

Einsatz des Betriebssystems QNX Neutrino in der Medizintechnik

MedConf 2013 – München

16.10.2013

Mechatronic AG

Bernd Mohr

Mechatronic AG

Ihr Systempartner für

- Entwicklung
- Testequipment
- Beratung
- Production engineering
- Fertigung



Hier finden Sie uns

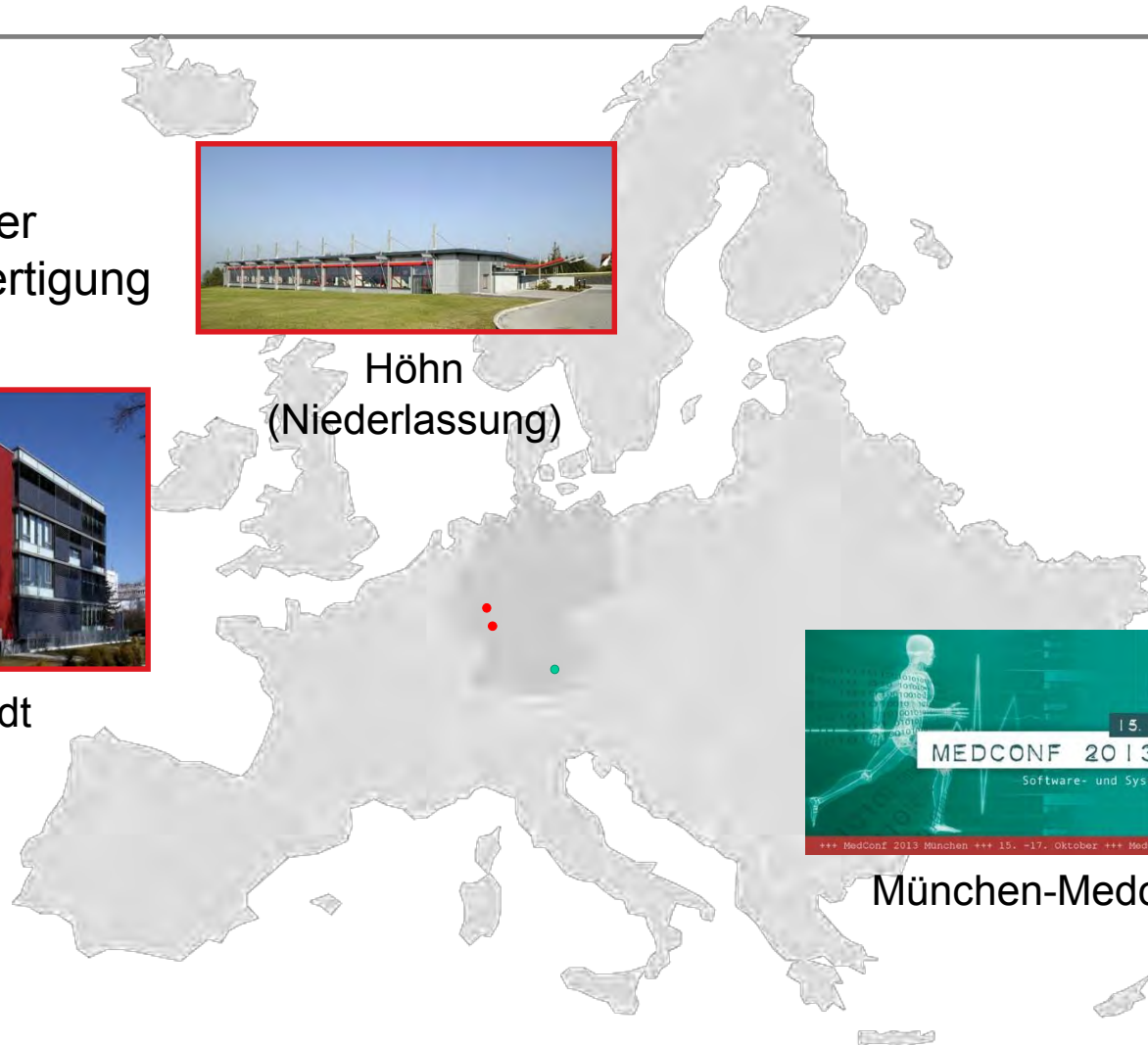
80 Mitarbeiter
R&D und Fertigung



Höhn
(Niederlassung)



Darmstadt



München-Medconf-Stand 5

Agenda

1. Vor- und Nachteile eines Betriebssystems
2. Kriterien für die Betriebssystemauswahl
3. Kurzbeschreibung des Projekts
4. QNX Neutrino
5. Herausforderungen
6. Erkenntnisse
7. Fazit

Agenda

1. Vor- und Nachteile eines Betriebssystems

2. Kriterien für die Betriebssystemauswahl

3. Kurzbeschreibung des Projekts

4. QNX Neutrino

5. Herausforderungen

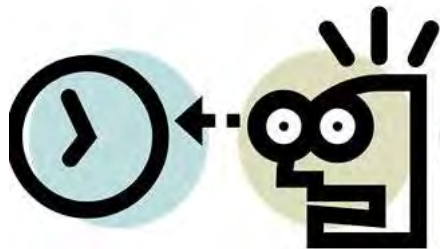
6. Erkenntnisse

7. Fazit

Wozu ein Betriebssystem?



- Steigende Anforderungen an die Produkte



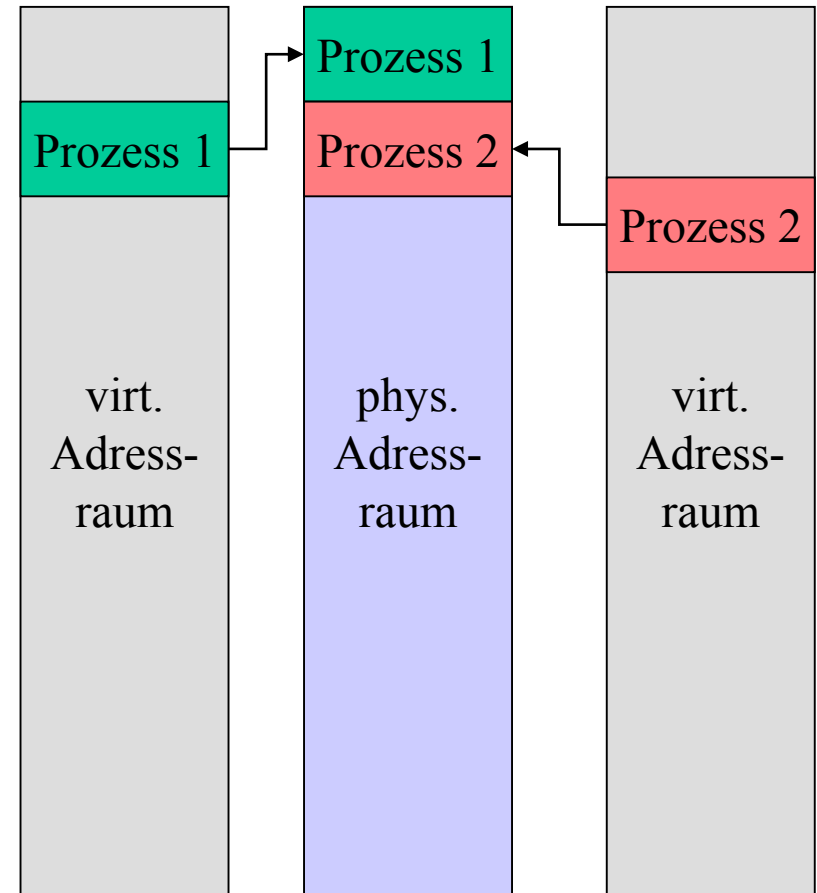
- Kürzere Entwicklungszyklen

Typische Vorteile eines Betriebssystems


- + Verwaltet die Ressourcen eines Computersystems, z.B.
 - + Rechenzeit
 - + Speicher
 - + Peripherie
- + Fertige Treiber für Peripherie (UART, USB, I²C, SPI, Ethernet, Display, Massenspeicher, Dateisysteme)
- + Stellt einheitliche Anwendungen - API zur Verfügung
 - + Leichtere Portierbarkeit existierender Software
 - + Fertige Softwarepakete (z.B. Windowing Toolkit, Datenbank)

Typische Vorteile eines Betriebssystems

- + Voneinander unabhängige Prozesse in unterschiedlichen Adressräumen (Segregation)
- + Entwicklung ohne Emulator/ICD möglich (nahezu wie Desktop-Applikation)
- + Konzentration der Entwicklung auf eigentliche Applikation



Typische Nachteile eines Betriebssystems

- Höhere HW-Anforderungen (Rechenleistung, Flash, RAM, MMU)
- 
- Nicht alle Betriebssysteme laufen auf jeder Hardwareplattform
- Einschränkungen bei der Entwicklung eigener Hardware

Agenda

1. Vor- und Nachteile eines Betriebssystems

2. Kriterien für die Betriebssystemauswahl

3. Kurzbeschreibung des Projekts

4. QNX Neutrino

5. Herausforderungen

6. Erkenntnisse

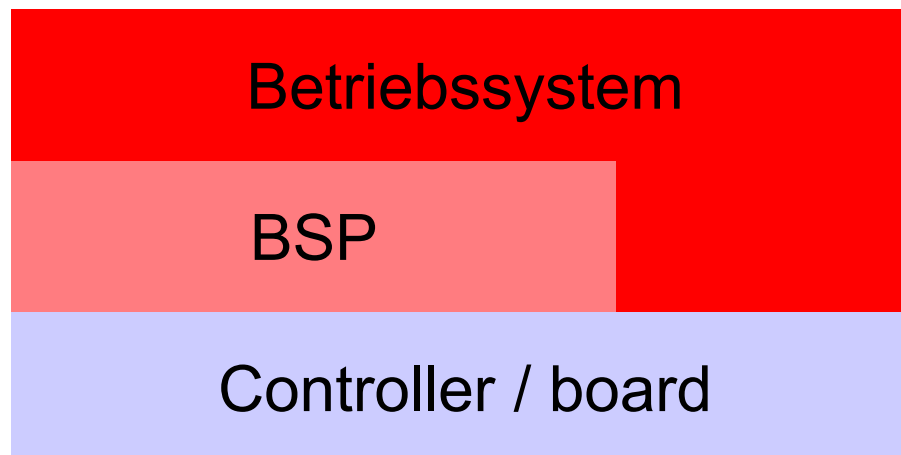
7. Fazit

Kriterien für die Betriebssystemauswahl

- Welche Zertifizierungen vorhanden? (Thema SOUP)
- Leistungsumfang (z.B. unabhängige Prozesse)
- Echtzeitfähigkeit (hart / weich)
- Kosten
 - Einmalige Kosten
 - Lizenzen pro Baugruppe (Royalties)
 - Entwicklerlizenzen
 - Lizenzgebühr für BSP
 - Schulungen

Kriterien für die Betriebssystemauswahl

- Board Support Package (BSP) für die gewünschte Plattform erhältlich?



Kriterien für die Betriebssystemauswahl

- Mögliche Quellen für ein BSP
 - Controllerhersteller (oft kostenlos)
 - Betriebssystemhersteller
 - Dritthersteller (typischerweise Dienstleister rund um ein bestimmtes Betriebssystem)
 - Auftragsentwicklung
- Know-how des Entwicklerteams
- Support



Ergebnis für konkretes Projekt

- → Wahl fiel auf QNX Neutrino
Hauptargument: Microkernel System, zertifiziert für sicherheitskritische Anwendungen
- Dazu passender Controller: i.MX35 von Freescale

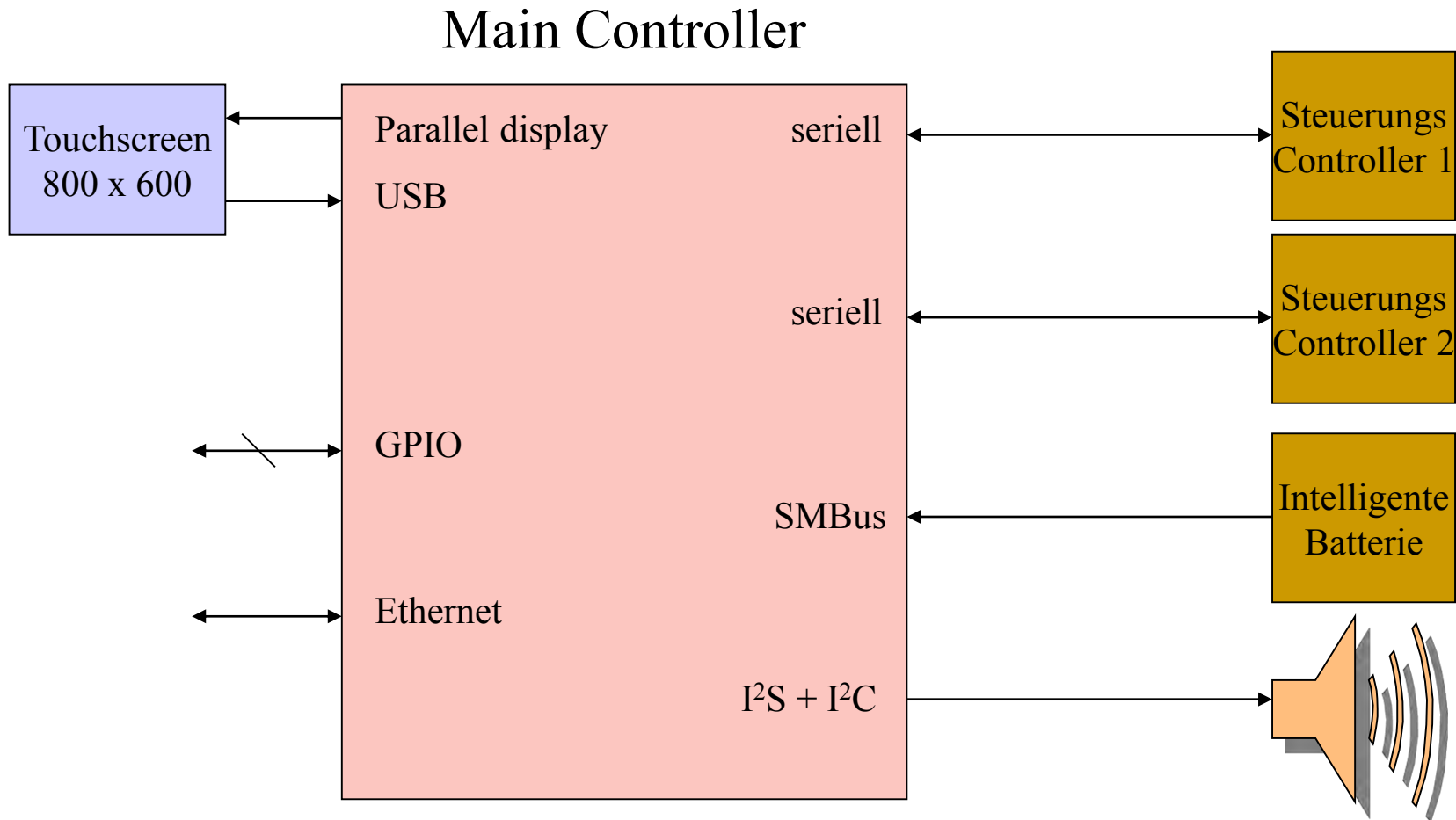
Agenda

1. Vor- und Nachteile eines Betriebssystems
2. Kriterien für die Betriebssystemauswahl
- 3. Kurzbeschreibung des Projekts**
4. QNX Neutrino
5. Herausforderungen
6. Erkenntnisse
7. Fazit

Kurzbeschreibung des Projektes: Anforderungen

- Sicherheit:
 - Klassifizierung Medizinprodukt in Europa CE IIb
 - Funkt. Sicherheitsanforderung gem. SIL 3 entspr. IEC 61508
 - SW Klassifizierung „C“ gem. IEC 62304
 - ‚Segregation‘ nach IEC62304: getrennte SW-Prozesse durch MMU und μ -Kernel geschützt
- Unterstützung für grafische Benutzeroberfläche (GUI)
- Prioritäten: Zuverlässigkeit und Sicherheit wichtiger als Stückkosten und Ressourcenverbrauch
- Geringe Entwicklungskosten und -Zeiten

Kurzbeschreibung des Projektes: Anforderungen



Kurzbeschreibung des Projektes: Ausgangssituation

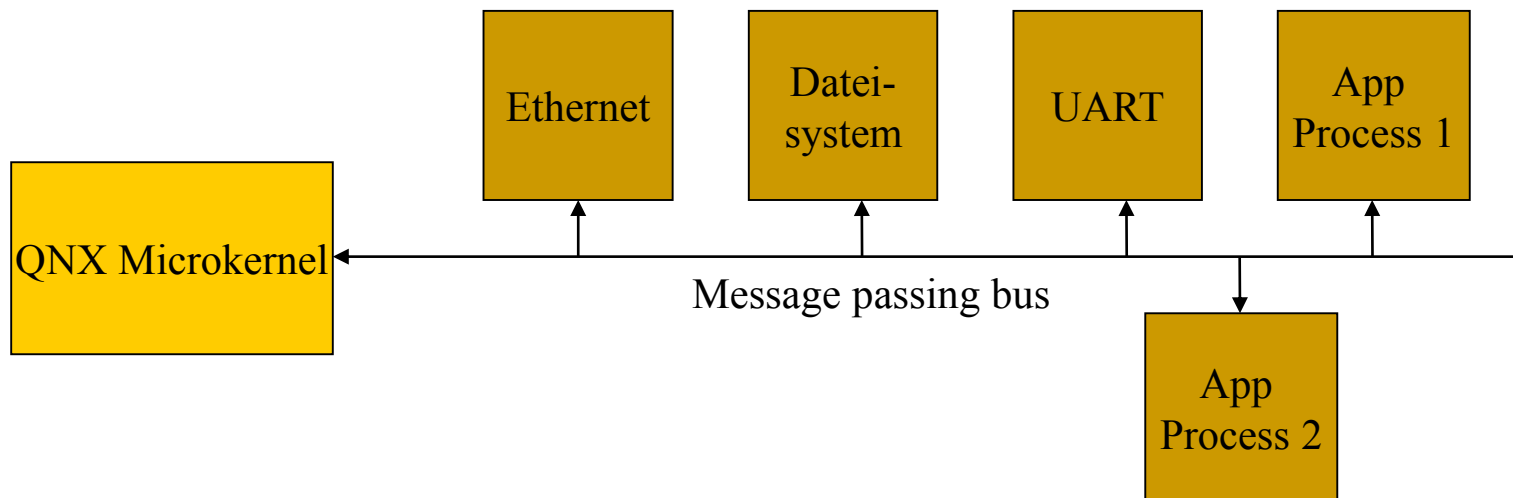
- Zu Projektbeginn war keine praktische Erfahrung mit QNX Neutrino beim Team vorhanden
- Erfahrung mit Embedded- und Desktop-Linux war vorhanden
- Viertägige QNX Schulung wurde durchgeführt
- Standortübergreifende Entwicklung geplant
 - GUI wurde von Kunden entwickelt
- Programmiersprache C++

Agenda

1. Vor- und Nachteile eines Betriebssystems
2. Kriterien für die Betriebssystemauswahl
3. Kurzbeschreibung des Projekts
- 4. QNX Neutrino**
5. Herausforderungen
6. Erkenntnisse
7. Fazit

QNX Neutrino

- Echtes Microkernel-Betriebssystem



QNX Neutrino

- IEC 61508 SIL3 zertifiziert
 - Auch in einer nicht-zertifizierten, preiswerteren Variante erhältlich
 - Sicherheitsaussagen über funktionale Anforderungen können direkt vom Hersteller übernommen werden.
- Erfüllt IEC 62304
- Offene Bugs werden von QNX im Rahmen von Bugreport Reviews und Auswirkungsanalysen verfolgt



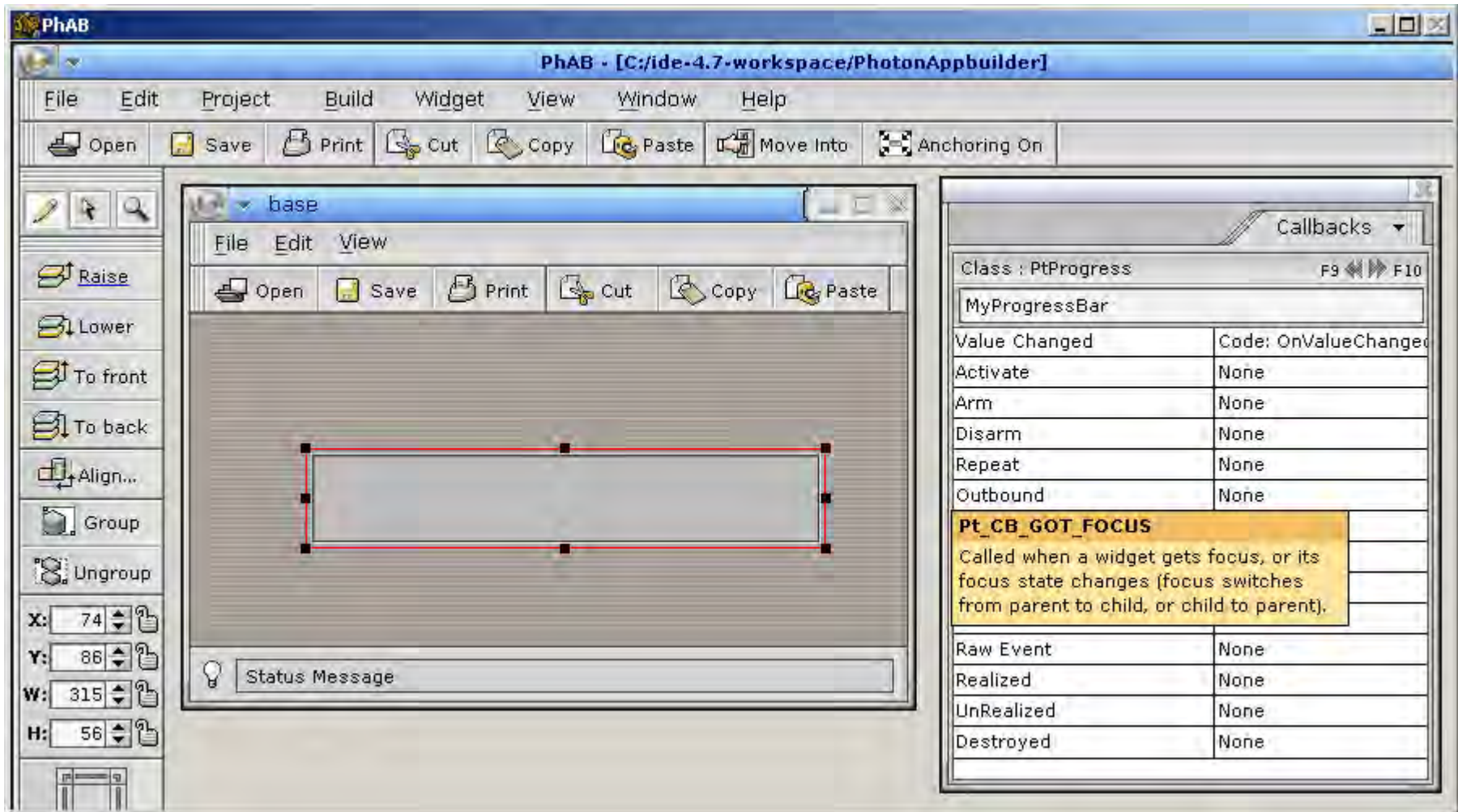
QNX Neutrino

- Unterstützt harte Echtzeit
- Sog. Unixoides Betriebssystem
- POSIX (Portable Operating System Interface) konform
- Volle C und C++ Unterstützung
- VMware Image im Internet kostenlos verfügbar
- Unterstützte Architekturen: Z.B. ARM, MIPS, PowerPC, SH-4, x86

QNX Photon microGUI

- Windowing tool kit
- C – Callback-Schnittstelle
- Screen builder tools, unterstützen drag and drop
- Auch Qt möglich, wurde aber nicht verwendet

QNX Photon microGUI screen builder



QNX Neutrino: Entwicklungsumgebung



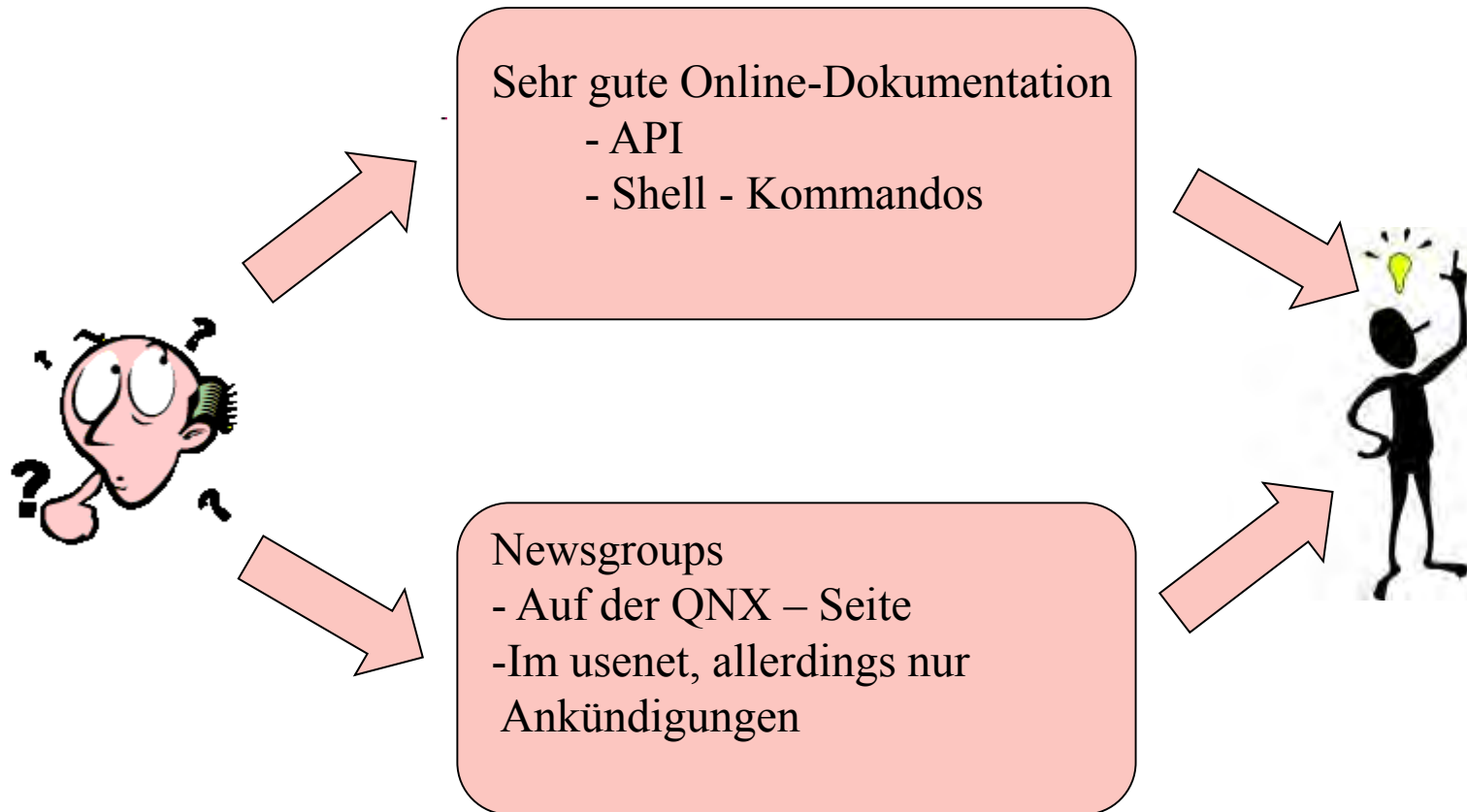
Eclipse-basierte IDE Momentics

- Eclipse plugins für zB. Versionskontrolle benutzbar
- 30-Tage Evaluierungslizenz möglich
- Entwickeln, Debuggen, Profilen über Host-PC
- Zugriff auf Target-Dateisystem über Drag and Drop



GNU Toolchain

QNX Neutrino: Dokumentation und Support

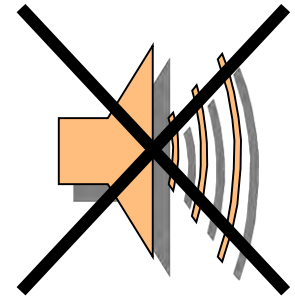


Agenda

1. Vor- und Nachteile eines Betriebssystems
2. Kriterien für die Betriebssystemauswahl
3. Kurzbeschreibung des Projekts
4. QNX Neutrino
- 5. Herausforderungen**
6. Erkenntnisse
7. Fazit

Problemstellung: Audio-Ausgabe

- Software für Audio-Ausgabe war vorhanden und auf EVA-Board lauffähig.
- Ausgabe der Samples erfolgte über I²S Port, welcher nicht durch SW Konfiguration zu ändern war.
- Auf der Baugruppe wurde ein anderer, nicht von der Software unterstützter Port verwendet.
- Durch Umprogrammierung eines im Controller enthaltenen Audio-Mux Funktionsblocks wurde das Problem ohne Hardware-Redesign gelöst.



Agenda

1. Vor- und Nachteile eines Betriebssystems
2. Kriterien für die Betriebssystemauswahl
3. Kurzbeschreibung des Projekts
4. QNX Neutrino
5. Herausforderungen
- 6. Erkenntnisse**
7. Fazit

Erkenntnisse

- Schwerpunkt Ressourcenverbrauch: GUI
- Vorgaben wurden eingehalten:
 - „Wir legen den Fokus auf die eigentliche Anwendung“
 - „Wir schreiben keine Treiber“
- Standortübergreifende Entwicklung gut möglich
 - Mechatronic: Applikation
 - Kunde: GUI

Erkenntnisse

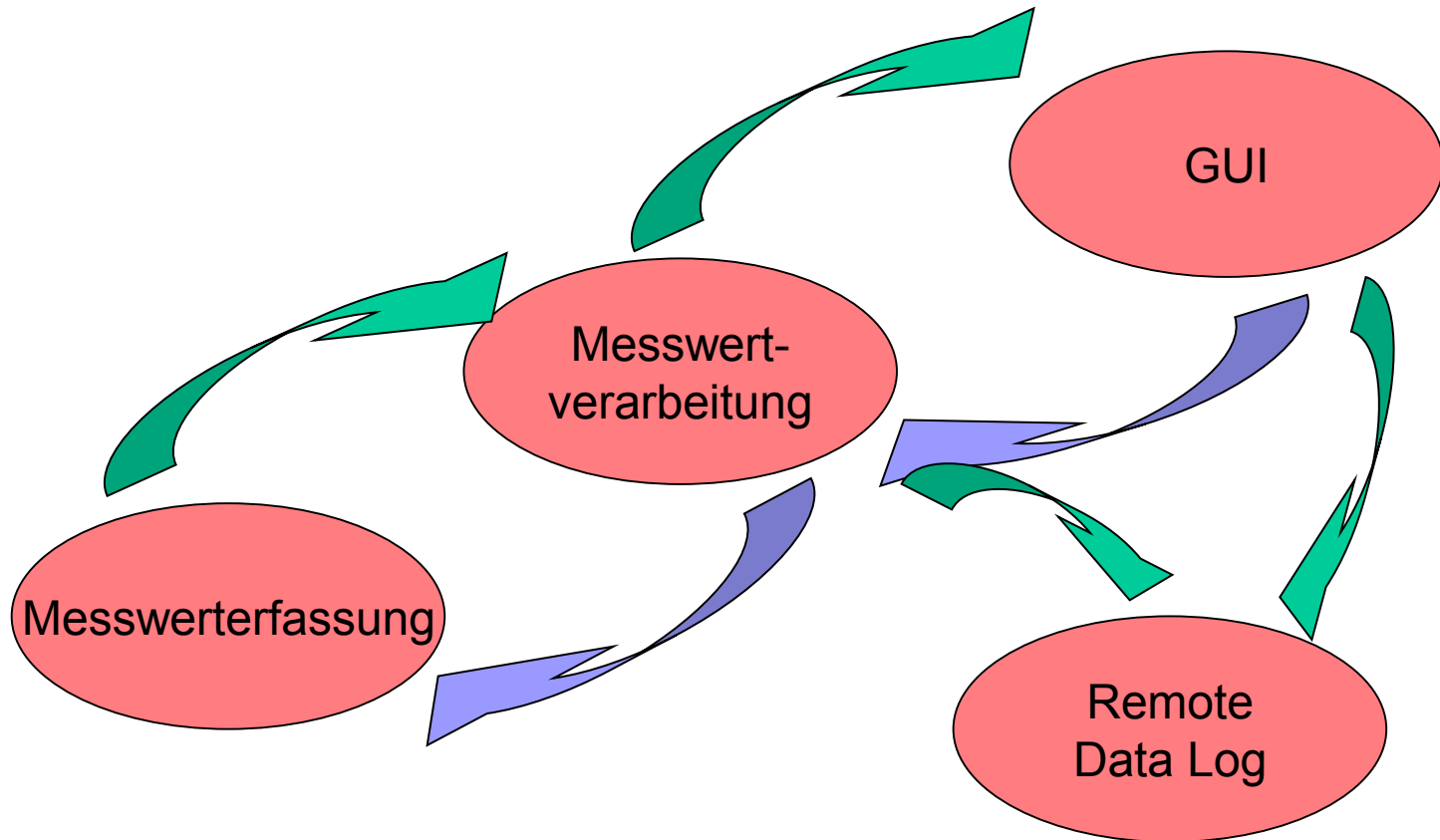
- BSP entscheidet welche Peripherie wie genutzt wird
 - Intensive Einbindung der SW-Entwicklungsabteilung bei Auswahl und Anbindung der HW-Komponenten notwendig.
→ Wo immer möglich, an EVA Board orientieren!
- Review der verwendeten GPIOs durch den BSP Hersteller sehr hilfreich

Erkenntnisse: Software von Drittanbietern

- Betriebssysteme bieten mehrere Verfahren zur IPC, u.a. POSIX konform. Nachteile:
 - Bei den meisten „klassischen“ Methoden muss der Sender alle Empfänger kennen
 - Änderungen an der Applikation sind vergleichsweise umfangreich
 - Keine Unterstützung zum Testen

Erkenntnisse: Software von Drittanbietern

- Inter-Prozess-Kommunikation

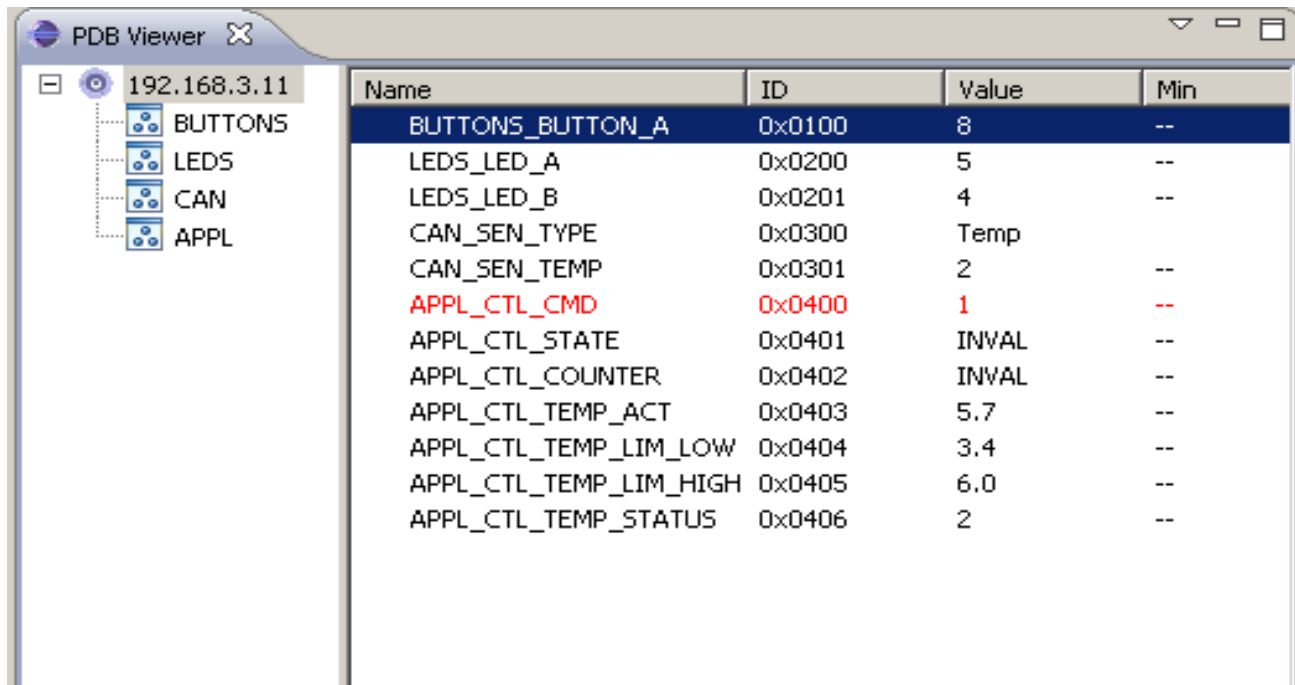


Erkenntnisse: Software von Drittanbietern: PDB

- „Process Data Board“ (eine Art Datenbank im RAM) hält alle Prozessdaten
- Empfänger können sich bei PDB für bestimmte Daten registrieren. Benachrichtigung erfolgt über Call-Back Funktion
- Sämtliche IPC durch PDB realisiert
- PDB ist eigener Prozess, dadurch geschützt („Segregation“)

Erkenntnisse: Software von Drittanbietern: PDB

- Inspektion des PDB Inhaltes zur Laufzeit über die IDE
- Änderung des PDB Inhaltes zur Laufzeit über die IDE möglich



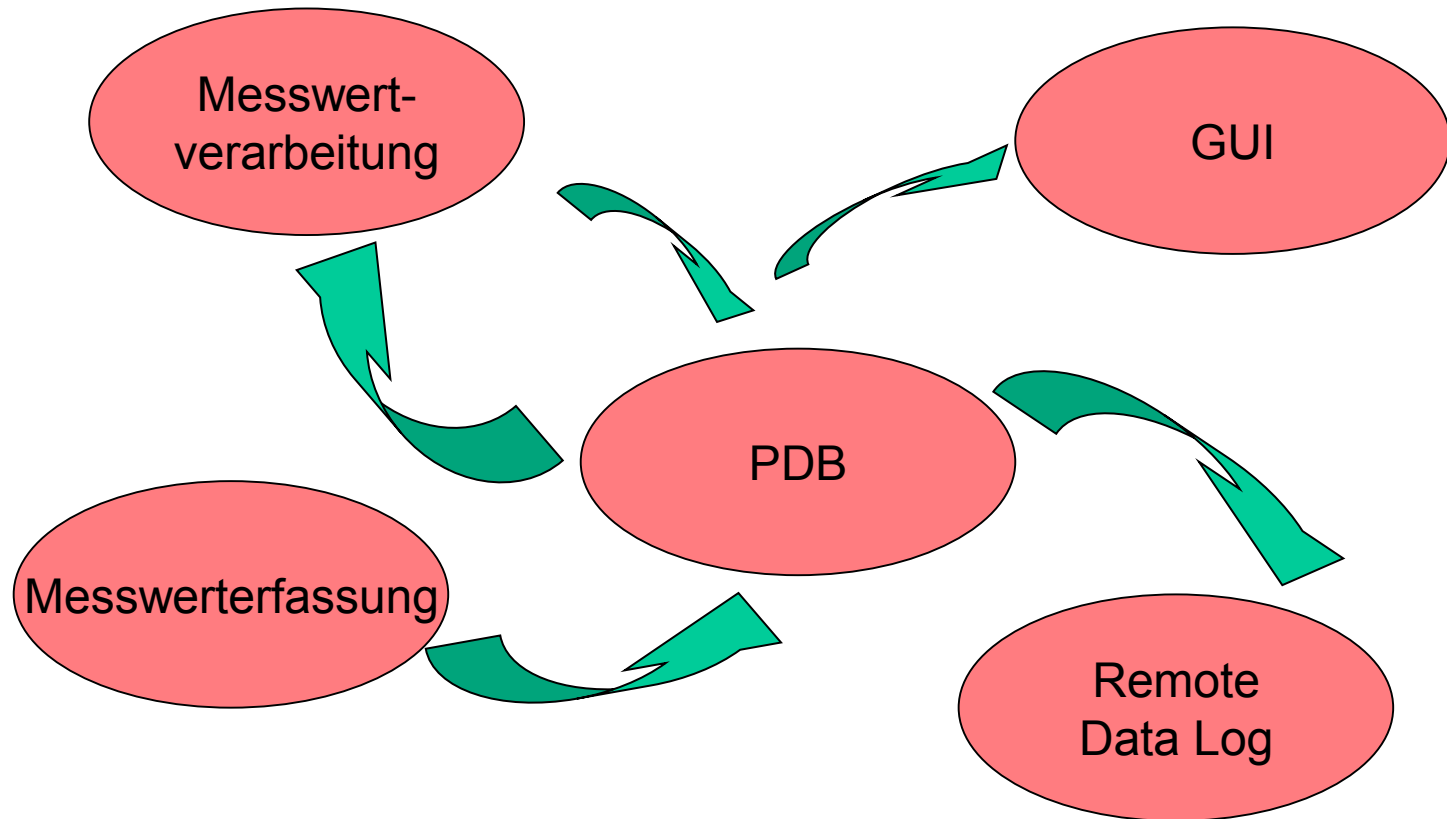
Name	ID	Value	Min
BUTTONS_BUTTON_A	0x0100	8	--
LEDS_LED_A	0x0200	5	--
LEDS_LED_B	0x0201	4	--
CAN_SEN_TYPE	0x0300	Temp	--
CAN_SEN_TEMP	0x0301	2	--
APPL_CTL_CMD	0x0400	1	--
APPL_CTL_STATE	0x0401	INVAL	--
APPL_CTL_COUNTER	0x0402	INVAL	--
APPL_CTL_TEMP_ACT	0x0403	5.7	--
APPL_CTL_TEMP_LIM_LOW	0x0404	3.4	--
APPL_CTL_TEMP_LIM_HIGH	0x0405	6.0	--
APPL_CTL_TEMP_STATUS	0x0406	2	--

Erkenntnisse: Software von Drittanbietern: PDB

- Ideale Unterstützung für
 - Modultests
 - Systemtests
 - Verteilte Entwicklung
- Erzwingt ein sauberes MVC (Model View Controller) Konzept
- Abstrahiert IPC

Erkenntnisse: Software von Drittanbietern

- Inter-Prozess-Kommunikation



Agenda

1. Vor- und Nachteile eines Betriebssystems
2. Kriterien für die Betriebssystemauswahl
3. Kurzbeschreibung des Projekts
4. QNX Neutrino
5. Herausforderungen
6. Erkenntnisse
- 7. Fazit**

Fazit

Bei komplexeren Projekten kommt man
um die Benutzung eines Betriebssystems kaum noch herum,
insbesondere wenn eine GUI verwendet wird.

Fazit

Hard- und Softwareentwicklung sind stärker verzahnt als eine
Entwicklung ohne Betriebssystem

Fazit

Nicht alle Team-Mitglieder müssen Erfahrung
im embedded-Bereich haben

Mechatronic

the medical engineers