

Real-Time & Embedded Systems 2019



10

Summary

Uwe R. Zimmer - The Australian National University

Summary

Table of Contents

- | | | | |
|---|---|---|---|
| <p>1. Introduction & Real-Time Languages</p> <p>1.1. Features (and non-features) of a real-time system</p> <p>1.2. Components of a real-time system</p> <p>1.3. Real-time languages criteria</p> <p>1.4. Real-time languages: CSP, Esterel, Real VPL, Timed CSP, Real-time Java, POSIX</p> <p>2. Physical coupling</p> <p>2.1. Physical phenomena</p> <p>2.2. Measuring temperature</p> <p>2.3. Measuring range and relative speed</p> <p>2.4. Examples</p> <p>3. Converters & Interfaces</p> <p>3.1. Analogue signal chain in digital system</p> <p>3.2. AD converters: flash pipelined, SAR, EROB – Missing codes, DNL, INL</p> <p>3.3. Examples</p> <p>4. Time & Space</p> <p>4.1. What is time? / What is embodiment?</p> <p>4.2. Interfacing with time</p> <p>4.3. Synchronizing timing requirements</p> <p>4.4. Subsystem timing requirements</p> | <p>Identifiers / requirements / failure cases / implementation / error recovery</p> <p>7.3. Language support</p> <ul style="list-style-type: none"> • Ada, POSIX (static, off-line analysis mostly) – RT-Java (on-line, dynamic scheduling) <p>8. Resource control</p> <p>8.1. Resource synchronization primitives</p> <p>8.2. Resource reclaiming schemes</p> <p>8.3. Real-time resource control</p> <p>9. Reliability</p> <p>9.1. Terminology</p> <p>9.2. Faults</p> <p>9.3. Redundancy</p> <p>9.4. Redundant tasks</p> | <p>Definers / requirements / failure cases / implementation / error recovery</p> <p>5.4. Asynchronous transfer of control / interrupts in context</p> <p>• Interrupts and AIC in real-time Java and Ada</p> <p>6. Synchronization</p> <p>6.1. Shared memory based synchronization</p> <p>6.2. Message based synchronization</p> <p>6.3. Synchronization models, addressing modes, message structures</p> <p>6.4. Indeterminism in message based synchronization</p> <p>7. Scheduling</p> <p>7.1. Basic real-time scheduling</p> <p>7.2. Real-time task (RTT)</p> <p>7.3. Real-time task (RTT)</p> <p>7.4. Real-time task (RTT)</p> <p>7.5. Real-time task (RTT)</p> <p>7.6. Real-time task (RTT)</p> <p>7.7. Real-time task (RTT)</p> <p>7.8. Real-time task (RTT)</p> <p>7.9. Real-time task (RTT)</p> <p>7.10. Real-time task (RTT)</p> <p>7.11. Real-time task (RTT)</p> <p>7.12. Real-time task (RTT)</p> <p>7.13. Real-time task (RTT)</p> <p>7.14. Real-time task (RTT)</p> <p>7.15. Real-time task (RTT)</p> <p>7.16. Real-time task (RTT)</p> <p>7.17. Real-time task (RTT)</p> <p>7.18. Real-time task (RTT)</p> <p>7.19. Real-time task (RTT)</p> <p>7.20. Real-time task (RTT)</p> <p>7.21. Real-time task (RTT)</p> <p>7.22. Real-time task (RTT)</p> <p>7.23. Real-time task (RTT)</p> <p>7.24. Real-time task (RTT)</p> <p>7.25. Real-time task (RTT)</p> <p>7.26. Real-time task (RTT)</p> <p>7.27. Real-time task (RTT)</p> <p>7.28. Real-time task (RTT)</p> <p>7.29. Real-time task (RTT)</p> <p>7.30. Real-time task (RTT)</p> <p>7.31. Real-time task (RTT)</p> <p>7.32. Real-time task (RTT)</p> <p>7.33. Real-time task (RTT)</p> <p>7.34. Real-time task (RTT)</p> <p>7.35. Real-time task (RTT)</p> <p>7.36. Real-time task (RTT)</p> <p>7.37. Real-time task (RTT)</p> <p>7.38. Real-time task (RTT)</p> <p>7.39. Real-time task (RTT)</p> <p>7.40. Real-time task (RTT)</p> <p>7.41. Real-time task (RTT)</p> <p>7.42. Real-time task (RTT)</p> <p>7.43. Real-time task (RTT)</p> <p>7.44. Real-time task (RTT)</p> <p>7.45. Real-time task (RTT)</p> <p>7.46. Real-time task (RTT)</p> <p>7.47. Real-time task (RTT)</p> <p>7.48. Real-time task (RTT)</p> <p>7.49. Real-time task (RTT)</p> <p>7.50. Real-time task (RTT)</p> <p>7.51. Real-time task (RTT)</p> <p>7.52. Real-time task (RTT)</p> <p>7.53. Real-time task (RTT)</p> <p>7.54. Real-time task (RTT)</p> <p>7.55. Real-time task (RTT)</p> <p>7.56. Real-time task (RTT)</p> <p>7.57. Real-time task (RTT)</p> <p>7.58. Real-time task (RTT)</p> <p>7.59. Real-time task (RTT)</p> <p>7.60. Real-time task (RTT)</p> <p>7.61. Real-time task (RTT)</p> <p>7.62. Real-time task (RTT)</p> <p>7.63. Real-time task (RTT)</p> <p>7.64. Real-time task (RTT)</p> <p>7.65. Real-time task (RTT)</p> <p>7.66. Real-time task (RTT)</p> <p>7.67. Real-time task (RTT)</p> <p>7.68. Real-time task (RTT)</p> <p>7.69. Real-time task (RTT)</p> <p>7.70. Real-time task (RTT)</p> <p>7.71. Real-time task (RTT)</p> <p>7.72. Real-time task (RTT)</p> <p>7.73. Real-time task (RTT)</p> <p>7.74. Real-time task (RTT)</p> <p>7.75. Real-time task (RTT)</p> <p>7.76. Real-time task (RTT)</p> <p>7.77. Real-time task (RTT)</p> <p>7.78. Real-time task (RTT)</p> <p>7.79. Real-time task (RTT)</p> <p>7.80. Real-time task (RTT)</p> <p>7.81. Real-time task (RTT)</p> <p>7.82. Real-time task (RTT)</p> <p>7.83. Real-time task (RTT)</p> <p>7.84. Real-time task (RTT)</p> <p>7.85. Real-time task (RTT)</p> <p>7.86. Real-time task (RTT)</p> <p>7.87. Real-time task (RTT)</p> <p>7.88. Real-time task (RTT)</p> <p>7.89. Real-time task (RTT)</p> <p>7.90. Real-time task (RTT)</p> <p>7.91. Real-time task (RTT)</p> <p>7.92. Real-time task (RTT)</p> <p>7.93. Real-time task (RTT)</p> <p>7.94. Real-time task (RTT)</p> <p>7.95. Real-time task (RTT)</p> <p>7.96. Real-time task (RTT)</p> <p>7.97. Real-time task (RTT)</p> <p>7.98. Real-time task (RTT)</p> <p>7.99. Real-time task (RTT)</p> <p>7.100. Real-time task (RTT)</p> | <p>Identifiers / requirements / failure cases / implementation / error recovery</p> <p>7.3. Language support</p> <ul style="list-style-type: none"> • Ada, POSIX (static, off-line analysis mostly) – RT-Java (on-line, dynamic scheduling) <p>8. Resource control</p> <p>8.1. Resource synchronization primitives</p> <p>8.2. Resource reclaiming schemes</p> <p>8.3. Real-time resource control</p> <p>9. Reliability</p> <p>9.1. Terminology</p> <p>9.2. Faults</p> <p>9.3. Redundancy</p> <p>9.4. Redundant tasks</p> |
|---|---|---|---|

Summary

Summary

Introduction & Real-Time Languages

- **Features** (and non-features) of a real-time system
 - Features, definitions, scenarios, and characteristics.
- **Components of a real-time system**
 - Converters, interfaces, sensors, actuators, communication systems, controllers, ...
- **Software layers of a real-time system**
 - Algorithms, operating systems, protocols, languages, concurrent and distributed systems.
- **Real-time languages criteria**
 - Mostly high integrity, predictable languages with means for explicit time scopes.
- **Examples of actual real-time languages**

Summary

Summary

Physical coupling

- **Physical phenomena**
- **Measuring temperature**
 - Thermoelements, thermocouples, Thermoresistors, Thermistors, Noise temperature (measurement) and many others ...
- **Measuring range and relative speed**
 - Triangulation, Time of flight, Intensity, Doppler methods, Interferometry
- **Examples: Common acoustical and optical sensors**



Summary

Summary

Converters & Interfaces

- **Analogue signal chain in a digital system**
 - Sampling data, aliasing, Nyquist's criterion, oversampling
 - Quantization (LSB, rms noise voltage, SNR, ENOB), Missing codes, DNLL, INL
- **A/D converters:**
 - Integrating (Single- / Dual-slope), Flash, Pipelined, SAR, Tracking, $\Sigma\text{-}\Delta$, $\Sigma\text{-}\Delta$ DDA, n-th order $\Sigma\text{-}\Delta$.
- **Examples:**
 - Fast and simple A/D converter example: National Semiconductor ADC08200
 - Multi-channel A/D data logging interface example: National Semiconductor LM12L458
 - Simple 8-bit μ -controller example: Motorola MC68HC05, Propeller.
 - Complex 32-bit μ -controller examples: AVR32 and Motorola MPC565 (including TPU.s).
- **General device handling / sampling control / language requirements**



Summary

Summary

Asynchronism

- **Interrupts / Signals**
 - Device / system / language / operating-system level interrupt control.
 - Characteristics of interrupts and signals.
- **Exceptions**
 - Exception classes / granularity / parametrisation / propagation.
 - Resumption and termination, specific language issues.
- **Atomic Actions**
 - Definition / requirements / failure cases / implementation / error recovery.
- **Asynchronous transfer of control / Interrupts in context**
 - Interrupts and ATC in real-time Java and Ada.



Summary

Summary

Time & Space

- **What is time? / What is embodiment?**
 - Approaches by different faculties to understand the foundations of this course
- **Interfacing with time**
 - Formulating local, time-dependent constraints
 - Access time, delay processes, timers
 - Timeouts, asynchronous transfer of control
- **Specifying timing requirements**
 - Formulating global timing-constraints
 - Understanding time-scope parameters (and expressing them in different languages)
- **Satisfying timing requirements**
 - Real-time logic approach & Complex systems approach



Summary

Summary

Synchronization

- **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.
- **Message based synchronization**
 - Synchronization models
 - Addressing modes
 - Message structures
 - Examples



Summary

Summary

Scheduling

- **Basic real-time scheduling**
 - Fixed Priority Scheduling (FPS) with Rate Monotonic (RMPO) and Deadline Monotonic Priority Ordering (DMPO).
 - Earliest Deadline First (EDF).
- **Real-world extensions**
 - Aperiodic, sporadic, soft real-time tasks.
 - Deadlines different from period.
 - Synchronized tasks (priority inheritance, priority ceiling protocols).
 - Cooperative and deferred pre-emption scheduling.
 - Fault tolerance in terms of exception handling considerations.
- **Language support**
 - Ada, POSIX

© 2019 Uwe R. Zimmer, The Australian National University

page 958 of 961 (chapter 10: "Summary" up to page 961)



Summary

Summary

Resource Control

- **Resource synchronization primitives**
 - Evaluation criteria for resource synchronization methods.
 - Atomicity, liveness, and double interaction.
- **Resource reclaiming schemes**
 - Basic reclaiming
 - Early start algorithm
 - Restriction vector
 - Resource reclaiming with task migration
- **Real-time resource control**
 - Policy and run-time issues to be considered.

© 2019 Uwe R. Zimmer, The Australian National University


page 959 of 961 (chapter 10: "Summary" up to page 961)



Summary

Summary

Reliability

- **Terminology**
 - Faults, Errors, Failures – Reliability.
- **Faults**
 - Fault avoidance, removal, prevention  Fault tolerance.
- **Redundancy**
 - Static (TMR, NMR) and dynamic redundancy.
 - N-version programming, and dynamic redundancy in software design.
- **Reduce & Formalise**
 - Ravenscar profile.
 - Real-time logic.

© 2019 Uwe R. Zimmer, The Australian National University

page 960 of 961 (chapter 10: "Summary" up to page 961)

