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.
This course will be given by

Uwe R. Zimmer

Tutoring and labs by

Benjamin Wang

Electronics design by

Mark Turner

[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.

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

1. **Introduction & Real-time languages**
   1. Staking out the field
   2. Features (and non-features) of a real-time system
   3. Components of a real-time system
   4. Real-time languages
      - Ada
      - Esterel
      - Pearl
      - VHDL
      - Timed CSP
      - Real-time JAVA
      - POSIX

2. **Physical coupling**
   1. Physical values
   2. Introduction to sensors
   3. Frequently employed sensors

3. **Interfaces**
   1. Analogue signal chain in a digital system
   2. Analog-Digital converters
   3. Interface devices
   4. μ-controllers

4. **Time & Embodiment**
   1. What is time? / What is embodiment?
   2. Time: notion, delays, time-out
   3. Interfacing with time
   4. Specifying timing requirements
   5. Satisfying timing requirements

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

---

### Topics

1. **Introduction & Real-time languages**
   1. Staking out the field
   2. Features (and non-features) of a real-time system
   3. Components of a real-time system

2. **Physical coupling**
   1. Physical values
   2. Introduction to sensors
   3. Frequently employed sensors

3. **Interfaces**

4. **Time & Embodiment**
   1. What is time? / What is embodiment?
   2. Time: notion, delays, time-out
   3. Interfacing with time
   4. Specifying timing requirements
   5. Satisfying timing requirements

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

---

### Topics

1. **Introduction & Real-time languages**
2. **Physical coupling**
3. **Interfaces**
4. **Time & Embodiment**
   1. Analogue signal chain in a digital system
   2. Analog-Digital converters
   3. Interface devices
   4. μ-controllers

5. **Asynchronism**
6. **Synchronisation**
7. **Scheduling**
8. Resource control
9. Reliability & Fault-tolerance
1. Introduction & Real-time languages
   5.1. Interrupts, signals, exceptions
   5.2. Atomic Actions
   5.3. Asynchronous transfer of control
2. Physical coupling
3. Interfaces
4. Time & Embodiment
5. Asynchronism
6. Synchronisation
7. Scheduling
8. Resource control
9. Reliability & Fault-tolerance

7.1. Basic real-time scheduling
7.2. Real-world extensions
7.3. Language support
8. Resource control
9. Reliability & Fault-tolerance

1. Introduction & Real-time languages
6.1. Variable-based synchronization
6.2. Message-based synchronization
7. Scheduling
8. Resource control
9. Reliability & Fault-tolerance

6.1. Variable-based synchronization
6.2. Message-based synchronization
7. Scheduling
8. Resource control
9. Reliability & Fault-tolerance

8.1. Resource synchronization primitives
8.2. Resource reclaiming schemes
8.3. Real-time resource control
8. Resource control
9. Reliability & Fault-tolerance
### Table of Contents

| 1. Introduction & Real-time Languages | 9.1. Terminology |
| 1.1. Features (and non-features) | 3.4. Context handler sampling control / language requirements |
| 1.2. Components of a real-time system | 1.4. Asynchronous transfer of control / Interrupts in context |
| 1.3. Real-time languages criteria | 1.5. What is time? / What is embodiment? |
| 1.4. Examples of actual real-time languages: | 1.6. Thermoelement, thermocouple, thermistor, noise temperature measurement |
| 2. Physical coupling | 2.7. Earliest Deadline First (EDF) |
| 2.1. Physical phenomena | 2.8. Fixed Priority Scheduling (FPS) with Rate Monotonic (RM) Deadline Monotonic Priority Ordering (DM) |
| 2.2. Measuring temperature | 2.9. Aperiodic, sporadic, soft real-time tasks – Deadlines shorter than period – |
| 2.3. Measuring range and relative speed | 2.10. Earliest Deadline First (EDF) |
| 2.4. Examples | 2.11. Earliest Deadline First (EDF) |
| 3.3. A/D Converters (flash, pipelined, Sigma-Delta) | 3.15. Earliest Deadline First (EDF) |
| 3.4. Examples | 3.16. Earliest Deadline First (EDF) |
| 4. Time & Space | 4.1. What is time? / What is embodiment? |
| 4.1. What is time? / What is embodiment? | 4.2. Approach to different technologies |
| 4.2. Approach to different technologies | 4.3. Defining local time-dependent constraints – Access time, delay processes, double interaction |
| 4.3. Defining global timing constraints | 4.4. Satisfaction timing requirements – A/D, D/A, DAC, event counter implementations, zero recovery |
| 4.4. Satisfaction timing requirements – A/D, D/A, DAC, event counter implementations, zero recovery | 5.1. Communication |
| 5.2. Exceptions | 5.1.1. Asynchronous / Synchronous |
| 5.3. Atomic Actions | 5.1.2. Error Handling / Error Recovery |
| 6. Synchronisation | 5.1.3. Error Handling / Error Recovery |
| 6.1. What is time, interval time | 5.2.1. Reliability, fault tolerance |
| 6.2. Asynchronism | 5.2.2. Reliability, fault tolerance |
| 7. Scheduling | 5.2.3. Reliability, fault tolerance |
| 7.1. Basic real-time scheduling | 5.2.4. Reliability, fault tolerance |
| 7.2. Real-time scheduling | 5.2.5. Reliability, fault tolerance |
| 7.3. Scheduling | 5.3.2. Aperiodic, sporadic, soft real-time tasks – Deadlines shorter than period – |
| 7.4. Aperiodic, sporadic, soft real-time tasks – Deadlines shorter than period – | 5.3.3. Earliest Deadline First (EDF) |
| 5.3.3. Earliest Deadline First (EDF) | 5.4. Earliest Deadline First (EDF) |
| 5.3.3. Earliest Deadline First (EDF) | 5.4. Earliest Deadline First (EDF) |
| 5.3.3. Earliest Deadline First (EDF) | 5.4. Earliest Deadline First (EDF) |

### Topics

1. **Introduction & Real-time Languages**
   - Introduction and Real-time Languages
   - Real-time Languages Criteria
   - Examples of Actual Real-time Languages

2. **Physical Coupling**
   - Physical Phenomena
   - Measuring Temperature
   - Measuring Range and Relative Speed
   - Examples

3. **Converters & Interfaces**
   - Analogue Signal Chain in Digital Systems
   - Sampling, Holding, Quantisation, Oversampling
   - A/D Converters (Flash, Pipelined, Sigma-Delta)
   - Examples

4. **Time & Space**
   - What is Time? / What is Embodiment?
   - Approach to Different Technologies
   - Defining Local Time-Dependent Constraints
   - Satisfaction Timing Requirements

5. **Synchronisation**
   - Communication
   - Exceptions
   - Atomic Actions

6. **Scheduling**
   - Basic Real-Time Scheduling
   - Real-Time Scheduling
   - Aperiodic, Sporadic, Soft Real-Time Tasks – Deadlines Shorter than Period –

8. **Resource control**
   - Policies and policies issues to be considered

9. **Reliability & Fault-tolerance**
   - Introduction
   - Faults, Events, Failures – Reliability
   - Faults
   - Fault Avoidance, Removal, Prevention
   - Reliability
   - Reliability in Software Design
   - Reliability in Hardware
   - Reliability in Software Design
   - Reliability in Hardware
   - Reliability in Software Design