchronism*
6. *Synchronisation*

6.1. Variable-based synchronization

6.2. Message-based synchronization

7. *Scheduling*
8. *Resource control*
9. *Reliability & Fault-tolerance*

Organization & ToC

Topics

1. Introduction & Real-time languages
2. Physical coupling
3. Interfaces
4. Time & Embodiment
5. Asynchronism
6. Synchronisation
7. Scheduling

7.1. Basic real-time scheduling

7.2. Real-world extensions

7.3. Language support

8. Resource control
9. Reliability & Fault-tolerance

Organization & ToC

Topics

1. *Introduction & Real-time languages*
2. *Physical coupling*
3. *Interfaces*
4. *Time & Embodiment*
5. *Asynchronism*
6. *Synchronisation*
7. *Scheduling*
8. *Resource control*

8.1. Resource synchronization primitives

8.2. Resource reclaiming schemes

8.3. Real-time resource control

9. *Reliability & Fault-tolerance*

Organization & ToC

Topics

1. *Introduction & Real-time languages*
2. *Physical coupling*
3. *Interfaces*
4. *Time & Embodiment*
5. *Asynchronism*
6. *Synchronisation*
7. *Scheduling*
8. *Resource control*
9. *Reliability & Fault-tolerance*

9.1. Terminology

9.2. Faults

9.3. Redundancy

9.4. Reduce & Formalise

Organization & ToC

Table of Contents

1. Introduction & Real-Time Languages

1.1. Features (and non-features) of a real-time system

1.2. Components of a real-time system

1.3. Real-time languages criteria

1.4. Examples of actual real-time languages:

- Ada, Esterel, Pearl, VHDL, Timed CSP, Real-time JAVA, POSIX

2. Physical coupling

2.1. Physical phenomena

2.2. Measuring temperature

- Thermoelements, thermocouples, thermoresistors, thermistors, noise temperature measurement) and others

2.3. Measuring range and relative speed

- Triangulation, time of flight, intensity, Doppler methods, interferometry

2.4. Examples:

- Time-of flight ultrasound, time-of-flight laser, Doppler current profiler

3. Converters & Interfaces

3.1. Analogue signal chain in adigital system

- Sampling data, aliasing, Nyquist's criterion, oversampling
- Quantization (LSB, rms noise voltage, SNR, ENOB) – Missing codes, DNL, INL

3.2. A/D converters: flash, pipelined-flash, SAR, Σ-Δ, n-th order Σ-Δ

3.3. Examples:

- Fast and simple A/D converter example
- Multi-channel A/D data logging interface example
- Simple 8-bit µcontroller example
- Complex 32-bit µcontroller example: TPU: µprogramming, atomic states, µengine scheduling, max. latency analysis, NEXUS debugging port

3.4. General device handling / sampling control / language requirements

4. Time & Space

4.1. What is time? / What is embodiment?

- Approaches by different faculties to understand the basis for this course

4.2. Interfacing with time

- Formulating local time-dependent constraints – Access time, delay processes, detect timeouts (in different languages)

4.3. Specifying timing requirements

- Formulating global timing-constraints – Understanding time-scope parameters (and expressing them in different languages)

4.4. Satisfying timing requirements

- Real-time logic and complex systems approach

5. Asynchronism

5.1. Interrupts / Signals

- Device / system / language / operating-system level interrupt control
- Characteristics of interrupts and signals

5.2. Exceptions

- Exception classes / granularity / parametrisation / propagation – Resumption and termination, specific language issues

5.3. Atomic Actions

- Definition / requirements / failure cases / implementation / error recovery

5.4. Asynchronous transfer of control / Interrupts in context

- Interrupts and ATC in real-time Java and Ada

6. Synchronization

6.1. Shared memory based synchronization

- Flags, condition variables, semaphores, conditional critical regions, monitors, protected objects.
- Guard evaluation times, nested monitor calls, deadlocks, simultaneous reading, queue management.
- Synchronization and object orientation, blocking operations and re-queuing.

6.2. Message based synchronization

- Synchronization models, addressing modes, message structures
- Selective accepts, selective calls
- Indeterminism in message based synchronization

7. Scheduling

7.1. Basic real-time scheduling

- Fixed Priority Scheduling (FPS) with Rate Monotonic (RMPO) Deadline Monotonic Priority Ordering (DMPO)
- Earliest Deadline First (EDF)

7.2. Real-world extensions

- Aperiodic, sporadic, soft real-time tasks – Deadlines shorter than period – Cooperative and deferred pre-emption scheduling – Fault tolerance in terms of exception handling considerations – Synchronized talks (priority inheritance, priority ceiling protocols)

7.3. Language support

- Ada, POSIX (static, off-line analysis mostly) — RT-Java (on-line, dynamic scheduling)

8. Resource control

8.1. Resource synchronization primitives

- Evaluation criteria for resource synchronisation methods
- Atomicity, liveliness, and double interaction

8.2. Resource reclaiming schemes

- Basic reclaiming, early start, and restriction vector algorithms
- Resource reclaiming with task migration

8.3. Real-time resource control

- Policy and run-time issues to be considered

9. Reliability

9.1. Terminology

- Faults, Errors, Failures – Reliability

9.2. Faults

- Fault avoidance, removal, prevention, Fault tolerance

9.3. Redundancy

- Static (TMR, NMR) and dynamic redundancy
- N-version programming, and dynamic redundancy in software design

9.4. Reduce & Formalise

- Ada Ravenscar profile
- Real-time Logic