

**Zeitschrift:** Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association Suisse des Electriciens, de l'Association des Entreprises électriques suisses

**Band:** 79 (1988)

**Heft:** 17

**Artikel:** Echtzeit-Software für Mikrocomputer

**Autor:** Huguenin, F.

**DOI:** <https://doi.org/10.5169/seals-904074>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. [Siehe Rechtliche Hinweise.](#)

### **Conditions d'utilisation**

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. [Voir Informations légales.](#)

### **Terms of use**

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. [See Legal notice.](#)

**Download PDF:** 23.11.2024

**ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>**

# Echtzeit-Software für Mikrocomputer

F. Huguenin

**Die heutigen Mikrocomputeranwendungen werden immer komplexer. Es müssen deshalb neue Wege gesucht werden, um die grossen Software-Investitionen über mehrere Geräte-Generationen hinweg abzusichern. Der folgende Aufsatz will einen Überblick über die diesbezüglichen aktuellen Probleme geben und die verfügbaren Echtzeitbetriebssysteme für 16- und 32-Bit-Mikroprozessoren von Intel kurz darstellen.**

**L'utilisation généralisée des microprocesseurs force l'utilisateur à repenser les méthodes de base concernant la protection de l'investissement logiciel à long terme. L'article suivant présente un résumé des problèmes actuels et discute la palette de logiciel temps réel d'Intel pour les microprocesseurs 16 et 32 bit.**

Es ist eine Tatsache, dass heutzutage die Mikrocomputeranwendungen immer mehr Software benötigen. Eine Folge davon ist, dass nicht nur die Ingenieure in den Entwicklungsabteilungen, sondern bis zu einem gewissen Masse auch die Entscheidungsträger sich mit der sogenannten Systemsoftware befassen müssen. Die Software-Krise (Man findet keine Leute! Alles dauert zu lange, und es kostet viel zu viel!) zwingt langsam, aber sicher zu einem Umdenken und führt beinahe zwangsläufig zur Verwendung von eingekauften Systemsoftwarepaketen.

Die Zeiten, da der Entwicklungsingenieur je nach Anwendung eine eigene Echtzeit-Executive geschrieben hat, ist vorbei. Heute kauft man diese Software so einfach ein wie einen VLSI-Baustein, was sowohl für den Anwender als auch für die Entscheidungsträger in den Firmen etliche Vorteile einbringt:

- *Die Standardisierung* der Systemsoftware innerhalb der Firma minimiert die Zeit- und Schulungsverluste sowie alle damit verbundenen Probleme des Personalwechsels.
- *Die Wiederverwendbarkeit* der Systemsoftware führt dazu, dass verschiedene Endprodukte die genau gleiche Basissoftware besitzen. Dadurch werden die Wartungskosten klein gehalten.
- *Die Flexibilität* erlaubt dem Anwender, die Systemsoftware entsprechend seiner Applikation zu konfigurieren.
- *Die Dokumentation* lässt sich durch Konventionen in der Verwendung der Systemsoftware wesentlich verbessern, da diese immer gleich bleibt.
- *Die Kompatibilität* wird über verschiedene Generationen von Mikroprozessoren gewährleistet, da die verwendete Systemsoftware über Gerätegenerationen hinweg beibehalten wird.

Der Einkauf von Standard-Software bringt aber auch einige Nachteile:

- *Der Preis* von Systemsoftware ist nicht unbedeutend, wobei meist noch mit Implementationsgebühren zu rechnen ist.
- *Die Komplexität* kann manche Entwicklungsingenieure davon abhalten, diese Softwarepakete zu benutzen.
- *Die Anwendung* einer gekauften Systemsoftware erfordert ein Umdenken in vielen Belangen. So ist z.B. die Systemsoftware für alle Beteiligten an einem Projekt transparent. Es können deshalb in einer Entwicklungsabteilung nicht mehr so einfach «Software-Königreiche» entstehen.
- *Der Speicherbedarf* ist naturgemäss höher, was sich auf die Hardwarekosten auswirkt.

*Aber sind das echte Nachteile?* Man bedenke, wie hoch die Kosten für Entwicklungsingenieure heute sind! Kostet nicht die Programmierung einer kleinen Echtzeit-Executive inklusive Testen und Validieren ein Vielfaches von dem eines eingekauften Produkts? Dieser Aufsatz will einen kurzen Überblick über die verfügbaren Echtzeitbetriebssysteme für Intel-Mikroprozessoren geben und dabei auf einige technische Merkmale eingehen.

## Realtime-Betriebssystem iRMX 86

Schon sehr früh in der Mikrocomputer-Geschichte (70er Jahre) wurde an der Entwicklung eines effizienten Betriebssystems für Echtzeitanwendungen gearbeitet. Damals benutzte man aber meist selbstgeschriebene Software, mit der man die Mikrocomputer der 8085er-Klasse für Realtimeanwendungen «zurechtbog». Intel führte Ende der 70er Jahre das erste Echtzeit-Betriebssystem iRMX 80

### Adresse des Autors

François Huguenin, Intel Semiconductor AG, Talackerstrasse 17, 8065 Zürich.

(Real Time Multitasking Executive)

auf dem Markt ein. Dies war eine kleine Revolution, erlaubte jenes doch bereits, recht komplexe Echtzeitanwendungen zu realisieren. Auf dieser Linie erfolgte wenig später (1980) die Einführung des Echtzeitbetriebssystems iRMX 86. Dieses war eine Weiterentwicklung des nunmehr bekannten und breit verwendeten iRMX 80. Der Erfolg des Konzeptes wurde noch dadurch verstärkt, dass man ins iRMX 86 auch die Unterstützung der einzelnen Multibus-I-Hardwaremodule integrierte. iRMX 86 unterstützte somit quasi von Haus aus das Dual-Port-Memory-Access-Konzept von Multibus I für Mehrrechner-Architekturen. Mit dem iRMX 86 konnte dem Benutzer eine vollständige Palette von Softwaremodulen angeboten werden, die er je nach Anwendung im Zielsystem einbetten konnte.

Das Betriebssystem iRMX 86 baut auf dem erfolgreichen Schichten-Modell-Konzept auf. Dabei kann jede Schicht vom Benutzer frei im Zielsystem konfiguriert werden (Fig. 1). Die so gewonnene Flexibilität erlaubt, die Betriebssystem-Software den Bedürfnissen anzupassen. Die beim iRMX 86 vorhandenen Schichten sind in der Figur 1 angegeben.

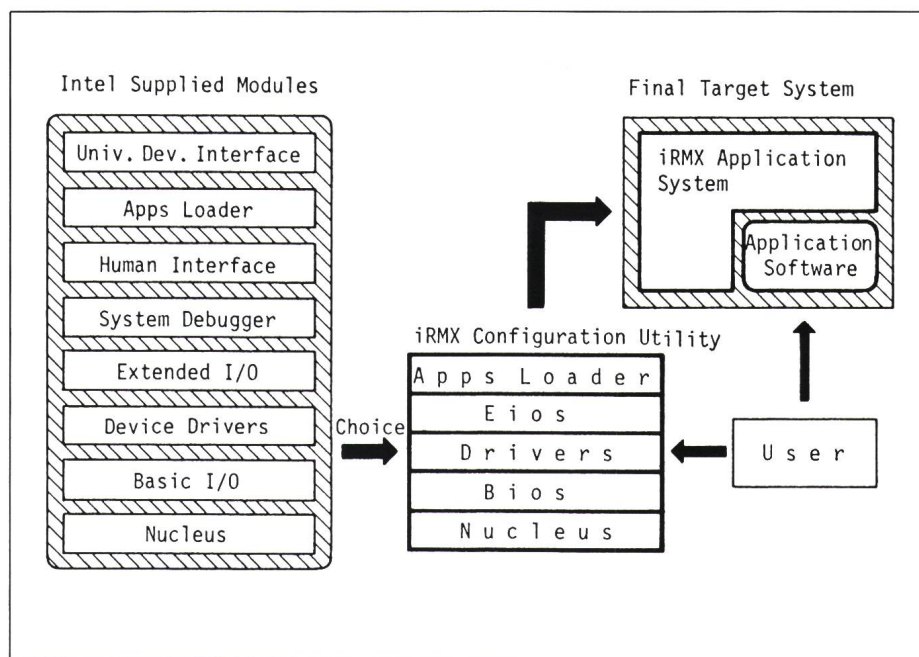
Mit der Zeit wurden die Fähigkeiten von iRMX 86 weiterentwickelt und dem technischen Fortschritt angepasst. Folgende Funktionen werden zurzeit dem Anwender u.a. angeboten:

- Multiuserfähigkeit,
- Entwicklungsumgebung auf dem Zielsystem mit allen wichtigen Sprachen,
- PROM-Programmierunterstützung,
- verschiedenste Treiber für alle von Intel angebotenen Multibus-I-Karten,
- Netzwerkfähigkeit mit Hilfe vom iRMXNET und Virtual-Terminal-Funktion.

Selbstverständlich kann das ganze Betriebssystem in EPROM-Speichern installiert werden, was vor allem bei Anwendungen in eingebetteten Systemen vorteilhaft ist. Im übrigen ist der Benutzer frei, mit Hilfe eines interaktiv laufenden Konfigurationsmenüs eine eigene massgeschneiderte Version von iRMX 86 zu generieren.

## Realtime-Betriebssystem iRMX 286

Mit der Zeit sind immer mehr Anwendungen an die Adressierbarkeits-



**Figur 1 iRMX-Konfigurationsprozess**

Der Benutzer kann sein Applikationssystem mit Hilfe der iRMX Configuration Utility massgeschneidert konfigurieren.

- |                        |   |
|------------------------|---|
| 1 Nucleus              | Betriebssystem-Grundfunktionen  |
| 2 Basic I/O            | System für den Zugriff auf Speichermedien und Ein-/Ausgabeeinheiten                 |
| 3 Device Drivers       | Treiber-Module für Ein- und Ausgabe   |
| 4 Extended I/O         | System für den synchronen Zugriff auf Sekundärspeicher und Eingabeeinheiten         |
| 5 System Debugger      | Testhilfen-Subsystem  |
| 6 Human Interface      | System für die Steuerung der Benutzer-Interfaces                                    |
| 7 Apps Loader          | Application Loader zum Laden und Starten von Applikationen aus dem Sekundärspeicher |
| 8 Univ. Dev. Interface | Universelle Programmentwicklungsstelle  |

grenze des 8086-Chips gestossen. Dies hat mit dazu beigetragen, dass Intel das Echtzeit-Softwarepaket mit einem neuen aufwärts kompatiblen Produkt, dem iRMX 286, ergänzt hat. Wie der Name sagt, läuft es auf dem 80286-Chip. Es nützt dessen Adressiermöglichkeiten bis auf 16MByte voll aus.

Um bei der Einführung von neuen Softwareprodukten (speziell beim 32-Bit-Realtime-Betriebssystem, das gegen Ende des Jahres angekündigt werden soll) eine irreführende Namensgebung zu vermeiden, wurde der Familienname leicht geändert: iRMX 86 wurde zu iRMX I, iRMX 286 zu iRMX II und iRMX 386 zu iRMX III. Die Figur 2 zeigt die wichtigsten Betriebssysteme, die auf Intel-Mikroprozessoren laufen können.

Beim iRMX-II-Echtzeitbetriebssystem wird der 80286-Chip im sogenannten Protected Virtual Address

Mode (PVAM) betrieben [1]. Es ergeben sich dadurch einige Vorteile:

- Es sind mehr Speicher direkt adressierbar,
- Es ist ein neuartiger Schutz gegen die häufigsten Fehler im Programmieren vorgesehen (Segmente sind durch die 80286-Chip-Hardware zugriffsgeschützt),
- Es wird eine erweiterte Palette mit neuen Funktionen angeboten.

Unter den neuen Funktionen befindet sich auch neu (ab Release 3, Frühjahr 1988) die Unterstützung für die Multibus-II-Architektur [2]. Diese erlaubt nun, verschiedene Betriebssysteme mit iRMX in einer Multibus-II-Anwendung transparent zu koppeln [3]. iRMX II ist selbstverständlich im Kompatibilitätsmodus auf dem 80386-Chip voll lauffähig.

Die Hauptmerkmale von iRMX II sind in Tabelle I angegeben. Neben diesen sind die vielen Spezialprogramme für iRMX II zu erwähnen, wie High-Level-Debugger, Virtual Terminal, Print Spooler, x2.5-Netzwerkanschluss, Source-Control-System für die Software-Entwicklung usw. Seit kurzem ist iRMX II auch auf der PC-AT-Architektur, die sogenannte PC-Plattform, verfügbar.

### Realtime-Kernel iRMK I.1

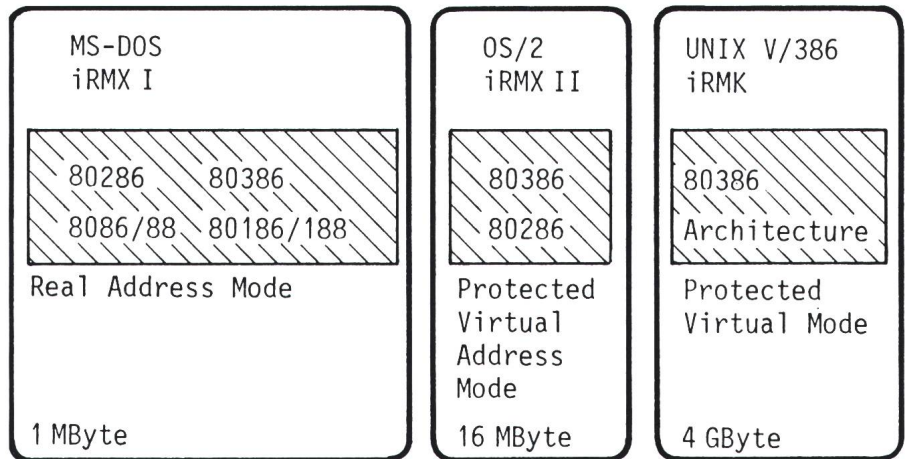
Das neueste Mitglied in der Echtzeit-Softwarefamilie von Intel ist der Realtime-Kernel<sup>1</sup> iRMK I.1. Bei diesem Softwarepaket handelt es sich um den ersten, speziell für den 80386-Chip geschriebenen 32-Bit-Kernel, der erlaubt, die schnellen Echtzeitvorgänge mit einer ebenso schnellen Echtzeitsoftware zu steuern. iRMK I.1 ist auch der Kern des 32-Bit-Realtime-Softwarepakets iRMX 386 (iRMX III), das zurzeit in Entwicklung steht. iRMK I.1 wird voll aufwärtskompatibel mit dem iRMX-386-Nukleus sein und somit die Kontinuität für den Anwender sicherstellen (Fig. 3). iRMK I.1 wurde auf hohe Leistung getrimmt. Alle Systemaufrufe benötigen nur einige Mikrosekunden. Die Interrupt-Latenzzeit z.B. beträgt im Maximum 35 µs, der Kontextwechsel 6,6 bis max. 50 µs. Die Hauptmerkmale von iRMK I.1 sind in Tabelle II aufgeführt.

Die Figur 4 zeigt die einzelnen Module, die rund um den Kern konfiguriert werden können. iRMK I.1 benötigt einzig den 80386-Chip als Basishardware. Somit kann jeder Anwender seine eigenen massgeschneiderten Module für die Systemhardware entwerfen und einbauen. Der Einsatz des Kernels bedeutet für den Benutzer einen etwas höheren Aufwand beim Programmieren und verlangt deshalb mehr Sorgfalt beim Entwurf. Diese Nachteile werden aber durch die Flexibilität und die Einsatzmöglichkeiten von iRMK I.1 mehr als kompensiert.

### Ausblick und Entwicklungstendenzen

Es steht ausser Frage, dass sich mit dem Einsatz von standardisierten

<sup>1</sup> Der Kernel beinhaltet die Grundfunktionen des Betriebssystems. Der Nukleus baut auf dem Kernel auf und bietet einen höheren Programmierkomfort.



Figur 2 Die gängigsten Betriebssysteme für Intel Microprozessoren

*Protected Virtual Address Mode (PVAM):*

Betriebsmodus des 80286 erlaubt, unter Verwendung der integrierten Zugriffsschutzmechanismen auf den virtuellen Speicher zuzugreifen.

*Virtual Mode:*

Dieser Betriebsmodus des 80386 erlaubt, 8086-Programme in einer PVAM-Umgebung laufen zu lassen.

### Hauptmerkmale des iRMX II

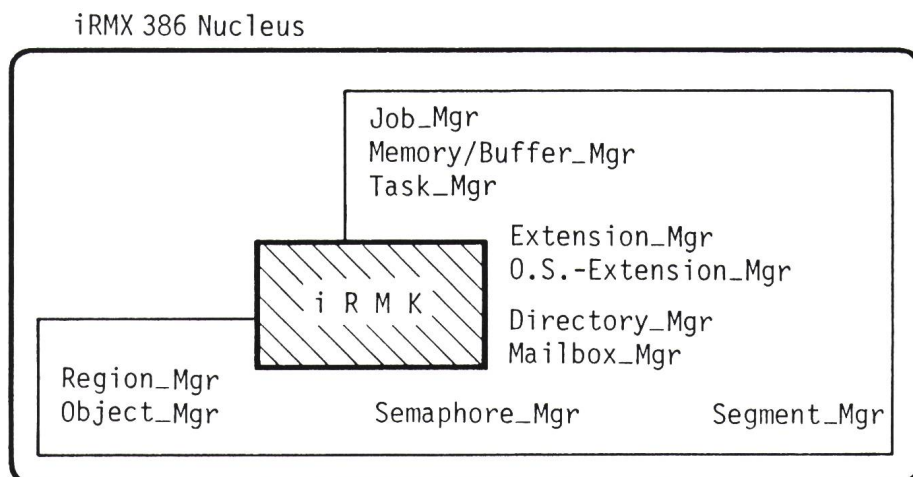
- 16-MByte-Adressraum mit Zugriffsschutz
- Höhere Leistungsfähigkeit als iRMX I durch Verbesserung des Nukleus
- Kompatibilität mit iRMX I
- Neue Scheduling Mechanismen (Round Robin)
- Neue Benutzerkommandos wie z.B. der Background-Befehl für quasi parallele Verarbeitung von Programmen
- Dynamisches LOG ON für das Mehrbenutzersystem
- Volle Multibus-II-Unterstützung mit den entsprechenden Treibern für die Multibus-II-iSBC-Karten (Disk, Tape, Communication)

Tabelle I

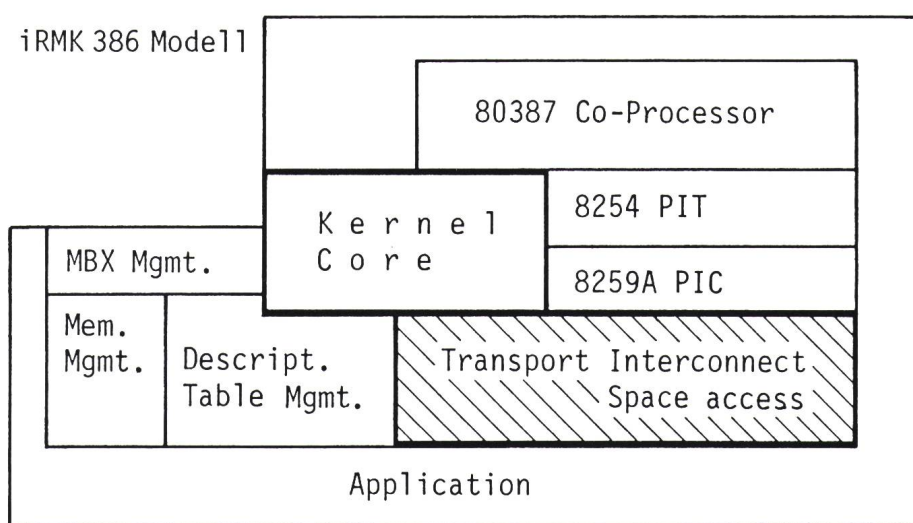
### Hauptmerkmale des iRMK I.1

- 4-GByte-Adressraum
- 32-Bit-Echtzeit-Executive mit entsprechendem Satz von Funktionsaufrufen
- Optimierte Struktur für die 80386-Architektur
- Kompakter Code, konfigurierbar mit Modulen (der Kern benötigt im Maximum einen 28-KByte-Code-Speicher)
- Multibus-II-Transportschnittstelle für Multibus-II-Message-Passing
- Voll installierbar in EPROM
- Lauffähig auf dem neuen 32-Bit-Mikrochip 80376 der 80386-Familie

Tabelle II



Figur 3 iRMX III-Nukleus-Architektur



Figur 4 iRMK I.1-Architektur

Softwarepaketen neue Möglichkeiten für die Realisierung von Anwendungen eröffnen. Die bereits in der Einführung erwähnten Vorteile kommen während der Produktlebensdauer voll zum Tragen. Sie erlauben, mit der Zeit den gewünschten Standardisierungseffekt zu erreichen, und garantieren somit den langfristigen Schutz der Basisentwicklung in einem Unternehmen. Nicht zu vernachlässigen ist auch der klare Migrationspfad, der dem Anwender offensteht. Er kann sicher sein, dass nicht nur die Applikation, sondern auch die Betriebssystemsoftware mit der Entwicklung Schritt hält. Dies wird in Zukunft – vor allem in Anbetracht der Einführung immer neuer Chip-Generationen – umso wichtiger sein, wenn die Echtzeitsoftware die generelle Halbleiterentwicklung voll mithalten soll.

#### Literatur

- [1] Operating systems writer's guide 286. Intel document 121960-002. Santa Clara/California, Intel Semiconductor Corporation.
- [2] Multibus II transport protocol specification and designer's guide. Intel Document 45358-001. Santa Clara, Intel Semiconductor Corporation.
- [3] F. Huguenin: Multibus II – «Message-Passing» Software. Elektroniker – (1987)9, S. 93...98.