



Echtzeitfähige Kommunikation

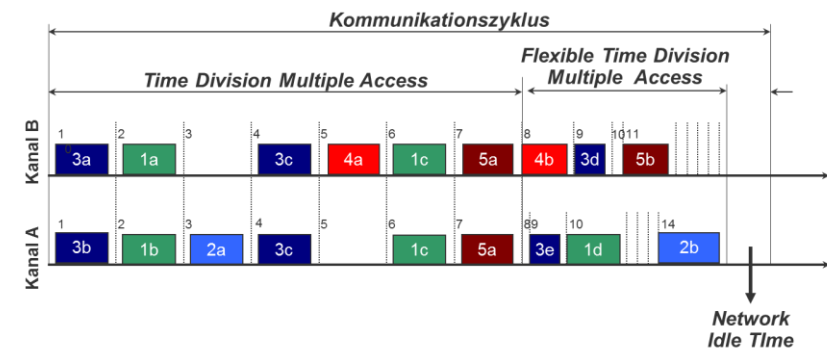
Flexray

Flexray

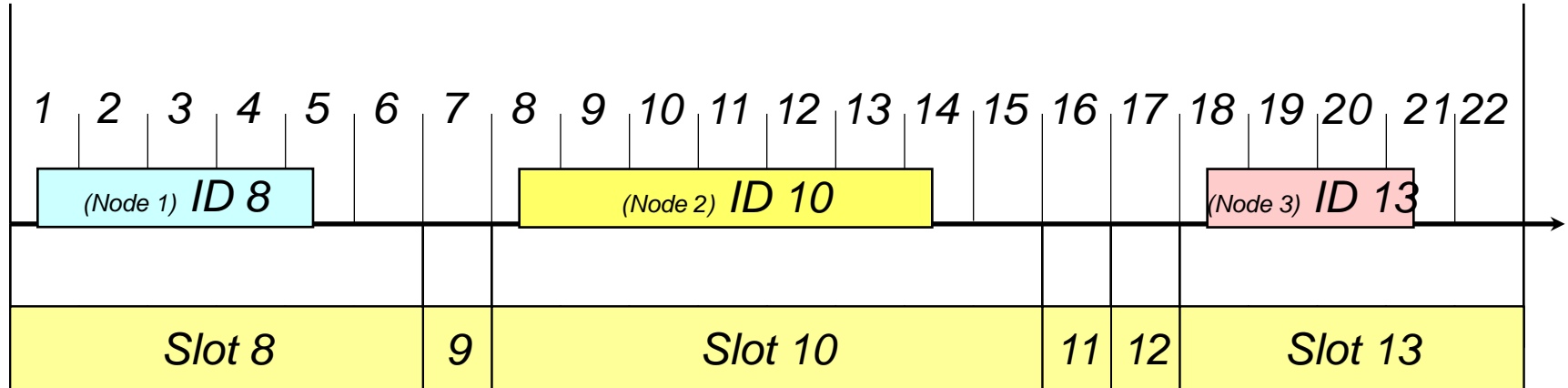
- Motivation:
 - Entwicklung eines kostengünstigen und damit automotivetauglichen Kommunikationsprotokolls, dass sowohl zeit- als auch ereignisgesteuerte Datenübertragung ermöglicht.
- Zwei Schemata integriert:
 1. TDMA – time division multiple access (statisches Segment)
 2. FTDMA – flexible time division multiple access (dynamisches Segment), zusätzlich implementiert FTDMA bei Flexray eine Priorisierung
- Grundlegende Kommunikationsstruktur:
 - Unterteilung in Zyklen (Zyklus besteht aus statischem u. dynamischen Segment, sowie Bereichen zur Synchronisierung)
 - Es gibt 64 dieser Zyklen, wobei diese sich unterscheiden können

		CAN	
Kanäle	1	1	1/2
Bus Datenrate	20 Kbit/sec	≤ 1 Mbit/sec	10 Mbits/sec
Zeitgesteuert	nein	nein	ja
Deterministisch	nein	nein	ja
Redundanz	nein	nein	möglich
Arbitrierung	Master	CSMA	TDMA

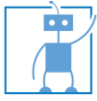
FlexRay komplementiert CAN und LIN



FTDMA in Flexray



- Einführung von Minislots, die genau einem Knoten zur Verfügung stehen
- Will dieser Knoten in einem Minislot senden, so verlängert sich der Minislot entsprechend → es fallen am Ende des Zyklus Minislots weg, es muss aber sichergestellt werden, dass die Nachricht auf jeden Fall bis zum Ende des gesamten Zyklus beendet ist
- Frühere Minislots stehen somit für eine höhere Priorität



Echtzeitfähige Kommunikation

Zusammenfassung

Zusammenfassung

- Die Eignung eines Kommunikationsmediums für die Anwendung in Echtzeitsystemen ist vor allem durch das Medienzugriffsverfahren bestimmt.
- Die maximale Wartezeit ist bei
 - CSMA/CD: unbegrenzt und nicht deterministisch bzw. bei Switched Ethernet keine Priorisierung \leftrightarrow keine Eignung für Echtzeitsysteme
 - CSMA/CA, tokenbasierten Verfahren: begrenzt, aber nicht deterministisch (abhängig von anderen höherpriorigen Nachrichten)
 - zeitgesteuerten Verfahren: begrenzt und deterministisch.
- Die Priorisierung der Nachrichten wird von CSMA/CA und tokenbasierten Verfahren unterstützt.
- Nachteil der zeitgesteuerten Verfahren ist die mangelnde Flexibilität (keine dynamischen Nachrichten möglich).
- Trotz diverser Nachteile geht der Trend hin zum Ethernet.

Trends: Real-Time Ethernet

- Es existieren verschiedene Ansätze
 - Beispiel: Ethercat von Beckhoff
 - Die Nachrichten entsprechen dem Standardnachrichtenformat von Ethernet
 - Pakete werden von einem Master initiiert und werden von den Teilnehmern jeweils weitergeleitet.
 - Jeder Knoten entnimmt die für ihn bestimmten Daten und kann eigene Daten anfügen.
 - Die Bearbeitung erfolgt on-the-fly, dadurch kann die Verzögerung minimiert werden.
 - Beispiel: Profinet von Siemens
 - Drei verschiedene Protokollstufen (TCP/IP – Reaktionszeit im Bereich von 100ms, Real-time Protocol - bis 10ms, Isochronous Real-Time - unter 1ms)
 - Profinet IRT benutzt vorher bekannte, reservierte Zeitschlitze zur Übertragung von echtzeitkritischen Daten, in der übrigen Zeit wird das Standard-Ethernet Protokoll ausgeführt
 - IEEE Time-Sensitive Networking Task Group (IEEE 802.1)
 - Entwicklung eines neuen Standards auf Ethernetbasis, u.a. mit Nachrichtenpräemption, Uhrensynchronisation (auf Basis von IEEE 1588), Reservierung
 - Nachfolgestandard von AVB (Audio-Video-Bridging) mit dem Ziel weitere Domänen zu unterstützen

Was Sie aus diesem Kapitel mitgenommen haben sollten

- Kenntnisse der für Echtzeitsysteme relevanten Schichten des ISO/OSI-Modells
- Verschiedenen Mechanismen der Medienzugriffsschicht und ihre Bewertung hinsichtlich der Echtzeitfähigkeit, insbesondere für welche Nachrichten einfache Aussagen zur maximalen Verzögerung gemacht werden können
- Verschiedene Kommunikationstechnologien und deren wesentliche Konzepte hinsichtlich der ISO/OSI-Schichten
- Ansätze für Echtzeit-Ethernet (wesentliche Konzepte)

Klausurfragen

- Klausur Wintersemester 07/08 (4 Punkte = 4min)
 - Erläutern Sie kurz die wesentlichen Unterschiede zwischen TokenRing, TokenBus und Ethercat in Bezug auf Topologie und Mediumszugriffverfahren.
A: TokenRing benötigt einen physikalischen Ring, TokenBus und Ethercat beruhen auf einer beliebigen Topologie und realisieren den Ring auf logischer Ebene; bei TokenRing & TokenBus kann jeder Knoten eine Nachricht senden, wenn er im Besitz des Tokens ist, bei Ethercat werden Nachrichten immer vom Master initiiert und sonstige Knoten hängen ihre Daten an
- Wiederholungsfragen:
 1. Was ist der Unterschied zwischen dominanten und rezessiven Bits.
A: Falls gleichzeitig dominante und rezessive Bits gesendet werden, ist am Bus das dominante Bit zu sehen.
 2. Nennen Sie zwei Mechanismen zur Bitsynchronisierung und erklären Sie diese.
A: Bit Stuffing oder Manchestercodierung
 3. Was ist der Unterschied zwischen CSMA/CD und CSMA/CA?
A: CD kann Kollisionen nur erkennen, CA löst diese, so dass keine Nachricht verloren geht.
 4. Erläutern Sie zwei verschiedene Ansätze um Ethernet echtzeitfähig zu machen.
A: siehe Folie 361
 5. Beurteilen Sie die Kommunikationsprotokolle Ethernet, CAN, TTP nach Ihrer Echtzeitfähigkeit und gehen Sie vor allem auf die Möglichkeit zur Vorhersage der maximalen Nachrichtenlatenz ein.
A: siehe Folie 360



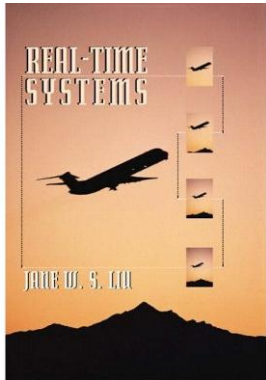
Kapitel 7

Echtzeitbetriebssysteme

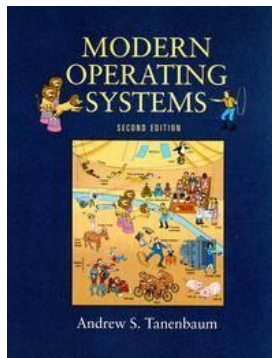
Inhalt

- Grundlagen
- Betrachtung diverser Betriebssysteme:
 - Domänenspezifische Betriebssysteme:
 - OSEK
 - TinyOS
 - Klassische Echtzeitbetriebssysteme
 - QNX
 - VxWorks
 - PikeOS
 - Linux- / Windows-Echtzeitvarianten
 - RTLinux/RTAI
 - Linux Kernel 2.6
 - Windows CE

Literatur



Jane W. S. Liu, Real-Time
Systems, 2000



Andrew S. Tanenbaum: Modern
Operating Systems, 2001

Dieter Zöbel, Wolfgang Albrecht:
Echtzeitsysteme: Grundlagen und
Techniken, 1995



Arnd Heursch et al.: Time-critical tasks in Linux 2.6, 2004

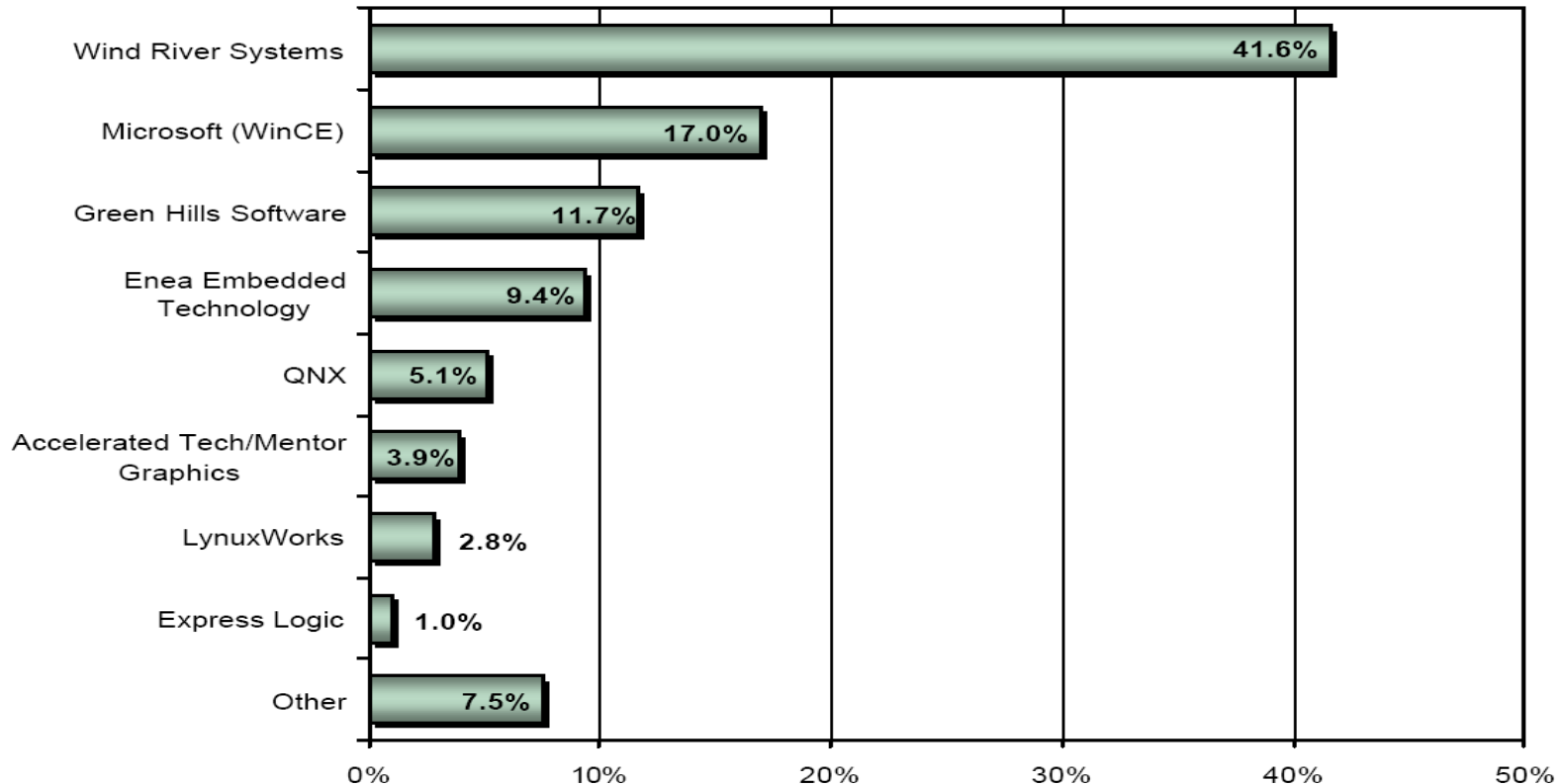
http://inf3-www.informatik.unibw-muenchen.de/research/linux/hannover/automation_conf04.pdf



Interessante Links

- <http://www.mnis.fr/en/support/doc/rtos/>
- <http://aeolean.com/html/RealTimeLinux/RealTimeLinuxReport-2.0.0.pdf>
- <http://www.osek-vdx.org/>
- <http://www.qnx.com/>
- <http://www.windriver.de>
- <http://www.fsmlabs.com>
- <http://www.rtai.org>
- <http://www.tinyos.net/>
- <http://www.sysgo.com/>

Marktaufteilung (Stand 2004)



Marktanteil am Umsatz, Gesamtvolumen 493 Mio. Dollar, Quelle: The Embedded Software Strategic Market Intelligence Program 2005

Anforderungen an Echtzeitbetriebssysteme

- Echtzeitbetriebssysteme unterliegen anderen Anforderungen als Standardbetriebssysteme:
 - stabiler Betrieb rund um die Uhr
 - definierte Reaktionszeiten
 - parallele Prozesse
 - schneller Prozesswechsel (geringer Prozesskontext)
 - echtzeitfähige Unterbrechensbehandlung
 - echtzeitfähiges Scheduling
 - echtzeitfähige Prozesskommunikation
 - umfangreiche Zeitdienste (absolute, relative Uhren, Weckdienste)
 - einfaches Speichermanagement

Fortsetzung

- Unterstützung bei der Ein- und Ausgabe
 - vielfältigste Peripherie
 - direkter Zugriff auf Hardware-Adressen und -Register durch den Benutzer
 - Treiber in Benutzerprozessen möglichst schnell und einfach zu implementieren
 - dynamisches Binden an den Systemkern
 - direkte Nutzung DMA
 - keine mehrfachen Puffer: direkt vom Benutzerpuffer auf das Gerät
- Einfachste Dateistrukturen
- Protokoll für Feldbus oder LAN-Bus, möglichst hardwareunterstützt
- Aufteilung der Betriebssystemfunktionalität in optionale Komponenten (Skalierbarkeit)



Echtzeitbetriebssysteme

Kriterien zur Beurteilung

Beurteilung von Echtzeitbetriebssystemen

- Folgende Aspekte werden wir genauer betrachten:
 - Schedulingverfahren
 - Prozessmanagement
 - Speicherbedarf (Footprint)
 - Garantierte Reaktionszeiten

Schedulingverfahren

- Fragestellung:
 - Welche Konzepte sind für das Scheduling von Prozessen verfügbar?
 - Gibt es Verfahren für periodische Prozesse?
 - Wie wird dem Problem der Prioritätsinversion begegnet?
 - Wann kann eine Ausführung unterbrochen werden?

Arten von Betriebssystemen

- Betriebssysteme werden in drei Klassen unterteilt:
 - Betriebssysteme mit **kooperativen Scheduling**: es können verschiedene Prozesse parallel ausgeführt werden. Der Dispatcher kann aber einem Prozess den Prozessor nicht entziehen, vielmehr ist das Betriebssystem auf die Kooperation der Prozesse angewiesen (z.B. Windows 95/98/ME)
 - Betriebssysteme mit **präemptiven Scheduling**: einem laufenden Prozess kann der Prozessor entzogen werden, falls sich der Prozess im Userspace befindet. (z.B. Linux, Windows 2000/XP)
 - **Präemptible Betriebssysteme**: der Prozessor kann dem laufenden Prozess jederzeit entzogen werden, auch wenn sich dieser im Kernelkontext ausgeführt wird.

→ Echtzeitsysteme müssen präemptibel sein.

Prozessmanagement

- Bewertung eines Betriebssystems nach:
 - Beschränkung der Anzahl von Prozessen
 - Möglichkeiten zur Interprozesskommunikation
 - Kompatibilität der API mit Standards (z.B. POSIX) zur Erhöhung der Portabilität

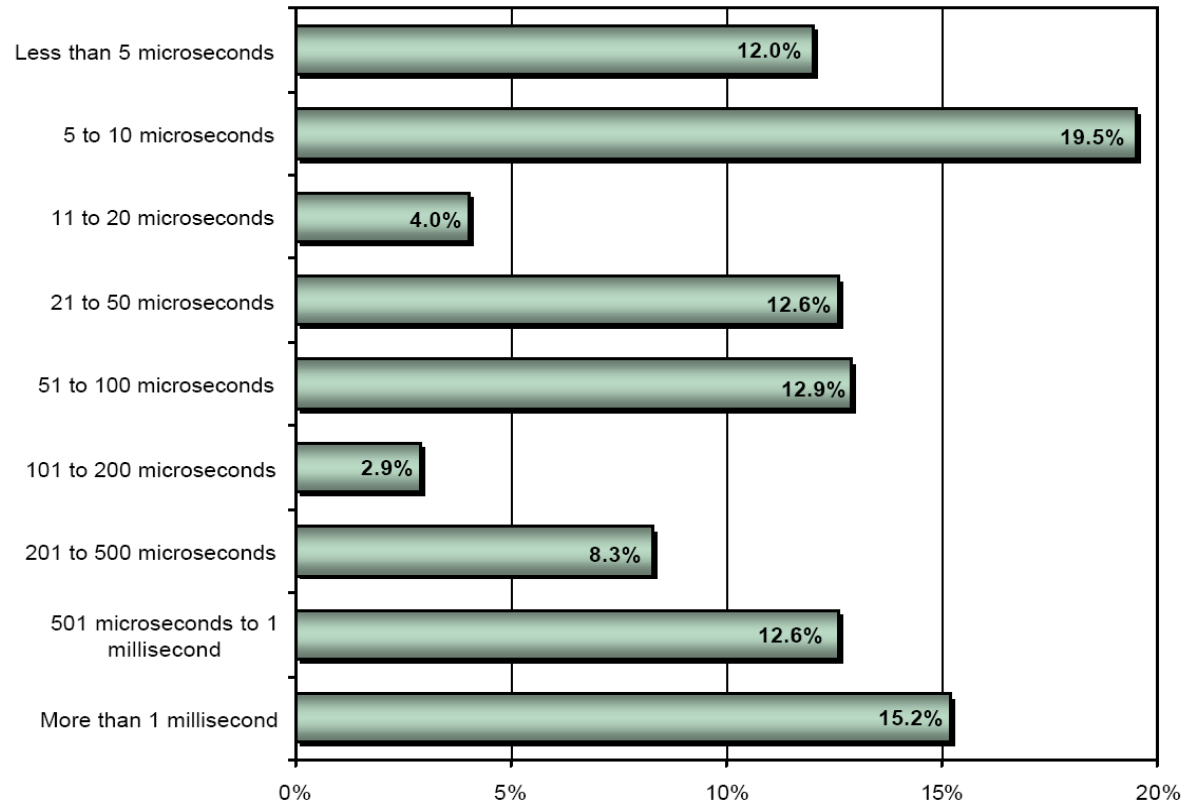
Speicherbedarf

- Echtzeitbetriebssysteme werden auf sehr unterschiedlicher Hardware ausgeführt
 - Der verfügbare Speicher variiert sehr stark.
 - Typische Betriebssystemfunktionalitäten (z.B. Dateisysteme, graphische Oberfläche) werden oft gar nicht benötigt.
- Echtzeitsysteme müssen aus diesen Gründen skalierbar sein:
 - Möglichkeit zur Auswahl einzelner Module entsprechend den Anforderungen an die Funktionalität der Anwendung.
 - Entscheidend ist der **minimale Speicherbedarf (Footprint)**.

Reaktionszeiten

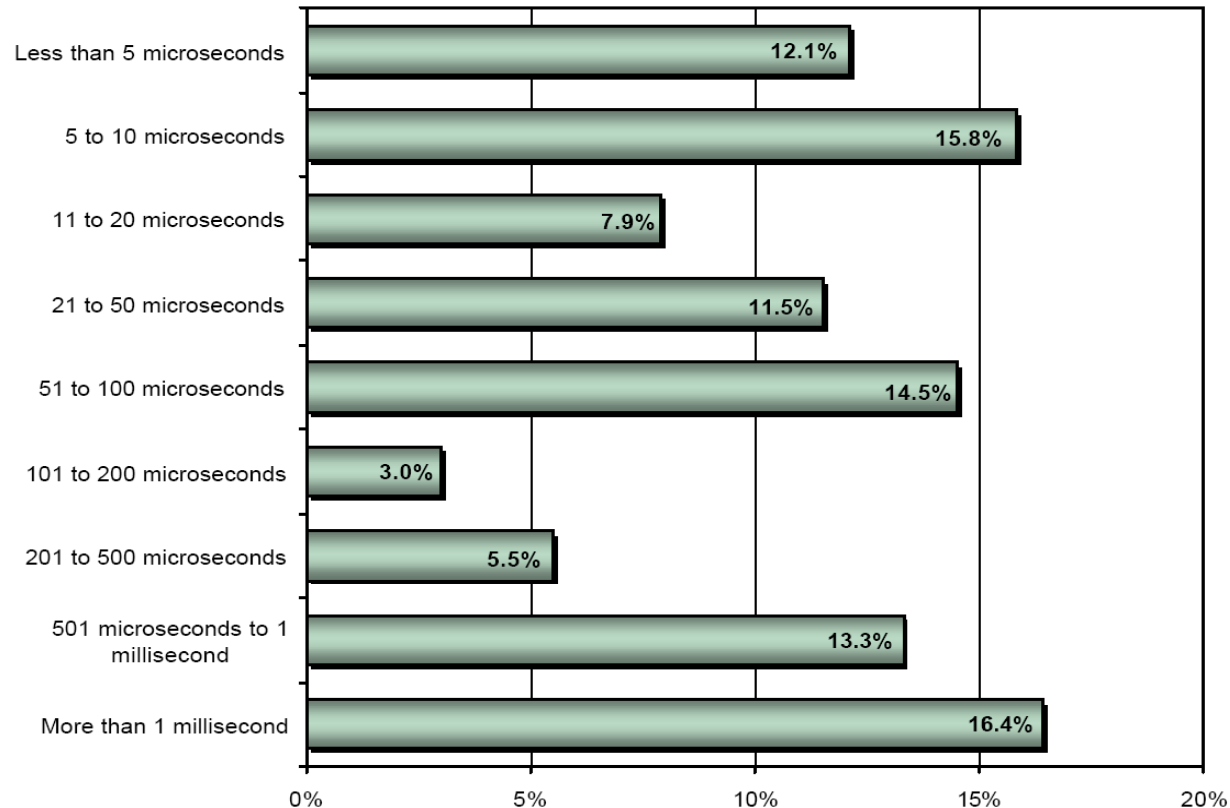
- Die Echtzeitfähigkeit wird durch die Messung folgender Zeiten bestimmt:
 - **Unterbrechungsantwortzeiten (interrupt latency)**: der Zeitraum zwischen dem Auftreten einer Unterbrechung und der Ausführung des ersten Befehls der dazugehörigen Unterbrechungsbehandlungsroutine
 - **Schedulinglatenz (scheduling latency)**: Zeit von der Ausführung des letzten Befehls des Unterbrechungsbehandlers bis zur Ausführung der ersten Instruktion des Prozesses, der durch das Auftreten der Unterbrechung in den bereiten Zustand wechselt.
 - Zeiten für einen **Kontextwechsel (context switch latency)**: Zeit von der Ausführung des letzten Befehls eines Prozesses im Userspace bis zur Ausführung der ersten Instruktion des nächsten Prozesses im Userspace.

Anforderungen an Unterbrechungsantwortzeiten



Typische Anforderungen an Antwortzeiten, Quelle: The Embedded Software Strategic Market Intelligence Program 2005

Anforderungen an Kontextwechselzeiten



Typische Anforderungen an den Kontextwechsel, Quelle: The Embedded Software Strategic Market Intelligence Program 2005