

**Eine projektspezifische Anwendung von PEARL  
auf einer Konfiguration von IBM-PC und INTEL 8086  
mit dem Real-time-Betriebssystem iRMX86**

Dipl.-Inform. Heribert Müller  
DORNIER SYSTEM GMBH  
Postfach 1360, 7990 Friedrichshafen

## **I. Abstract**

Im Rahmen des Projekts Luftbildauswerteanlage für die Bundeswehr (LbAA-BW) ist zur Programmentwicklung eine PEARL-Umgebung (WERUM-PEARL Programmiersystem) geschaffen worden, die der Bedingung genügt, daß als Zielrechner ein "embedded system" auf 8086 Basis vorhanden ist und die Programmerstellung deshalb auf einem getrennten Entwicklungssystem durchgeführt werden muß. Als Entwicklungssystem dient hierbei der IBM PC AT/02.

Voraussetzung zum Einsatz für die PEARL-Umgebung ist die Bereitstellung des Betriebssystems AT/RTX (entspricht dem INTEL-BS iRMX86), da das Laufzeitsystem auf diesem Betriebssystem aufsetzt. Mit dieser SW-Konfiguration wird die operationelle Software erstellt.

Zusätzlich ist auf dem IBM PC eine konfigurierbare BS-Version von iRMX86 vorhanden, dessen Nucleus und BIOS (Basic I/O System) für das Zielsystem konfiguriert wird. Darüberhinaus ist ein PEARL-Laufzeitsystem für den Zielrechner verfügbar, das speziell auf das konfigurierte Betriebssystem zugeschnitten ist.

Während der Testphase wird die Applikation zusammen mit dem speziellen PEARL-Laufzeitsystem und dem abgemagerten Betriebssystem mit Hilfe des PCSBC-Target System Program Loader in den RAM-Bereich des Zielrechners geladen. Der iSDM86 (System Debug Monitor) ermöglicht die Programmausführung und das interaktive Testen der Applikation ähnlich wie mit einem In-Circuit-Emulator.

Intention der Ausführung ist es, eine HW- und SW-Konfiguration aufzuzeigen, mit der eine komplexe Anwendung realisiert wird.

## 2. Kurzbeschreibung der Luftbild-Auswerte-Anlage

Die Luftbildauswerteanlage hat zur Aufgabe, Luftbilder, welche von verschiedenen optischen bzw. optronischen Sensoren aufgenommen wurden, schnell und präzise auszuwerten.

Sie umfaßt die Gerätekomponenten:

- Leuchttisch mit Brückenstereoskop und angeflanschter S/W-Fernsehkamera
- Mikrokartenprojektor (MKP) mit integrierter Farbkamera und Display
- S/W-Monitor und Farbmonitor
- Grafik-Display-Controller und Bildmischeinheiten, um die Monitore grafikfähig zu machen
- sowie eine übergeordnete Rechneranlage

Für den Auswertevorgang ist es zunächst notwendig, sich mit den Geländegegebenheiten, von denen Luftbilder angefertigt werden, vertraut zu machen; eine wichtige Voraussetzung für die spätere Zuordnung markanter Punkte und Ziele in einem aufgenommenen Bild zum Kartenmaterial (UTM-Koordinatensystem).

Dazu können auf das MKP-Display Kartenausschnitte projiziert und am Farbmonitor vergrößert wiedergegeben werden. Nach Eintreffen der auszuwertenden Bilder ist in einem ersten Schritt ihre Lage auf dem Leuchttisch zu bestimmen (innere Orientierung). Im Anschluß daran muß die bereits erwähnte Zuordnung von bekannten Punkten auf dem Film zu den UTM-Koordinaten vorgenommen werden. Die Tischkoordinaten der ausgewählten Punkte im Film und die zugehörigen UTM-Koordinaten werden vom LbAA-Rechner automatisch übernommen und dann zugeordnet. Mit diesen Informationen kann berechnet werden, mit welchem 'Blickwinkel' eine Kamera das Gelände erfaßt hat (äußere Orientierung). Die Berechnungsergebnisse werden in eine Wissensbasis eingebracht, mit der jedwede Auswertung photogrammetrisch oder interpretativ erfolgen kann.

Es lassen sich mit ihr sehr schnell Zielpositionen, Zielausdehnung, Bildstrecken, Objektgrößen und Entfernungen berechnen, wodurch insbesondere der Zeitbedarf für die Auswertung reduziert und das Informationsalter verringert wird.

Bei Bedarf werden die Auswertergebnisse an die übergeordneten Führungssysteme weitergeleitet.

Zentrale Steuereinheit für die Gerätekomponenten und deren Zusammenspiel innerhalb der LbAA-BW ist das MUDAS-Datenprozessorsystem 426 (MUDAS= Modulares Universelles Daten-Akquisitions-System). Es ist für den operationellen Betrieb als 'embedded system' ausgelegt, umfaßt ausschließlich die erforderlichen Komponenten des operationellen Prozeßrechnersystems und enthält die Software für die Vorbereitung und Durchführung der Luftbildauswertung. Mit zur Software gehört neben den Auswerteverfahren die Ansteuerung aller angeschlossenen Geräte einschließlich der Nutzerschnittstelle für die dialogorientierten Funktionen.

### **3. Das MUDAS-Datenprozessorsystem 426 als zentrale Steuereinheit innerhalb der LbAA-BW** (Zielsystem in der Softwareentwicklungskonfiguration)

#### **3.1 Hardwarekonfiguration**

Die im Projekt eingesetzte Hardware MUDAS-DPS 426 besteht aus einem  $\mu$ -Bus-Kartensystem auf 8086-Basis. Die CPU wird mit 5 MHz betrieben und enthält

- die INTEL 8087 Floating Point Unit
- 3 Timer
- 1 MB EPROM/RAM Speicher, davon 128 KB batteriegepuffert

Als Geräte-Schnittstellen werden benötigt:

- 2 vierfach V24-Interface Karten (RS 232 C)
- 1 zweifach V24-Karte mit Ausnutzung aller V24 Leitungen für spezielle Geräteanschlüsse (Funk, Gleichstromdatenübertragung)
- 1 serielle Test-/Wartungsschnittstelle (ON-BOARD USART)

### 3.2 Softwarekomponenten des Zielsystems

Bei der Auswahl eines Betriebssystems und der zu verwendenden Sprachen war die Anforderung an die Geschwindigkeit und Gegebenheiten der Gerätekommunikation sowie die Komplexität der Software und der Auswertelgorithmen zu berücksichtigen. Außerdem war die Hardware MUDAS DPS-426 vorgegeben und PEARL als Hauptprogrammiersprache vom Auftraggeber gewünscht.

Die Entscheidung fiel zugunsten eines Real-time Betriebssystems, auf dem Laufzeitroutinen bewährter Hochsprachen (HOL) aufsetzen können. Die Zuordnung der Sprachen zu den zu realisierenden Problemlösungen wurde hierbei nach deren Eignung getroffen.

Somit umfassen die im Zielsystem verwendeten Softwarekomponenten:

a) das Real-time Betriebssystem iRMX86 von INTEL

Dieses Betriebssystem umfaßt mehrere Layer, die aufeinander aufbauen, besonders fein konfigurierbar sind und damit den operationellen Erfordernissen genau angepaßt werden können. Benutzt werden von den iRMX86 Möglichkeiten nur die des NUCLEUS und des BIOS (Basic I/O-System).

Mit diesen beiden Subsystemen stehen die Grundfunktionen zur Task-, Interrupt-, Speicher- und E/A-Verwaltung zur Verfügung. Mit der Wahl und Verwendung dieses Betriebssystems ist beabsichtigt, es für weitere echtzeitorientierte Projekte einzusetzen und dafür Erfahrungen zu gewinnen.

b) die Applikationssoftware

Der überwiegende Teil wird in WERUM-PEARL geschrieben. Die photogrammetrischen Verfahren zur Berechnung von Zielkoordinaten mit gegenseitiger Zuordnung der UIM-Koordinatenwelt und der Tischkoordinatenwelt sind in FORTRAN-86 programmiert, wohingegen betriebssystembezogene und mathematische Interfaceprozeduren in der dafür besonders geeigneten Sprache, PL/M-86 formuliert sind, z.B.

- Gerätetreiber
- Interface Prozeduren zwischen den Treibern und den PEARL-Programmen
- Interface Prozeduren zwischen Betriebssystem und PEARL, insbesondere im Hinblick auf die Device Connection Behandlung
- die Anbindung mathematischer Funktionen (SIN, COS, ...) an die PEARL-Programme

c) das PEARL-Ablaufsystem BAPAS (Basis für Prozeßautomatisierungssysteme)

Dem Betriebssystem aufgesetzt ist ein gegenüber der Normalversion modifiziertes Laufzeitsystem (vgl. Abschnitt 4.4), das Routinen zu

- Bit- und Characterbehandlung
- Typwandlung
- formatierte Ausgabe
- task-, Synchronisations-, Zeit- und Interrupt-Behandlung usw. bereitstellt.

d) die Run-time-libraries

- Interface Library zum 8087-Prozessor
- Floating-point exception handling Prozeduren
- Conversion Library
- Mathematische Library mit logarithmischen Funktionen, trigonometrischen Funktionen usw.
- nicht mathematische FORTRAN-86 Run-time-Libraries

## 4. Das Entwicklungssystem

### 4.1 Hardwarekonfiguration

Die Programmentwicklung erfolgt auf IBM-PC AT/02's auf 80286-Basis, wobei ein PC mit

- 640 KB Arbeitsspeicher
- 2 x 20 MB Winchester
- 1,2 MB Floppy Disk
- Streamer für DOS-Backup's
- Drucker
- 2 serielle/2 parallele Interfaces ausgerüstet ist.

### 4.2 Softwarekomponenten des Entwicklungssystems

Betriebssysteme: MS-DOS und AT/RTX

Die Winchesterplatten sind in 2 Partitions unterteilt: eine MS-DOS-Partition und eine RMX-Partition, d.h. auf jedem PC sind 2 Betriebssysteme verfügbar, wovon eines je nach Einstellung der aktiven Partition gebootet wird. Mit dem Programm FDISK, das auf beiden Systemen vorhanden sein muß, kann auf die jeweils andere Partition und damit auf das andere Betriebssystem umgeschaltet werden.

**MS-DOS** ist das Standardbetriebssystem auf dem PC. Benutzt wird die Version 3.0.

Das Betriebssystem **AT/RTX** ist eine Portierung des INTEL BS iRMX86 (Version 6 bzw. Version 7) auf IBM PC durch die Firma RTCS (Real Time Computer Science Corporation, USA).

AT/RTX ist ein real-time, multi-tasking und multi-programming Softwaresystem für die iSBC's (Single Board Computer) von INTEL sowie für andere auf iAPXxy basierende Mikro-Computersysteme. Das gesamte Softwaresystem besteht aus einer Sammlung von Layers, die je nach Bedarf verwendet und konfiguriert werden können. Jedem Layer sind bestimmte Aufgaben zugeordnet.

Der Nucleus als Herz des Systems übernimmt die Aufgaben des Multitasking, der Taskkontrolle, der Synchronisation und der Taskkommunikation.

Das IOS (Basic und Extended) ist vorgesehen für das File-Management und bildet das device-unabhängige Interface zu I/O-Devices. Die device-abhängigen Gerätetreiber einschl. Terminal-Treiber und Terminal Support Code ist Bestandteil des BIOS.

Das Human Interface steuert ein Applikationssystem mit Kommandos, die vom Terminal aus eingegeben werden können.

Die UDI-Schnittstelle (Universal Development Interface) ist ein Software-Interface, das erlaubt, daß Compiler, Editoren und sonstige SW-Entwicklungstools auf die Möglichkeiten des RMX-Systems zugreifen.

Vervollständigt wird das Gesamtsystem durch den Application Loader, den System- und Real-Time-Debugger, den Terminal Handler und den Crash Analyzer.

Unter den beiden für die Programmentwicklung benutzten Betriebssysteme ist folgende Software verfügbar.

AI/RIX-System	MS-DOS-System
- Sprachen + Compiler	
a) WERUM PEARL incl. PEARL-Ablaufsystem	a) siehe RTCS/UDI
b) FORTRAN 86	
c) PL/M-86	
d) ASM-86	
- Utilities	
a) alle INTEL-Utilities, die auf UDI lauffähig sind	a) RTCS/UDI Bereitstellung der UDI- Schnittstelle unter MS-DOS
z.B. - LINK86 - LOC86 - OII86 - LIB86 - CREF86	Damit sind alle UDI-lauf- fähigen Utilities unter MS-DOS verfügbar.
b) AEDIT	b) PCSBC-Target System Programm Loader PCSBC ist ein IBM PC Terminal Emulator Programm, das Bestandteil von RTCS/UDI ist und zusammen mit dem iSDM86 (System Debug Monitor) arbeitet. Der iSDM86 muß auf dem TARGET installiert sein (EPROM)
c) ICU (Interactive Configuration Utility) zum Konfigurieren des Betriebssystems iRMX86 für das Target-System	c) VIERM III DEC-VI-Terminal Emulator, der als Kommunikationsbasis zwischen VAX-Rechnern und dem PC verwendet, aber auch zusammen mit dem iSDM86 benutzt werden kann.
d) DOS2RIX Lesen und Konvertieren von DOS-Disketten mit Abspeichern der Files auf der RIX-Partition	
e) RIX2DOS Konvertieren und Schreiben von RIX-Files auf DOS-Disketten (360KB- oder 1,2 MB-Format)	



### 4.3 PEARL-Programmentwicklung und Ausführung auf dem Entwicklungssystem

Auf dem IBM-PC ist die PEARL-Programmentwicklung entsprechend der WERUM PEARL Sprachbeschreibung /1/,/2/ uneingeschränkt möglich. Dies gilt für alle Programme, die auch auf dem IBM-PC zur Ausführung kommen.

Der PEARL-Compiler läuft auf der UDI-Schnittstelle und erzeugt ASM-86 Code. Zur Programmausführung auf dem IBM-PC ist als Interface zum Betriebssystem AT/RTX das PEARL-Laufzeitsystem BAPAS notwendig, bestehend aus

BAPAS-K	(Kern)
BAPAS-FILEEA	(Datei Ein-/Ausgabe) und
BAPAS-FORMEA	(formatierte Ein-/Ausgabe).

Ausführbar sind die PEARL-Programme auf dem IBM-PC nur, wenn der Floating Point Prozessor und das AT/RTX verfügbar sind.

Das BAPAS-K entspricht dem Betriebssystemkern NUCLEUS und benutzt die dort definierten Library-Routinen, während BAPAS-FILEEA und BAPAS-FORMEA darüberhinaus Routinen des UDI-Layers benutzen.

Das PEARL-Testsystem zum Debuggen auf Sprachebene ist bei 640 KB Arbeitsspeicher nur eingeschränkt, d.h. bei kleineren PEARL-Programmen möglich, da allein für das iRMX86 Betriebssystem 312 KB benötigt werden und das Testsystem ca. 235 KB umfaßt.

### 4.4 PEARL-Programmentwicklung für das Zielsystem

Programme für das Zielsystem genügen einem leicht reduzierten Sprachumfang, da im Zielsystem kein File E/A-Transfer benötigt wird und auch kein formatiertes Lesen möglich ist. Einem formatierten Lesen von Daten steht die zu beliebigen Zeitpunkten mögliche Funktionstasteneingabe (Einzelzeichen) entgegen.

Somit enthält das PEARL-Laufzeitsystem ein E/A-Paket, welches nur noch das PUT-Statement für sequentielle Ausgabegeräte ohne File-Struktur unterstützt. Die PEARL-Statements READ, WRITE und GET werden nicht mehr bereitgestellt. Als Ersatz für das GET-Statement steht die Spezialroutine GETCHAR zur Verfügung, die das zeichenweise Einlesen auch für Control-Character ermöglicht und z.B. Funktionstasteneingaben ohne CR-Abschluß akzeptiert und deren Code nicht automatisch am Bildschirm reflektiert.

Das PEARL-Laufzeitsystem setzt nur noch auf dem NUCLEUS und auf dem BIOS auf, wodurch sich eine erhebliche Reduktion des Speicherbedarfs ergibt: auf der PEARL-Laufzeitsystemseite ist die Ersparnis ca. 10 KB und auf der Betriebssystemseite bis zu 172,5 KB Programmcode und 20 KB Daten ohne Berücksichtigung der Human Interface Kommandos und des UDI.

#### **4.5 Betriebssystemgenerierung von iRMX86 für das Zielsystem (NUCLEUS und BIOS)**

Zentral für die BS-Generierung ist die ICU (Interactive Configuration Utility), die alle Tools bereitstellt, um ein spezielles Applikationssystem zu generieren. Die ICU besteht aus 2 Phasen. In der 1. Phase führt sie zunächst durch ein Definitionsfile, in dem interaktiv und beliebig oft änderbar Angaben zur HW- und SW-Konfiguration gemacht werden können.

Unter anderem müssen im Definitionsfile alle sog. First-level Jobs (System- und User-Jobs) angegeben und spezifiziert werden. Die First-Level Jobs bilden eine Umgebung, in denen RMX-Objekte: Segmente, Mailboxen, Semaphore, Tasks und weitere Jobs residieren und ablaufen. Mit zur Umgebung gehört ein Memory Pool sowie eine Object-Directory.

Diese Jobs hängen alle an einem übergeordneten ROOT-Job, von dessen Initialisierungstask sie zur Laufzeit gestartet werden. Als Systemjobs werden im Definitionsfile im Rahmen des Projekts LbAA-BW

- das BIOS und während der Testphase
- der Real Time Debugger (vgl. Abschnitt 5) spezifiziert und auf der Userseite
- ein Job, der das gesamte Applikationsprogramm umfaßt und
- ein Job, der alle Device Connections im ROOT-Job Directory bereitstellt

Für die User-Jobs müssen im File neben ihrem Namen u.a. auch ihre Code-, Datensegment- und Stacksegmentadressen bekanntgegeben werden.

Die ICU generiert aufgrund der Eingaben Konfigurationsfiles (Assembler Makro Files), die im Zusammenhang mit dem NUCLEUS die HW-Umgebung beschreiben, im Zusammenhang mit dem BIOS Device Informationen für die Gerätetreiber umfassen und für jeden Betriebssystem-Layer die benutzbaren RMX-Calls, d.h. die für die Applikation aufrufbaren Betriebssystemroutinen definieren.

Außerdem stellt die ICU ein Master Submit File (Kommando File) für die eigentliche Betriebssystemgenerierung zusammen. Bei seiner Aktivierung wird zunächst die Assemblierung aller Makrofiles zu jedem Layer veranlasst. Danach werden alle Objektmodule mit den erforderlichen NUCLEUS- und BIOS-spezifischen Libraries zusammengebunden.

In der 2. Phase (zugleich 2. Teil des Master Submit Files) werden die Code-, Daten- und Stackteile aller Systemjobs mit dem LOC86 auf die gewünschten absoluten Adressen des Targets gelegt. Danach werden die Job-Module in eine Library eingetragen, die um die Applikationssoftware-Module erweitert und damit vervollständigt wird.

I.a. müssen die ICU-Durchläufe iteriert durchgeführt werden, um ein adressmäßig überschneidungsfreies Memory Mapping Layout aller Segmente mit möglichst wenig Speichersegmentierung zu erhalten.

## 5. Laden und Debuggen der operationellen Software

Als Voraussetzung für den Ladevorgang der Software in das Target-System Memory muß die gesamte System- und User-Software in einer Library als ein ladbares File auf der MS-DOS-Partition vorhanden sein. Dazu ist das von der ICU erstellte File von der RTX-Partition via Floppy auf die DOS-Partition zu übertragen. Von dort wird es mit Hilfe des PCSBC-Target System Program Loaders, der Bestandteil von RTCS/UDI ist, in die beim LOCATE-Vorgang vorgegebenen Adreßbereiche geladen. Korrespondierendes Programm beim Debug-Vorgang ist der iSDM86 (System Debug Monitor), der im TARGET-System abläuft und die vom IBM-PC kommenden Befehle durchführt.

Die Kommunikation zwischen HOST und TARGET erfolgt über die ON-BOARD USART Schnittstelle und erlaubt im Rahmen eines OFF-line Debuggings auf ASSEMBLER-Ebene u.a.

- eine Applikation mit Single Step Operationen oder mit voller Geschwindigkeit auszuführen
- Breakpoints zu setzen
- CPU Register auszuführen und zu ändern
- Memory Inhalte zu disassemblieren, auszuführen, zu ändern, zu kopieren oder nach bestimmten Werten abzusuchen

Über das Command Extension Interface (CEI) kann der Kommandovorrat des Monitors erweitert werden.

Wird der System-Debugger in den TARGET mitgeladen, dann erweitert sich die Kommandoliste des iSDMB6 um die View-Kommandos, mit denen iRMX86 Datenstrukturen und iRMX86 System Calls angezeigt werden können. Beide Debugger sind vom IBM-PC aus nicht interruptfähig.

Als Online-Debugger mit beliebiger Eingriffsmöglichkeit vom IBM-PC aus steht der Real Time Debugger zur Verfügung, der stets als Hintergrundjob mitläuft per CTRL/D aktiviert werden kann. Der Debugger bietet u.a. die Möglichkeit

- ihn als Task, Job oder System Exception Handler zu benutzen
- iRMX86 Objektlisten (.Jobliste, Tasks, Ready Tasks, Suspendierte Tasks, Task Queues etc.) anzuzeigen
- Jobs, Tasks, Mailboxen, Semaphore, Regions und Segmente zu untersuchen
- Speicherinhalte zu ändern oder auszuführen
- Breakpoints zu setzen, zu ändern, zu listen oder zu löschen

Um das PEARL-Testsystem für das MUDAS-DPS 426 in Zukunft einsetzen zu können, wird eine Grundvoraussetzung für seine Benutzung: ein Massenspeicheranschluß am DPS 426 vorbereitet. Mit der Verwendung des Testsystems ist das Debuggen auf Hochsprachenebene möglich, wodurch die Fehlersuche gegenüber dem Testen auf Assembler-Ebene erheblich erleichtert wird.

## **6. Aktueller Stand/Erfahrungen**

Im Projekt lbAA-BW ist bisher der Abschluß der Grobentwurfsphase erreicht. Die Codier- und Integrationsphase werden genaueren Aufschluß ergeben zur Effektivität der Softwareimplementierung, zur Handhabung der Testmethoden und zum Systemverhalten.

Erste Erfahrungen beim Übersetzen von PEARL-Programmen haben ergeben, daß der Compilervorgang gegenüber dem nachfolgenden Assemblerlauf minimal länger dauert (Durchschnittsverhältnis 55 % zu 45 %). Pro Minute werden im Durchschnitt 200 - 300 PEARL Statements compiliert und assembliert, wobei 25 - 28 PEARL-Quellzeilen 1 KB Plattenspeicher benötigen. Der Plattenspeicherbedarf eines erzeugten ASM-86 Files umfaßt ca. das 3,5 - 4fache des PEARL-Quellfiles und das 7-8fache des erzeugten Objektfiles.

#### Literatur

/1/ Wulf Werum, Hans Windauer  
"Introduction to PEARL"  
Vieweg-Verlag

/2/ PEARL Programming System for IBM PC's under PC/RIX or AT/RIX  
User Manual  
WERUM Datenverarbeitungssysteme GmbH