Kompakter Embedded Computer mit Touch-Bedienung

Adrian E. Weitnauer • Christoph Angst • René F. Oberhänsli

Immer mehr Produkte werden mit einer farbgrafischen Benutzerschnittstelle ausgestattet. In vielen Fällen werden diese Schnittstellen durch herkömmliche Industrie-PC (IPC) gesteuert. In diesem Aufsatz wird eine praxiserprobte Alternative, ein kompletter Einplatinenrechner mit berührungsempfindlicher Anzeige, vorgestellt. Das Modul ist auf dem Konzept einer grafikfähigen virtuellen Maschine aufgebaut und arbeitet geräuschlos mit passiver Kühlung. Anwendungen werden mithilfe frei erhältlicher Softwarewerkzeuge in einer einfach zu erlernenden Hochsprache erstellt.

Am Anfang eines Entwicklungsprojekts steht immer die Evaluation der Systemkomponenten im Vordergrund. In der Regel ist es sinnvoll, komplexe Rechnerhardware nicht selber zu entwickeln, sondern zuzukaufen. Die Rechnerhardware ist jedoch oft nur kurzzeitig unverändert erhältlich, weil sie einer ständigen Weiterentwicklung unterworfen ist. Hardware und die Betriebssystem-Software stammen außerdem oft nicht aus einer Hand. Bei Investitionsgütern mit langer Lebensdauer ist die Ersatzteilbeschaffung ein großes Problem. Was passiert, wenn nach wenigen Jahren der Steuerrechner ausfällt und kein identischer Ersatz mehr erhältlich ist? Oft wird dann ein Redesign notwendig, um die Anwendungssoftware auf die neue Hardware- und Softwareumgebung zu migrieren.

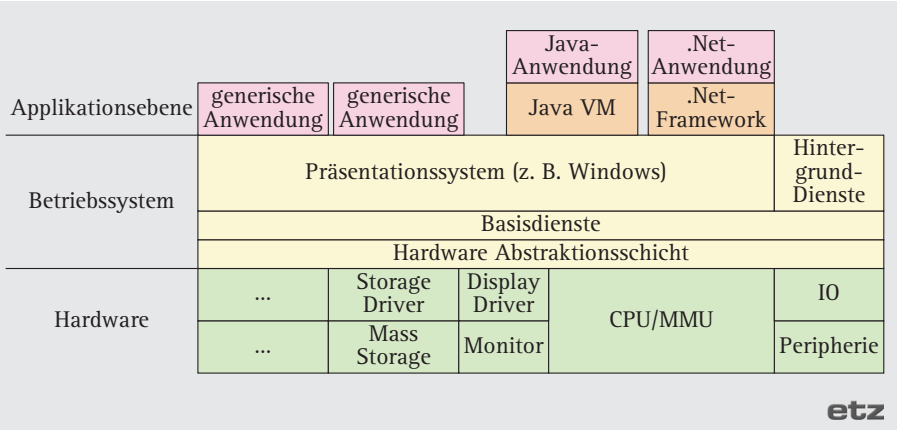

Bild 1. *Systemarchitektur eines Personal Computers*

Herkömmliche Systemintegration

Von einer modernen Benutzerschnittstelle wird mit Recht eine übersichtliche Benutzerführung erwartet, die grafisch ansprechend gestaltet ist und rasch reagiert. Völlig undenkbar ist ein Systemabsturz infolge eines Bedien- oder Programmierfehlers. Intuitiv zu bedienen sind Benutzeroberflächen, die auf berührungsempfindlichen Bildschirmen (Touch-Panels) basieren. Sie können bei Bedarf sogar vandalensicher aufgebaut werden. Es ist üblich, für solche Fälle IPC mit LCD-Monitor und Touchpanel als Maus-Ersatz einzusetzen. Die Anwendung ist dadurch eine ausgewachsene PC-Software mit der Gefahr von Treiber-Inkompatibilitäten, Timingproblemen, Schnittstellenfragen oder Unklarheiten rund um das Betriebssystem. Die großen Vorteile sind die nahezu unbegrenzte Ausbaufähigkeit, Speichertiefe und Rechenleistung. Allerdings werden gerade diese von solchen Systemen nur in wenigen Fällen überhaupt erwartet.

Vergleicht man ein fünfzehn Jahre altes PC-System mit einem aktuellen, fällt auf, dass die wichtigsten Aufgaben immer noch etwa gleich lang dauern: Das Verfassen eines Briefs oder das Entwickeln eines Spreadsheets nehmen immer noch gleich viel Zeit in Anspruch. Selbst der Bootprozess dauert bei beiden Systemen gleich lang. Der moderne Rechner wartet einfach viel schneller auf die Benutzereingaben und benötigt dafür deutlich mehr Strom als der alte. Dies illustriert, dass für viele Einsätze eine einfache Maschine mit einem schlanken Betriebssystem ausreichen würde.

Die Anwendungsentwicklung auf einem einfachen Betriebssystem gestaltet sich einfacher und übersichtlicher. Jeder Entwickler kennt mittlerweile die Randeffekte, die bei der Migration von Anwendungssoftware auf eine neue Betriebssystemversion auftreten können. Die Hersteller von Betriebssystemen versuchen die Arbeit des Entwicklers zu vereinfachen, indem Konzepte von virtuellen

Adrian E. Weitnauer, dipl. El.-Ing. ETH/SIA ist Inhaber der Weitnauer Messtechnik in Näfels, Schweiz.

E-Mail: adrian.weitnauer @weitnauer-messtechnik.ch

Christoph Angst, dipl. Ing. ETH ist bei der S-TEC electronics AG in Unterägeri, Schweiz, tätig.

E-Mail: s-tec@bluewin.ch

René F. Oberhänsli, dipl. El.-Ing. ETH ist CEO der S-TEC electronics AG in Unterägeri, Schweiz.

E-Mail: s-tec@bluewin.ch

Maschinen (Java, .Net-Framework) eingeführt werden, die das eigentliche Betriebssystem kapseln (**Bild 1**). Die Komplexität des Betriebssystems steigt zusammen mit den Anforderungen an die darunterliegende Hardware weiter an. Die vielen Schichten stammen nicht aus einer Hand, sodass nicht klar ist, wo Probleme auftreten können.

Maßgeschneiderte Systeme fehlen

Wer eine neue Benutzer-Maschine-Schnittstelle zusammenstellen will, steht aus oben genannten Gründen schnell vor unerwarteten Problemen. Das bekamen auch die Entwickler der S-TEC electronics AG [1] und der Weitnauer Messtechnik [2] konkret zu spüren. Als dann auch noch die evaluierte Hardware bereits nach einem halben Jahr abgekündigt wurde, brachen die Entwickler ihre erfolglose Suche auf dem Markt ab und begannen, ein eigenes Konzept für einen geeigneten Embedded-Computer (EC) mit grafischer Benutzeroberfläche zu entwerfen.

Ein Embedded Computer ist in einem technischen Gerät eingebettet. In der Regel kann auf übermäßige Rechenleistung zugunsten einer optimalen Energie- und Ressourceneffizienz verzichtet werden. Die Rechenleistung ist der Applikation angepasst. Die einfachste Anwendung, die man sich vorstellen kann, ist ein Bildschirm, auf dem einige Symbolbilder als Tasten dargestellt werden. Drückt man auf eine der „virtuellen Tasten“, soll ein Befehl an das angeschlossene System gesandt werden, dabei sollen die Tasten auf die Benutzereingaben reagieren, indem sie ihre Form verändern. Möchte man diese einfache Touchpanel-Anwendung auf einem handelsüblichen Industrie-PC erstellen, muss mit einer Hochsprache bereits ein recht anspruchsvolles Anwendungsprogramm geschrieben werden. Es muss die Tasten je nach der gewünschten Darstellungsart und dem ausgelösten Ereignis mit den entsprechenden Bilddateien anzeigen und gleichzeitig die gewünschten Befehlszeichenketten über eine der Schnittstellen absetzen. Trotz der Einfachheit bedarf es bereits einiger Programmierkenntnisse.

***Bild 2**. „eigerPanel“ 57, ein kompakter Embedded Computer mit langlebiger, brillanter LED-Hintergrundbeleuchtung*

Rechner-Konzept für Embedded Computer

Die Entwickler entschieden sich, einen neuen Komplettrechner samt integrierter Entwicklungsumgebung zu entwickeln. Das kompakte, vollgrafische Gerät sollte in einem einzigen Modul die gesamte Hardware umfassen: Single-Board-Computer, Grafikcontroller, Solid-State-Massenspeicher, TFT-Display mit Hinterleuchtung und Touch-Panel. Als Massenspeicher dient eine Compact-Flash-Karte. Die Ansteuerung des TFT-Moduls ist bei herkömmlichen Systemen oft mit zusätzlichen Schwierigkeiten verbunden. Ein direktes Ansteuern des Moduls mit dem Hauptprozessor ist erstens sehr umfangreich und schwierig zu realisieren, und zweitens verbraucht das Anwendungsprogramm viel Rechenleistung alleine für die Grafikdarstellung. Um das erste Problem ein wenig zu entschärfen, gibt es auf dem Markt Softwarewerkzeuge, die C-Sourcecode erzeugen. Dieser C-Sourcecode kann dann in das Anwendungsprogramm eingebunden werden.

Das Konzept des „eigerPanel“ (**Bild 2**) sah von Anfang vor, dass mächtige Grafikfunktionen direkt durch das Betriebssystem zur Verfügung gestellt werden. Um die wertvolle Rechenleistung möglichst vollständig dem Anwendungsprogramm zu reservieren, ist ein separater Grafikcontroller notwendig. Das TFT-Modul muss auf seiner Schnittstelle mit einer hohen Datenrate betrieben werden. Eine angenehme Bedienbarkeit setzt voraus, dass Bildwechsel möglichst rasch und dabei völlig flimmerfrei zu erfolgen haben. Der anwendungsspezifische Hardware-Grafikcontroller arbeitet deshalb mit einem eigenen Grafikspeicher, der aus zwei Segmenten besteht: dem „accessible video ram“ und dem „refresh video ram“. Dabei wird das Bild zuerst im versteckten Speichersegment aufgebaut und anschließend mit einer raschen Kopierfunktion in das sichtbare Speichersegment transferiert. Das Ergebnis ist eine ruck- und flimmerfreie Ausgabe.

Der Embedded Computer verfügt über ein Echtzeit-Betriebssystem, wobei ein Task eine virtuelle Maschine darstellt, die

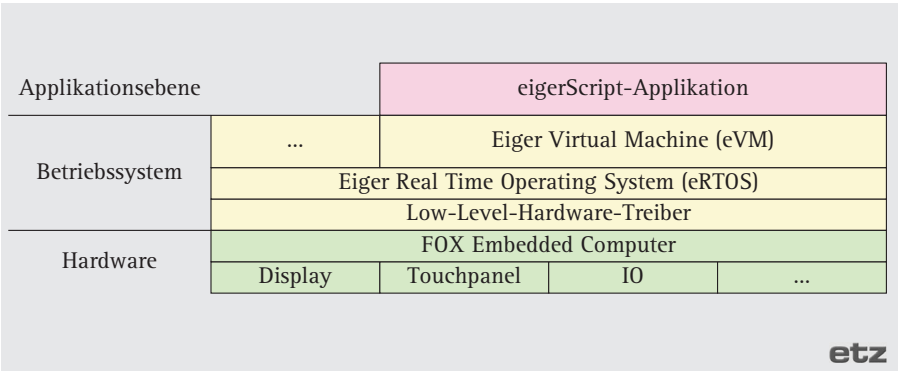

***Bild 3**. System- Architektur des „eigerPanel“*

den kompilierten Anwendungs-Programmcode ausführt. Die gesamte Architektur ist auf das Wesentliche vereinfacht, was trotz sehr geringen Energiebedarfs rasch ablaufende, effiziente Anwendungen ermöglicht (**Bild 3**). Die Zeit, die der Rechner zum Kaltstart inklusive Laden des Anwendungsprogramms benötigt, ist mit 3 s im Vergleich zu IPC-Systemen sehr kurz. Das System hat bei eingeschaltetem Bildschirm samt Hinterleuchtung eine Leistungsaufnahme von weniger als 4 W.

Das System arbeitet vollständig grafisch und hat keine Konsolenbetriebsart. Ein grafisches System wird nicht wie eine Konsole über Befehlseingaben bedient, sondern durch Ereignisse gesteuert. Ein Ereignis ist entweder ein Tastendruck, eine Berührung des Bildschirms, ein Input kennt das „eigerPanel“ eigentlich nur einfache pixelorientierte Zeichenroutinen: Punkte in einer beliebigen Farbe setzen ist dabei die elementarste Funktion.

Die Aufgabe, ein Bild darzustellen, verlangt vom Rechner, eine Datei zu laden und dann auf das Display auszugeben. Um die Dateigröße und damit die Ladezeit zu optimieren, wird ein proprietäres komprimiertes Verfahren verwendet.

Damit der Rechner auf Befehle reagieren kann, müssen nach einem Druck auf das Touch-Panel je nach Position des Fingers entsprechende Ereignisroutinen aufgerufen werden, welche im Anwendungsprogramm als berührungsempfindliche Flächen, sogenannte Hotspots, definiert sind. Mit dem System ist es bereits möglich, die in der Einleitung beschriebene Anwendung zu realisieren.

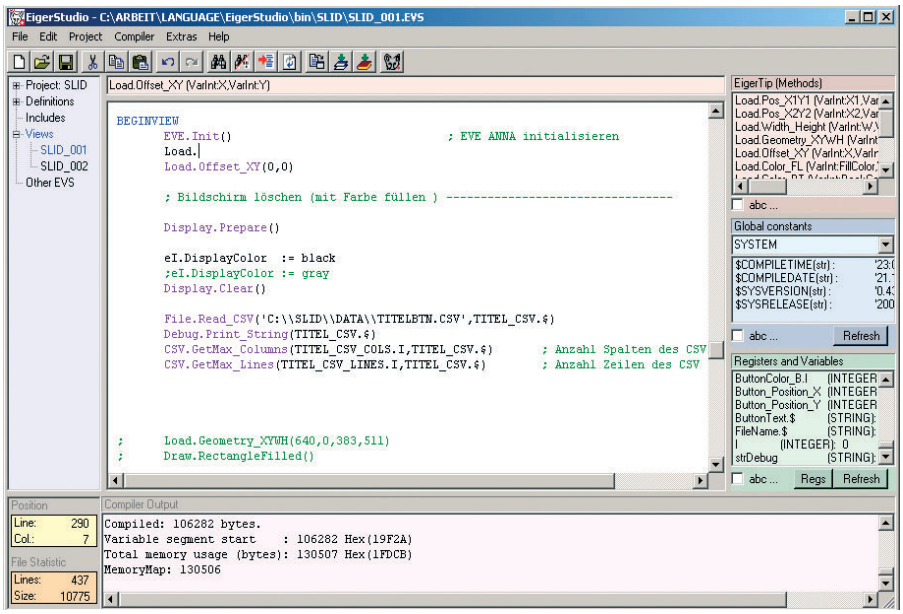

***Bild 4**. Die integrierte Entwicklungsumgebung „eigerStudio“*

über die verschiedenen IO-Kanäle oder eine Meldung von einem Timer. Daraus ergibt sich, dass ein darunterliegendes Echtzeitbetriebssystem auf Ereignisse warten und entsprechend reagieren muss.

Da das Betriebssystem keine Benutzerschnittstelle kennt, lädt der Computer beim Einschalten direkt das Anwendungsprogramm und führt es aus. Der wichtigste Task des Betriebssystems ist demnach die „Virtual Machine“, eine Abstraktionsebene, die eine Vielzahl von Befehlen und Funktionen zur Verfügung stellt. Das Anwendungsprogramm liegt als kompilierter Bytecode vor, der von der virtuellen Maschine interpretiert wird. Dies ermöglicht eine wirksame Fehlerprüfung und damit eine hohe Stabilität der Anwendung. Als grafisches System

Kontinuität dank virtueller Maschine

Eine große Herausforderung bestand darin, eine virtuelle Maschine, die den Programmcode interpretiert, zu definieren und zu bauen. Der wahrscheinlich größte Vorteil einer virtuellen Maschine ist die Unabhängigkeit von der realen Hardware. Sollte es in der Zukunft notwendig sein, die Hardware anzupassen, brauchen die einmal erstellten Anwendungen nicht neu kompiliert zu werden, denn die virtuelle Maschine lässt sich auf neue Maschinen portieren. Ein weiterer Vorteil ist die Abstraktion der realen Maschinenbefehle: Komplexe Funktionen lassen sich so als einen einzigen Maschinenbefehl der virtuellen Maschine realisieren. So lässt sich beispielsweise mit je einem einzigen Befehl ein Kreis zeichnen, eine Zeichenkette über eine Schnittstelle ausgeben oder eine komplette Textbox darstellen. Das Konzept der virtuellen Maschine erlaubt auch eine Skalierbarkeit: Minimalsysteme auf der Basis eines einzelnen Chips lassen sich genau gleich erstellen wie größere Grafiksysteme mit mächtigeren Rechenfähigkeiten. Die eigene Programmiersprache trägt dem optimal Rechnung, indem je nach Zielsystem unterschiedliche Bibliotheken eingebunden werden; der Quellcode selbst bleibt davon unberührt. Mit dem bislang beschriebenen System lassen sich schon vollständige Benutzeroberflächen für einfache Maschinen realisieren. Zusätzlich stellt die virtuelle Maschine auch einen Rechenstack zur Verfügung, der sowohl in ganzzahliger als auch in Fließkomma-Arithmetik arbeiten kann. Damit lassen sich dann auch Anwendungen realisieren, die einen erhöhten Rechenbedarf haben.

Eigene Hochsprache für die Anwendungsentwicklung

Der Entwickler programmiert seine Anwendungen vollständig auf einem Windows-PC und kann dadurch auch weiterhin seine gewohnten Tools nutzen. Speziell bei der Erstellung von Grafiken nutzt jeder Anwender verschiedene Werkzeuge. Die Eiger-Entwicklungssuite enthält ein Konvertierungsprogramm, um Grafiken der gängigsten Bildformate in das proprietäre EGI-Format zu wandeln. Mit einem weiteren Werkzeug, dem Font-Editor, können auf einfache Weise neue Zeichensätze erstellt werden. Das wichtigste Werkzeug aber ist das „eigerStudio“, eine integrierte Entwicklungsumgebung, mit der die Anwendungsprogramme für das „eigerPanel“ erstellt werden (**Bild 4**).

Die Anwendungsprogramme werden in einer speziell entworfenen Hochsprache „eigerScript“ erstellt. „eigerScript“ lehnt sich an syntaktische Elemente aus Pascal, C und Visual Basic an, ist aber eine reine prozedurale Sprache, die keine Objektorientierung kennt, womit sie nahe an das Konzept der virtuellen Maschine heranreicht und keine Speicherverwaltung benötigt. Die Sprachdefinition ist knapp und übersichtlich, damit die Einarbeitungszeit auch für Mitarbeiter ohne intensive Programmier-Erfahrung möglichst kurz ist. Die Entwicklungsumgebung gliedert rund um das Editorfenster kleine interaktive Informationsfenster, wie Projektbaum, Editorstatus, Compilerausgaben, Register- und Variableninformation, Konstanten- und Methodenübersicht, der virtuellen Maschine. Der Editor

***Bild 5**. Verkaufsautomat mit separatem Werbebildschirm, der ausschließlich auf „eigerPanel“ basiert*

arbeitet mit „live-parsing“, was bedeutet, dass er während der Eingabe das Layout des Quellcodes, insbesondere aber die Einfärbung der Quellcodeelemente anpasst. Ein Projekt besteht aus einzelnen Views, unter denen man sich unabhängige Teilprogramme vorstellen kann, die je eine Bildschirmansicht definieren. Innerhalb eines Projekts können sich Views gegenseitig aufrufen. Wie jede andere Sprache auch kennt „eigerScript“ ein Variablenmanagement, das Zeichenketten, verschiedene Ganzzahlentypen und Fließkommatypen unterstützt.

Anwendung in einem Verkaufsautomat

Eine bereits auf dem Markt eingeführte Anwendung zeigt die Flexibilität des Systems, das mit verschiedenen Bildschirmen erhältlich ist. In einem neuartigen Verkaufsautomat (**Bild 5**) ist die Benutzerschnittstelle durch ein „eigerPanel“ realisiert. Ein weiteres Panel mit Großbildschirm dient als dynamische Werbeplattform. Der bargeldlos ausgeführte Verkaufsautomat ist vollständig über das Internet fernwartbar. Dabei lassen sich sämtliche Betriebsparameter, wie Abverkaufszahlen, Vorratsbestand und viele weitere, auslesen. Die Preise lassen sich dynamisch gestalten, und selbst das Betriebssystem kann online erneuert werden, ohne dass Serviceeingriffe vor Ort notwendig wären. Trotzdem stellt die Verbindung mit dem Internet kein Risiko dar, denn dank des kompakten, proprietären Betriebssystems ist eine Gefährdung durch Schadsoftware praktisch ausgeschlossen.

Literatur

[1] S-TEC electronics AG: www.s-tec.ch
[2] Weitnauer Messtechnik: www.weitnauer-messtechnik.ch
[3] „eigerScript“ Schnelleinstieg: www.eigergraphics.com/download.htm
[4] Box-In-Kiosk: www.boxin.ch