

Asymmetric Multiprocessing (AMP) Runtime Framework

Das Konzept des Asymmetric Multiprocessing (AMP) bietet die Möglichkeit, auf CPUs mit mehreren Kernen einen Kern exklusiv für Aufgaben außerhalb des Linux-Systems zu nutzen. AMP-Lösungen für Echtzeit-Applikationen bieten durch ihren minimalen Overhead eine sehr geringe Latenz bei minimalem Jitter.

Im Rahmen eines mehrstufigen Boot-Prozesses können mit AMP-Konzepten auch Early-Action Applications (EAA) gestartet werden, um bestimmte Produkteigenschaften bereits vor dem vollständigen Start eines Linux-Systems verfügbar zu machen. Neben verbesserten Produkteigenschaften bieten AMP-Systeme eine gute Möglichkeit zur Konsolidierung von Hardware(-Kosten).

Verbessertes Start-up-Verhalten und Konsolidierung der Hardware-Kosten

Auf „seinem“ dedizierten CPU-Kern hat der Entwickler von Echtzeit-Software die Möglichkeit, eine Applikation mit voller Kontrolle über die Hardware und ohne Leistungsverlust durch ein Betriebssystem zu entwickeln. Die Programmierung des Threads auf dem separierten Kern erfolgt direkt auf der Hardware, also „bare-metal“ in Assembler oder C.

Über eine leistungsstarke Interprozessorkommunikation erfolgt der Datenaustausch mit dem Linux System. Auf den anderen Kernen des SoC stehen die uneingeschränkten Möglichkeiten eines industrietauglichen Embedded Linux-Systems zu Verfügung.

Features und Funktionen

Das emlix AMP Embedded Linux Runtime Framework bietet die folgenden Funktionen und Eigenschaften:

- Entwicklung von Software-Komponenten mit GNU Compiler (Linux Standard)
- Linux-Betriebssystemfunktionen für F&S Elektronik Systeme aStoneA9-V1-LIN SBC

- Messungen der Echtzeit-Eigenschaften mit Hilfe eines unabhängigen Stresstools unter Last
- Funktionen zur Kommunikation mit der Bare Metal-Applikation (Beispielapplikation)
- Kommunikation über Shared Memory und Interrupts mit eigenem Interface
- Steuerung von GPIO und serieller Schnittstelle (Beispielapplikation)
 - Bare Metal-Steuerung eines GPIO mit 100 kHz zur Reduktion des Jitter
 - Bare Metal-Steuerung eines weiteren GPIO mit variabler Frequenz
 - Bare Metal-Abfrage weiterer GPIOs als Tastatur
 - Status-Informationsausgabe über die serielle Schnittstelle
- Funktionen zum Laden und Starten der Bare Metal Software
- Dokumentation der Software-Komponenten und Übersetzungsprozesse

Mit dem AMP Embedded Linux Runtime Framework für Embedded Module mit i.MX 6 Quad CPU bietet emlix eine Ready to Develop-Lösung für Entwickler, die sofort starten und keinen Aufwand in die technisch komplexe und

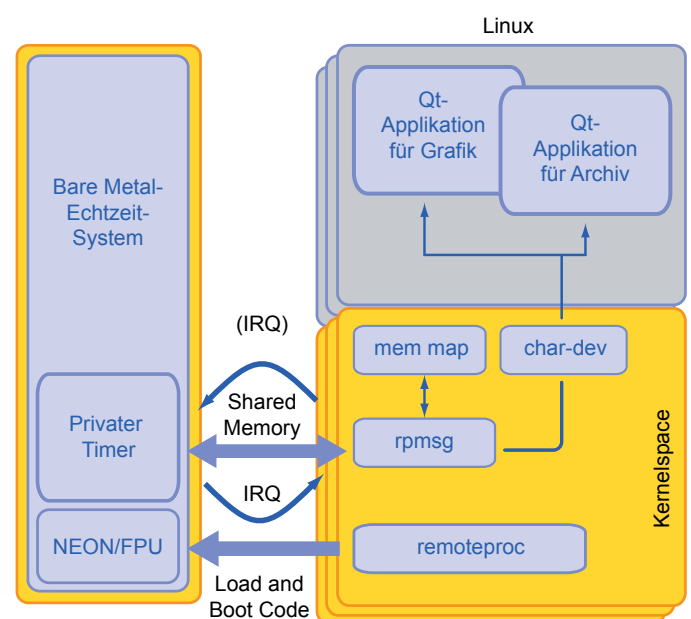


Bild 1: Aufteilung von vier ARM-Cortex A Kernen

aufwändige Entwicklung einer eigenen Multicore-Infrastruktur für Bare Metal-Applikationen investieren möchten.

Komponenten

Basierend auf einem – exemplarisch für das F&S aStoneA9-V1-LIN angepassten – Embedded Linux BSP mit Interfaces zum Laden und Starten von Software auf einem dedizierten CPU-Kern kann der Anwender ein bereits fertig implementiertes Shared-Memory-Interface und Interrupts zur Kommunikation zwischen Bare Metal -Applikationen und Linux System nutzen.

Eine Beispielapplikation zur Ansteuerung von Hardware zeigt die Funktionsweise des Systems und kann als Grundlage für eigene Applikationen dienen. Die folgenden Komponenten sind Bestandteil des emlix AMP Runtime Frameworks:

- Embedded Linux System als AMP-Master
- Linux-basiertes Tooling zur Steuerung des Echtzeit-CPU-Kerns
- Beispielhafte Bare-Metal-Applikation
 - CPU + MMU Initialisierung etc.
 - Ansteuerung von GPIOs (in/out)
 - Nutzung einer seriellen Schnittstelle
 - Shared Memory Kommunikation
- Beispielhafte Linux-Applikation als Gegenstelle zur Kommunikation mit der Bare-Metal-Software
- Toolchain zur Übersetzung aller Software-Komponenten

Services

Im Rahmen unserer ergänzenden Services erfolgt die Übergabe und die Einweisung in die Handhabung des Frameworks. Zielsetzung ist, dass Mitarbeiter des Kunden eigenständig Applikationen auf Basis des AMP Frameworks entwickeln können. Darüber hinaus bieten wir ein Support-Kontingent und Updates. Selbstverständlich übernehmen wir auf Wunsch die Portierung des Frameworks auf Prozessormodule anderer Hersteller sowie kundenspezifische Hardware.

AMP für Early-Action Application (EAA)

Das Konzept des AMP bietet neben harter Echtzeit die Möglichkeit, dass unmittelbar nach dem Power-Up eine Early-Action Application (EAA) in einem abgetrennten Kern gestartet wird. Diese Technik erlaubt es, kurz nach dem Einschalten und deutlich vor dem vollständigen Start des Linux-Systems anwendungsspezifische Aufgaben auszuführen.

Extrem verkürzte Start-up-Zeiten für dedizierte Applikationen

Die EAA läuft direkt auf dem dedizierten CPU-Kern ohne Nutzung eines Betriebssystems oder umfangreicher Libraries, also bare metal. Dieser unmittelbare Betrieb erlaubt eine minimale Initialisierungsphase, sodass die Funktionen dieser Anwendung innerhalb weniger Millisekunden bereitstehen. Die getrennten Systeme können über shared memory und Inter Processor Interrupts (IPI) Informationen austauschen. Folgende Szenarien für die Nutzung von EAA bieten sich an:

- Datenpuffer für später verfügbare Linux-Applikationen
- Kommunikation mit der Außenwelt beispielsweise über CAN
- Kombination Early-Action Datenpuffer und Kommunikation
- Asynchrones Ausgeben von Wartepaketen auf einem Bus
- Asynchrones Ausgeben von Grafik (animierter Boot-Screen)
- Ausführung von Legacy Software für Mikrocontroller

emlix GmbH

solutions @ emlix.com

<http://www.emlix.com>

Phone +49 (0) 551 / 30664-0

Fax +49 (0) 551 / 30664-11