PXROS-HR - das Sicherheits-Framework für embedded Systeme

Horst Lehser

HighTec EDV-Systeme GmbH
Feldmannstr. 98
D-66119 Saarbrücken
horst.lehser@hightec-rt.com

Zusammenfassung

Nach der Philosophie von PXROS leben Tasks in unabhängigen und isolierten Adressräume und kommunizieren nur über “Ereignisse” oder durch Übergabe von Objekten. Task werden als ladbare Einheiten verstanden, die keinen statischen Linkzusammenhang untereinander haben. Ladbar meint hier, dass Code und Daten unabhängig von der Adresslage sind und damit der Testaufwand auf ein Minimum reduziert wird. Das innovative an PXROS-HR ist, dass der Schutz nicht softwarebasiert implementiert ist, sondern die Memory Protection Unit (MPU) des TriCore verwendet wird – so als liefe jede Komponente (Kapsel) auf einem eigenen Controller. Diese Vorgehensweise sichert das eine Kapsel niemals die Integrität des Gesamtsystems gefährden kann.

Typische Sicherheitsanforderung sind: Der Nachweis der Korrektheit von Software, die Verfügbarkeit zu garantieren und ein robustes und testbares System zu haben. Ein Anwendung mit diesen Anforderungen hat das Ziel das Risiko für Mensch und Maschine auf ein vorgegebenes Maß zu beschränken. Das Risiko ist definiert als Produkt aus Fehlerwahrscheinlichkeit und dem Schadensausmaß. Der Speicherschutz verhindert eine Fehlerausbreitung und deshalb ist das Schadensausmaß gleich Null, so dass eine Fehlfunktion einer Komponente, die nicht selbst sicherheitsrelevante Funktionen enthält, nicht das Sicherheitsrisiko beeinflusst. Eine Fehlfunktion, die durch einen Softwarefehler verursacht wurde, wird von der Hardware (MPU) erkannt und zur Laufzeit behandelt. Dies ist wichtig für die Produkthaftung, da es nun möglich wird zu erkennen, wer einen Fehler verursacht hat. Somit können auch Software von Drittanbieter oder neue Funktionalität ohne Risiko einfach integriert werden. Softwarebestandteile können als wiederverwendbare Einheiten angesehen werden, die im Baukastenprinzip zusammengesetzt werden. Ein weiterer Vorteil von PXROS-HR ist, dass durch den Speicherschutz eine Kapselung realisiert wird, die eine Rückwirkungsfreiheit von Komponenten bedeutet. Dies ist ein entscheidender Vorteil, da dies die Komplexität vermindert und modulares Testen ermöglicht.

Eigenschaften der Plattform: Durch die Kapselung können bei gleichen Sicherheitsanforderungen redundante Systeme in Software realisiert werden. Die Kapselung ermöglicht auch die Umsetzung von shared library Konzepten und es kann sogar Open Software eingebettet werden, ohne dass die restliche Software und diese Lizenz fällt. Die Kommunikation in der Plattform geschieht durch geschützte Objekte (beliebige Granularität des Speicherschutzes), was effizient und sicher ist. Durch die unterschiedlichen Privilegstufen im TriCore lässt sich auch die Peripherie vor ungewollten Zugriffen schützen.

Effizienz: Die Kommunikation basiert auf der Übergabe eines Objektes ohne das Kopieren von Daten. Ein Objekt, das übergeben wird, verlässt den Adressraum des Senders und tritt in den Adressraum des Empfängers ein. Das Objekt kann eine beliebige Größe haben und somit ist der Speicherverbrauch minimal im Gegensatz zu Ansätzen, die den Schutz durch Mapping realisieren. Die Speichereffizienz ist entscheidend, da in Embedded Anwendungen typischerweise nur wenig RAM zur Verfügung steht. Ein weitere besondere Eigenschaft von PXROS-HR ist, die sogenannte interrupt transparency. D.h. der Kernel sperrt niemals Interrupts und bewahrt somit die Interruptfähigkeit des Mikrocontroller, was essentiell für Anwendungen ist, die nur einen minimalen Jitter zulassen.

Testbarkeit: In PXROS-HR können Komponenten dynamisch nachgeladen und in einem laufenden System gedebugged werden, ohne den darunter liegenden Kern anzuhalten. Die Kapselung ermöglicht auch das gleichzeitige Debuggen von mehreren Komponenten.

Migration: Durch die Virtualisierung in PXROS-HR kann eine Kapsel einen Software ohne Betriebssystem oder andere verwendeten Betriebssystemen enthalten. Auch die I/O kann virtualisiert werden. Diese Eigenschaften erhöhen die Portierbarkeit von Software. Des weiteren kann Software in einem Migrationsprozess in eine große Kapsel gesteckt werden und schrittweise in kleinere Kapsel gesteckt werden, bis der gewünschte Grad an Modularisierung erreicht wurde.

Kosten: Die strikte Kapselung aller Komponenten erlaubt sicherheitsrelevante Funktionalität zu zertifizieren und sogar neue Funktionalität, die nicht sicherheitsrelevant ist, hinzuzufügen ohne das bestehende Zertifikat zu verletzt. Dies geht nur, da die MPU die Rückwirkungsfreiheit garantiert.