

# Betriebssoftware für Preon32-Serie

## Merkmale

- Anwendungen in Java™ nutzbar
- Rechenoperationen bis 64-Bit (long, double)
- Unterstützung von Threads, Synchronisierung mit Monitoren
- Garbage Collection mit Speicherdefragmentierung
- Exception-Behandlung (auch RuntimeException)
- System-Properties zur Konfigurierung von Anwendungen



## Beschreibung

Die PreonVM ist eine virtuelle Maschine (VM) von VIRTENIO für den Einsatz in eingebetteten Systemen mit sehr geringen Ressourcen wie dem Funkmodul Preon32 von VIRTENIO. Die virtuelle Maschine ist hoch optimiert, benötigt kein zusätzliches Betriebssystem und wird direkt auf dem verwendeten Mikrocontroller betrieben. Der Ansatz sieht vor, dass Anwendungen in Java™ über einen Transformationsprozess für das Funkmodul Preon32 nutzbar werden. Im diesem Zusammenhang bietet VIRTENIO entsprechende Software-Bibliotheken für verschiedene Schnittstellen an. Damit wird die Programmierung von Anwendungen, die Sensordaten erfassen und Aktoren steuern, elegant möglich. Darüber hinaus sind die Funkschnittstelle für IEEE 802.15.4 und eine AES-Verschlüsselung in Hardware als innovative Bestandteile genannt, die in der virtuellen Maschine einfach nutzbar sind.

## Produktvorteile

Eingebettete Systeme mit minimalen Ressourcen wurden bisher in speziellen und wenig komfortablen Sprachen programmiert. Komplexe Anwendungen sind deshalb zeitaufwendig und mit einem hohen Fehlerpotential umzusetzen. Der Ansatz von VIRTENIO bringt den Komfort, die Sprachkomplexität und die Sicherheit einer virtuellen Maschine in Kombination mit einer objektorientierten Programmiersprache in die Welt der eingebetteten Systeme. Aktuell bietet VIRTENIO für sein Funkmodul Preon32 für drahtlose Netzwerke nach IEEE 802.15.4 die virtuelle Maschine als innovative Betriebssoftware an.

Durch die Verwendung einer virtuellen Maschine wird das Anwendungsprogramm für das Funkmodul unabhängig von der tatsächlich verwendeten Architektur. Das hat den Vorteil, dass Anwendungsprogramme unverändert und ohne einen manuellen Anpassungsprozess ausgeführt werden können, auch wenn bspw. der Prozessor oder Funktransceiver durch einen anderen Typ ausgetauscht werden.

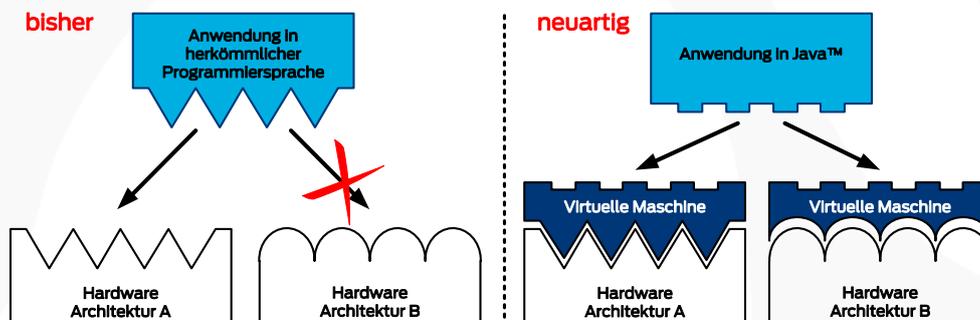


Abb. 1 Prinzip und Vorteil einer virtuellen Maschine wie Plattformunabhängigkeit der Anwendung

Gleichzeitig werden der Entwicklungsprozess der Hardware und des Anwendungsprogramms entkoppelt. Das Anwendungsprogramm wird gegen eine virtuelle Architektur mit entsprechenden Schnittstellen entwickelt, die über die Zeit unverändert bleibt.

Zudem wurde die virtuelle Maschine derart optimiert, dass Anwendungen auf einem 8/32-Bit- Mikrocontroller ab 8 KByte RAM und ab 128 KByte Flash performant ausgeführt werden können. Die komplette virtuelle Maschine mit Anbindung der Peripherie wie dem Funk-Transceiver nach IEEE 802.15.4 hat daher einen Speicherplatzbedarf von weniger als 60KByte Flash.

## Anwendungen für virtuelle Maschine vorbereiten

Auf dem Funkmodul Preon32 wird die von VIRTENIO entwickelte VM als Betriebssoftware eingesetzt. Mit dieser Betriebssoftware ist es möglich, dass Anwendungen, die als Quellcode in Java™ geschrieben sind, genutzt werden können, nachdem sie den unten abgebildeten Transformationsprozess durchlaufen haben.

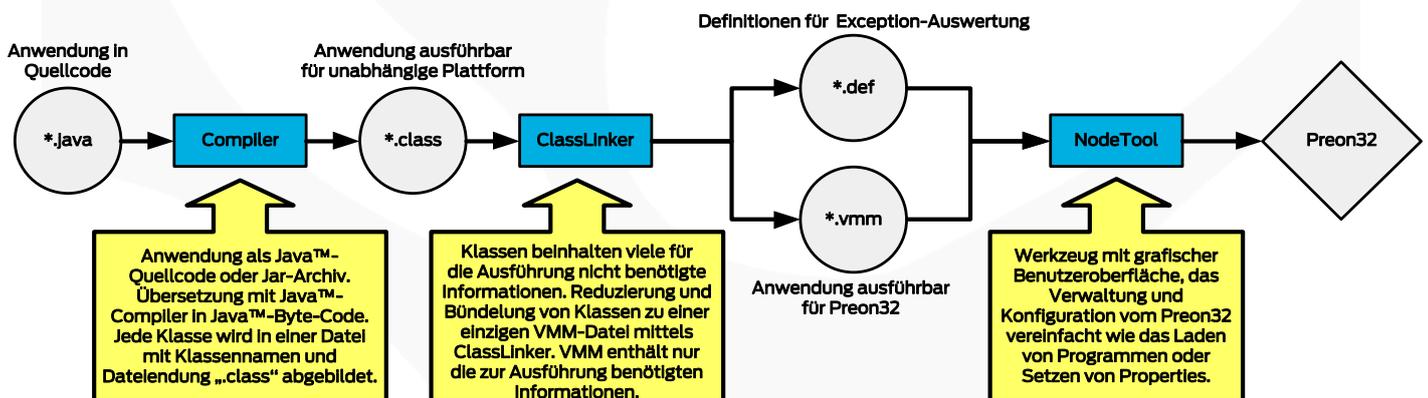


Abb. 2 Transformationsprozess zur Nutzung von Anwendungen in Java™ auf dem Funkmodule Preon32

## Merkmale der PreonVM

- Verwendung auf 8/32-Bit-Mikrocontroller
- Benötigt weniger als 60KByte ROM/Flash
- kein zusätzliches Betriebssystem ist notwendig (Bare-Metal-Ansatz)
- Unterstützt Datentypen von `char`, `byte` bis `long`, `float` und `double`
- Garbage Collector inkl. Defragmentierung des Arbeitsspeichers
- Unterstützt Threads inklusive Synchronisierung über `synchronized`, `Object.wait`, `Object.notify`, `Object.notifyAll`, `Thread.sleep`, `Thread.interrupt`
- Unterstützt Exceptions, Stacktraces, mehrdimensionale Arrays, dynamische Typüberprüfung
- Unterstützung eines schnellen Native Method Interfaces
- Unterstützt Event-Konzept für synchrone und asynchrone Events
- Verwendet vorgelinkte Java™-Klassen (Module) mit geringer Programmgröße und besserer Geschwindigkeit auf dem Zielsystem
- Einfache Portierbarkeit auf andere Plattformen durch geringe Hardware-Abhängigkeit
- Verschiedene Stack-Modelle für Systeme mit wenig und mehr Ressourcen

## ClassLinker Module

- ClassLinker von VIRTENIO transformiert und linkt Java™-Klassen zu Modulen (.vmm)
- Module können andere Module importieren und als Bibliothek oder als ausführbares Programm genutzt werden
- Module benötigen nur ca. 25% der Gesamtgröße aller ihrer enthaltenen Klassen

## Klassen- und Treiberbibliothek

- Laufzeitbibliothek erfordert 20-30kB ROM/Flash
- Hohe Kompatibilität durch viele Standardklassen
- Treiberbibliothek zur Verwendung interner und externer Schnittstellen sowie Peripherie wie Sensoren und Aktoren

## Zielsysteme

- $\geq 128k$  ROM,  $\geq 8k$  RAM, Mikrocontroller mit 8-, 16-, oder 32-Bit