

ENTWICKLUNGSTOOLS

Infotainment- und AUTOSAR-Konsolidierung auf einer ECU

Dieser Artikel zeigt, wie ein Hypervisor eine zuverlässige Lösung für die Konsolidierung von Infotainment- und AUTOSAR-Anwendungen auf demselben Steuergerät ermöglicht. Dies ist aufgrund der effizienten Leistung und der geringen Kosten für Portierungsanwendungen möglich.

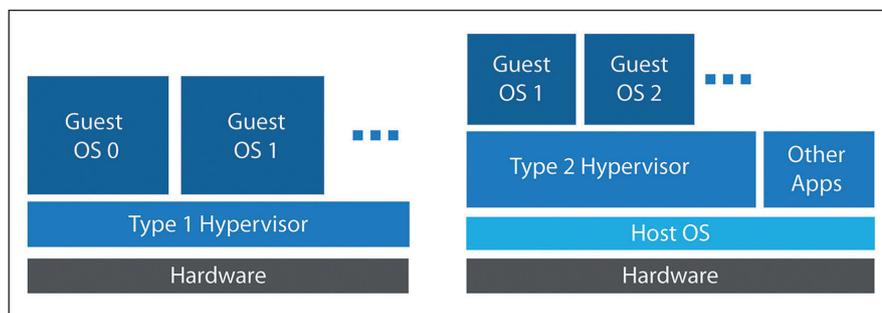


Bild 1: Es gibt zwei Typen von Hypervisoren, die häufig verwendet werden. © Siemens EDA

Neue und innovative Funktionen wie Benutzeroberflächen, Cloud-basierte Dienste, Ad-hoc-Netzwerke und autonomes Fahren erhöhen die Komplexität der Elektrik- und Elektronikarchitektur eines Fahrzeugs und stellen neue Anforderungen an vernetzte Softwaresysteme. Dies zwingt Automobilhersteller dazu, mehrere Geräte in einer einzigen High-Computing-Plattform zu konsolidieren.

Während dieser Ansatz zwar das Netzwerk des Fahrzeugs vereinfacht, stellt er die Architektur von Automobil-Softwaresystemen vor weiteren Herausforderungen in Bezug auf Sicherheit und Vernetzung. Für die fahrzeuginterne Kommunikation und sicherheitskritische Anforderungen sind häufig Echtzeitbetriebssysteme (RTOS) erforderlich, während Infotainment-Anwendungen, die nicht sicherheitsrelevant sind, auf gewöhnlichen Linux-Betriebssystemen ausgeführt werden. Ein Ansatz zur Lösung dieses Problems besteht darin, zwei Mikrocontroller auf demselben Steuergerät zu verwenden. Auf einer

MCU wird die Linux-basierte Anwendung ausgeführt, die für Aufgaben mit hohem Rechenaufwand wie beispielsweise KI-Algorithmen und Infotainment-Funktionen verantwortlich ist, während auf der anderen MCU einfache Echtzeitanwendungen zur Ausführung gebracht werden, die für die fahrzeuginterne Kommunikation verwendet werden. Beide MCUs sind dabei mit einer seriellen Peripherieschnittstelle verbunden, die die Kommunikation zwischen diesen beiden Anwendungen ermöglicht. Obwohl dieser Ansatz zwar den Einsatz

von Standard-Softwarearchitekturen ermöglicht, ist er ineffizient und recht kostspielig, da für jedes System eine eigene Hardware bereitgestellt werden muss. Ein anderer Ansatz besteht darin, Echtzeitanwendungen nach Linux zu portieren.

Siemens EDA (vorm. Mentor, A Siemens Business) bietet mehrere Produkte an, mit denen solche Probleme behoben werden können. Mit dem Mentor Embedded Multicore Framework beispielsweise können mehrere Betriebssysteme auf homogenen und heterogenen Multicore-Prozessoren bereitgestellt werden. Diese Lösung basiert auf dem OpenAMP-Standard, der von Mentor und Xilinx gemeinsam entwickelt wurde. Ein weiterer Ansatz ist der Einsatz des Mentor Embedded Hypervisor, der mithilfe von Virtualisierungstechniken mehrere Gast-Betriebssysteme auf demselben Prozessor konsolidiert.

Im Folgenden wird gezeigt, wie der Mentor Embedded Hypervisor eingesetzt werden kann, um eine Echtzeit-

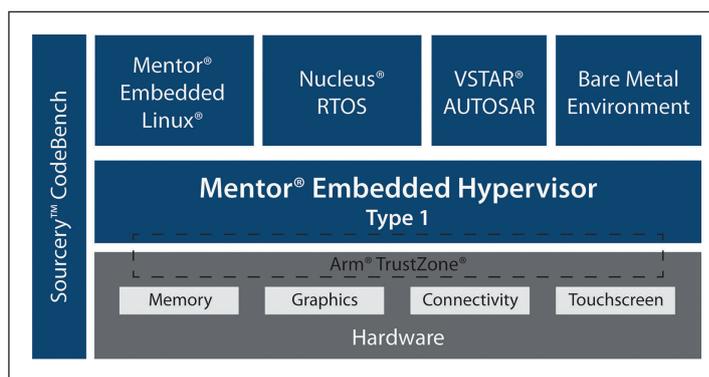


Bild 2: Der Embedded Hypervisor unterstützt mehrere Gast-Betriebssysteme wie Android, AUTOSAR, Mentor Embedded Linux und Nucleus RTOS.

© Siemens EDA

AUTOSAR-Anwendung mit einer Linux-basierten Anwendung zu konsolidieren.

Embedded Hypervisor

Es gibt zwei Typen von Hypervisors, die heute häufig verwendet werden (Bild 1):

- *Typ 1 Native Hypervisor:* Ein Hypervisor, der nativ auf Hardware ausgeführt wird, da er im Prinzip selbst als Betriebssystem fungiert.
- *Typ 2 Gehosteter Hypervisor:* Diese Art von Hypervisor muss von einem anderen Betriebssystem gehostet werden und ist nur für die Virtualisierung des Gast-Betriebssystems unter Verwendung der Ressourcen verantwortlich, die ihm vom Host-Betriebssystem zur Verfügung gestellt werden.

Der Embedded Hypervisor von Siemens EDA ist ein Typ-1-Hypervisor (Bild 2) und basiert auf einer Mikrokern-Architek-

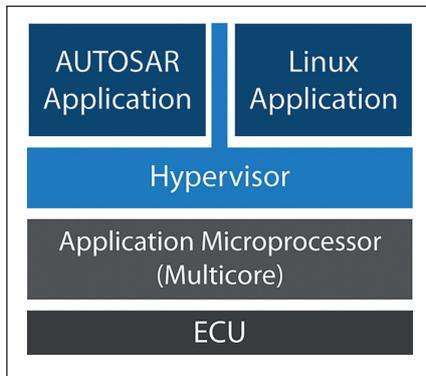


Bild 3: AUTOSAR-Demo-App © Siemens EDA

tegeräte und bietet damit mehr Flexibilität beim Systemdesign. Einige dieser Modelle umfassen:

- **Direkter Zugriff:** Weist Hardwaregeräte ausschließlich einer virtuellen Maschine zu, was direkten Zugriff ermöglicht. Dies ist vorteilhaft, wenn ein Hardwaregerät auf Systemebene so konzipiert ist, dass es

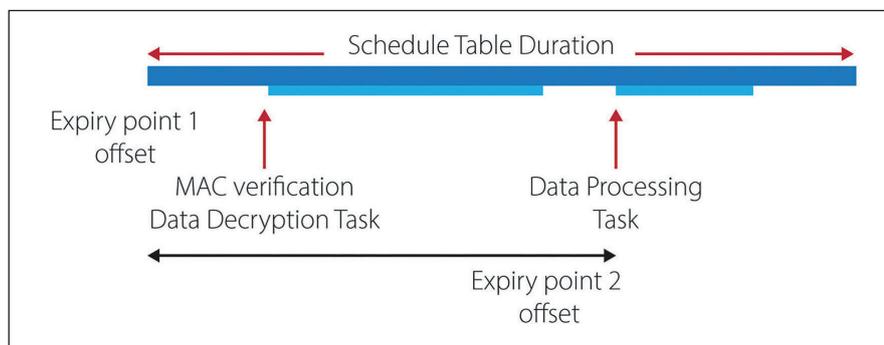


Bild 4: Scheduling und Taskübersicht © Siemens EDA

tur, bei der der Kern des Systems die grundlegenden Funktionen konsolidiert, die der Hypervisor für die Ausführung auf der Hardware benötigt.

Dieser Ansatz bietet auch den Vorteil, die kritischsten Teile des Systems für ein Höchstmaß an Sicherheit zu qualifizieren, da die Größe der vertrauenswürdigen Computerbasis im System verringert wird. Er minimiert auch den Speicherbedarf des Hypervisors entsprechend den Anforderungen der Gäste, z. B. nicht verwendete Funktionen, die dann so konfiguriert werden können, dass sie aus dem Hypervisor-Image entfernt werden.

Hardware-Gerätezugriff zwischen Guests

Der Embedded Hypervisor bietet mehrere Modelle für den Zugriff auf Periphe-

nur von einer virtuellen Maschine verwendet werden kann, da keine Hypervisor-Eingriffe erforderlich sind, um die Systemleistung zu verbessern.

- **Geteilter Zugriff:** Wenn ein Hardwaregerät einer virtuellen Maschine zugeordnet ist und erkannt wird, dass dieses Gerät gemeinsam genutzt wird, erfolgt die Behandlung auf der Ebene der virtuellen Maschine.
- **Emulierter Zugriff:** Ein gängiges Modell für die gemeinsame Nutzung von Hardwaregeräten zwischen virtuellen Maschinen, bei dem der Hypervisor den Gerätezugriff über Trap-and-Emulate-Techniken steuert, während jede virtuelle Maschine annimmt, dass ihr das Hardwaregerät zugeordnet ist.
- **Virtueller Zugriff:** Ähnlich wie beim

emulierten Zugriff, bei dem das Hardwaregerät dem Hypervisor zugeordnet ist, die virtuelle Maschine jedoch erkennt, dass ihr das Hardwaregerät nicht zugeordnet ist, und einen virtuellen Treiber für den Hardwarezugriff verwendet. Hier ist keine Emulation des Geräts erforderlich.

Virtualisierung

Die Verwendung von Virtualisierung in der Architektur von Automobilsoftware bietet einen besseren Ansatz, um diese Probleme zu lösen. Dies kann erreicht werden, indem verschiedene heterogene Automobilplattformen in virtuellen Maschinen (VM) gekapselt werden, die auf derselben Hardware ausgeführt werden. Dieser Ansatz bietet nicht nur eine effizientere Kommunikationsmethode durch Verwendung der Kommunikation zwischen VMs anstelle von seriellen Verbindungen, sondern reduziert auch die Kosten für das Hinzufügen eines dedizierten Mikrocontrollers zu jeder Plattform. Auch ältere Plattformen können als virtuelle Maschinen wiederverwendet werden, ohne dass neue Anpassungen erforderlich sind.

Die Kommunikation zwischen VMs kann mithilfe einfacher Zugriffstechniken für gemeinsam genutzten Speicher oder mithilfe strukturierterer Frameworks wie VirtIO erreicht werden. Obwohl AUTOSAR VirtIO nicht unterstützt, bietet es eine Spezifikation mit der Bezeichnung Complex Device Drives (CDDs), nach der nicht unterstützte Treiber wie VirtIO integriert werden können.

AUTOSAR Demo-App

Für die AUTOSAR-Demo-App bestand das Ziel darin, die strikten Echtzeitanforderungen für die Auslösung der Ablaufpunkte der Schedule Table zu erreichen. Die Schlüsselfaktoren für dieses Ziel in der virtualisierten Umgebung sind:

- IRQ-Routing zur virtuellen Maschine
- Generischer Interrupt-Controller-Zugriff
- Virtueller Timer-Zugriff

Bild 3 zeigt eine Übersicht über diese Demo-App. Die AUTOSAR-App wird als VM1 auf einem dedizierten Core ausgeführt. Die App nutzt die AUTOSAR-Kryptografiebibliothek, um zwei Vorgänge

auszuführen: MAC-Überprüfung und AES-Entschlüsselung verschlüsselter Daten (alle 100 ms). Eine andere App wird mit der Verarbeitung entschlüsselter Daten beauftragt. Dies wird mithilfe einer Schedule Table (Bild 4) organisiert. Die Linux-Anwendung wird als VM2 auf einem dedizierten Core ausgeführt und

```
*** MEHV Shell ***
MEHV:\>vminfo

VM ID=1
=====
Name:      vstar0
State:     Started
Boots:     1
VCPUs:     1
VCPU ID 0
  State:   Ready
  Role:    Primary

VM ID=2
=====
Name:      linux1
State:     Started
Boots:     1
VCPUs:     1
VCPU ID 1
  State:   Ready
  Role:    Primary
```

Bild 5: Wechseln zwischen Hypervisor- und Embedded Linux-Konsolen. © Siemens EDA

teilt die UART-Konsole mit dem Hypervisor, um den Wechsel zwischen der Hypervisor-Konsole und der Linux-Konsole zu ermöglichen, wobei letztere die Anbindung an das Linux-Dateisystem ermöglicht (Bild 5).

Laufzeitergebnisse der AUTOSAR-App

Es konnte bestätigt werden, dass bei Verwendung der AUTOSAR OS-Laufzeitmessung die Leistung der App beim Ausführen in einer virtualisierten Umgebung nicht beeinträchtigt wurde. Die Gesamt-CPU-Auslastung bei Ausführung in einer virtualisierten Umgebung stieg nur um circa 0,2%. Dies ist der Overhead, der durch IRQ-Routing und Hypervisor-Scheduler-Handling entsteht.

Da es für den Zugriff auf den virtuellen Timer oder die GIC-CPU-Schnittstelle keinen Overhead gab, wurden bei den Kernel-Ein- und Ausstiegszeiten für das AUTOSAR OS in einer virtualisierten Umgebung keine Änderungen vorgenommen. Die IRQ-Routing-Zeit war sehr kurz (561 ns für 1-GHz-Taktfrequenz), so dass die Latenz für den Timer-Interrupt des Betriebssystems nicht beeinflusst wurde.

Zusammenfassung

Gezeigt wurde, wie der Hypervisor von Siemens EDA eine zuverlässige Lösung für die Konsolidierung von Infotainment- und AUTOSAR-Anwendungen auf demselben Steuergerät ermöglicht.

Die Flexibilität bei der Konfiguration des Hypervisors ermöglicht es dem Anwender, den Eingriff des Hypervisors auf ein Minimum zu reduzieren, wodurch unterschiedliche Echtzeit-Anforderungen erfüllt werden.

Eine Standardisierung der Virtualisierung in der Architektur von Automobilssoftware wäre ein wichtiger Schritt zur Bereitstellung neuer Lösungen in heutigen Steuergeräten. Aus diesem Grund unterstützt Siemens EDA Hypervisor Standards wie VirtIO für virtuelle Geräte oder AUTOSAR für fahrzeuginterne Echtzeitanwendungen. ■ (oe)

www.mentor.com



Mohamed Mounir ist Embedded-Software-Ingenieur bei Siemens EDA und beschäftigt sich mit Virtualisierungskonzepten in Automobilsystemen.

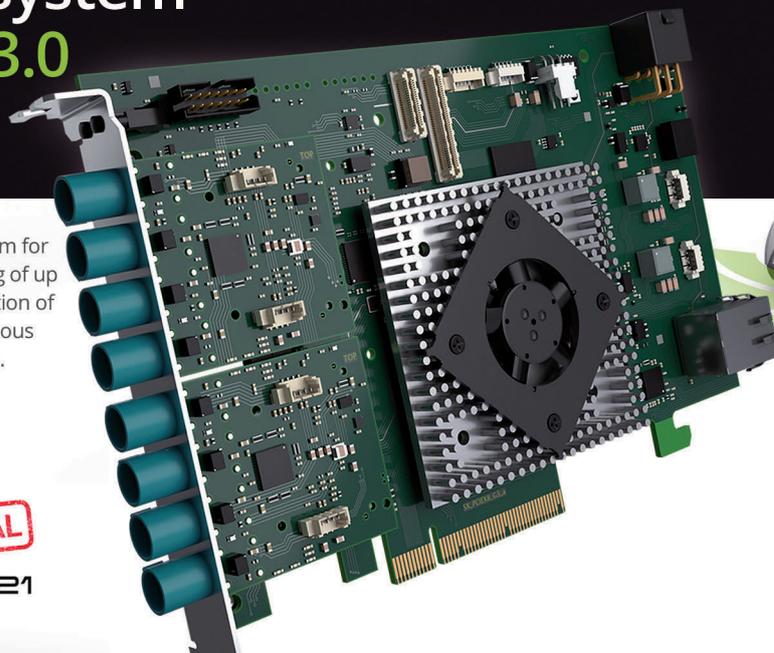
Modular video grabber and playback system SX proFRAME 3.0

SX proFRAME 3.0 - High-end real-time system for capture and replay of video data. Processing of up to eight 4K camera streams. Ideal for validation of ADAS and automated driving functions. Various camera adapters (GMSL, FPD-Link) available.

Get in touch with us: sales@solectrix.de

Visit us at...

DIGITAL
embeddedworld2021
Exhibition & Conference
... it's a smarter world



solectrix
high end electronics solutions

www.solectrix.de