

Industrielle IT

Einfache Migration zu Windows CE

26.11.13 | Autor / Redakteur: Jens Schneeweiss * / Franz Graser



Integrationsplattform Windows: Seit einigen Jahren geht der Trend hin zu einer Vereinheitlichung der bisher in sich geschlossenen industriellen EDV-Systeme. Ein Konsolidierungsvorhaben ist jedoch nicht immer problemlos möglich, da Mikrocontroller und PCs unterschiedliche Anforderungen abdecken. Windows CE bietet sich als Integrationsplattform an. (Bild: Walter Sturm, pixelio.de)

In der industriellen IT ist das Nebeneinander von Mikrocontroller-basierten Systemen und PCs Alltag. Für die Integration beider Welten ist Windows CE aus vielen Gründen eine geeignete Plattform.

Für viele Steuerungen haben sich über die Jahre hinweg individuelle Messwert- und Datenerfassungssysteme sowie Verarbeitungssysteme entwickelt. Solche individuell gewachsenen Systeme bestehen meist aus PC-Prozessrechnern oder -Arbeitsplätzen für Datenverarbeitung und Visualisierung auf der einen Seite sowie auf der anderen Seite aus Geräten auf Mikrocontrollerbasis, die häufig noch über serielle Schnittstellen oder

Industrie-Busse angebunden sind und die Datenerfassung übernehmen.

Seit einigen Jahren geht der Trend hin zur Vereinheitlichung solcher industriellen EDV-Systeme. Ein Konsolidierungsvorhaben ist jedoch nicht immer problemlos möglich, da beide Plattformen, nämlich Mikrocontroller und PCs, unterschiedliche Anforderungen abdecken. Die Mikrocontroller der Erfassungsgeräte sind oftmals in C programmiert, während die Visualisierungssysteme eher mit Sprachen wie Delphi, C++, Visual Basic 5.0 oder 6.0, arbeiten.

In solchen Fällen bietet sich Windows CE als Integrationsplattform an, da es die Echtzeitanforderung von Mikrocontrollern sowie Anforderungen an die GUI erfüllt und eine verhältnismäßig einfache Portierung des Quellcodes erlaubt. Ein weiterer Vorteil ist die Nutzung von innovativen Low-Level-Treibern, die die Entwicklung bzw. den Zukauf weiterer Hardware nicht erfordern. Außerdem besteht mit Windows CE bei der Erarbeitung eines Portierungskonzeptes die Möglichkeit, mit den gängigen Sprachen

(Delphi, C++, Visual Basic) zu arbeiten.

CE-Smartphones und Pocket-PCs eignen sich zur Evaluation

Am Anfang eines Konsolidierungsvorhabens steht allerdings auch die aufwändige Suche nach einer optimalen Windows-CE-Systemplattform. Es empfiehlt sich zunächst, ein bereits fertig integriertes System zu nutzen, um die Anforderungen und möglichen Schwierigkeiten im Betrieb mit der gewünschten Windows-CE Zielplattform besser beurteilen zu können. Dieses Vorgehen erlaubt einen raschen und zielgerichteten Entwicklungsbeginn der Integrationslösung.

Da PCs fester Bestandteil von Automatisierungssystemen sind, die klassische Aufgaben eines Prozessrechners übernehmen, muss auch die neue Plattform über diese Funktionen verfügen. Zu diesen Aufgaben gehören u. a. die Datenerfassung, die Steuerung und Regelung, die Protokollierung und Visualisierung der prozessrelevanten Daten sowie die Speicherung dieser Daten. Die neue Plattform sollte zudem die Fähigkeit besitzen, mit Maschinen und anderen Rechnern über diverse Anschlüsse und Schnittstellen direkt zu kommunizieren. Die neue Software muss demnach den Echtzeitanforderungen der Schnittstellen bzw. der Messwerterfassung nach außen genügen. Eventuell sind hier noch Mikrocontroller nötig, die allerdings nur noch zu Pegelanpassung bzw. Protokollkonvertierung genutzt werden.

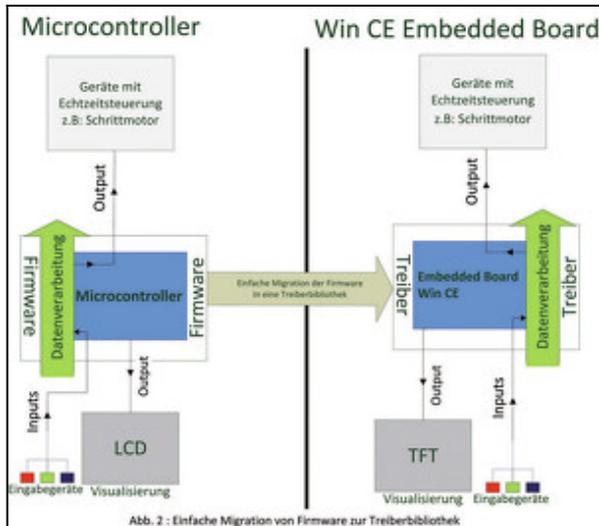
Durch die fortschreitende Miniaturisierung sind mittlerweile PDAs oder Pocket-PCs zahlreich auf dem Markt zu finden. Von Anfang an waren sie mit den klassischen Funktionen eines PCs ausgerüstet, die wichtigsten Anwendungen sind unter dem Begriff PIM-Software (Personal Information Manager) zusammengefasst. Heute lassen sich Pocket-PCs auch zur mobilen und kabellosen Datenerfassung nutzen und können in bestimmten Szenarien übliche Standardcomputer ersetzen.

Ein guter Teil dieser Geräte ist mit den Betriebssystemen, Programmierumgebungen und Software-Entwicklungstools ausgerüstet, wie man sie von PCs kennt. Daher bestehen umfangreiche Möglichkeiten, Softwaremodule von vorhandenen PC-basierten Systemen zu übernehmen. Sinnvoll für eine produktive Entwicklungsarbeit ist dabei die Nutzung von Microsofts .NET-Plattform für die vorher auf dem PC betriebenen Applikationen.

Zudem sind viele Smartphones mit den Betriebssystemen, Programmierumgebungen und Software-Entwicklungstools ausgerüstet, die bereits von PCs oder Pocket-PCs bekannt sind. So ist der Weg vorgezeichnet, Smartphones bei kleineren kabellosen Prozessrechneranwendungen als günstige Evaluationsplattformen für die Entwicklung des geeigneten Systems zu verwenden. Insbesondere beim Einsatz in großen Betrieben mit entsprechenden Datenmengen oder bei Arbeiten unter harten

Umgebungsbedingungen müssen solche prototypischen Smartphones allerdings möglichst rasch durch passendere Plattformen/Geräte ersetzt werden. Als solide Basis für die Konsolidierung eignet sich demnach eher die moderne PDA/Pocket-PC-Generation.

Vom Mikrocontroller-Code zur Treiberbibliothek



Die Migration der Mikrocontroller-Funktionen ist verhältnismäßig einfach möglich. Um die Mikrocontroller-Quellcodes zu nutzen, müssen sie in das Entwicklungssystem des Windows-CE-Systems übernommen werden. Diese Funktionen werden in eine Treiberbibliothek umgewandelt und können dann entweder mithilfe des Platform Builders bei Erstellung des Betriebssystems integriert oder über ein externes Modul (etwa per USB-Stick, CF- oder SD-Speicherkarte) in das Windows-CE- bzw. Windows Mobile/Pocket-PC-System geladen

werden.

Es handelt sich dabei um Stream-, Filter- und Kerneltreiber, die z. B. proprietäre Protokolle und Übertragungsraten ermöglichen. Die bestehenden C/C++-Quellcodes werden unter Windows CE weiter genutzt. Die Schnittstellen werden meist über eine Wrapper-Assembly als Application Programming Interface (API) in das .NET Compact Framework eingebunden und über eine Mobile-Geräte-Applikation genutzt.

Die Entwicklung einer solchen Treiberbibliothek verläuft in mehreren Schritten. Zunächst müssen die geforderten Schnittstellen unter Berücksichtigung der erforderlichen Protokolle evaluiert werden. Danach werden die bestehenden Quellcodes auf ihre weitere Verwendbarkeit überprüft. Diese Analyse der bestehenden Quellcodes erfolgt mit dem Ziel, ein Portierungskonzept zu erstellen. Im Detail bedeutet dies die Ergründung der Softwarestruktur, Evaluierung des Datenflusses und Auswahl der benötigten Quellcodes, um einen Ansatz für den Übergang zu .NET zu bestimmen.

Bei der Erstellung dieses Konzeptes müssen diverse Anforderungen wie eine Pufferung der Telegramme oder der Bedarf an asynchronen Datenaufrufen beachtet werden. Bei paralleler Nutzung des Systems durch mehrere Applikationen können

auch Multiple-Access-Treiber notwendig sein. Auf der Ebene der Protokollverarbeitung wird zudem häufig ein objektorientiertes Softwaredesign genutzt, besonders bei sehr komplexen Protokollen in den höheren Protokollschichten oder wenn mehrere Treibervarianten gleicher Art genutzt werden. Ein .NET Visual Studio Control für die Toolbox ermöglicht dabei eine einfache Integration in solche Softwareprojekte.

Die anschließende Portierung findet schrittweise statt, indem Quellcodes auszugsweise in CE-Projekten nutzbar gemacht werden. Dabei werden WinAPI-Zugriffsschichten adaptiert. Um die Treiber zu testen, werden systemnahe Prüfapplikationen für den Aufruf durch alle Protokollschichten benötigt. Erst nach erfolgreichem Test der Treiber mit den systemnahen Prüfapplikationen werden Wrapper-Assemblies für die Nutzung im .NET Compact Framework erstellt. Abschließend werden alle Softwareschnittstellen mit einer entsprechenden Applikation getestet.

Die Vorteile durch solche Gerätetreiber, die für den individuellen Bedarf optimiert sind, liegen in der fehlerfreien, direkten Kommunikation mit externen Geräten (ohne Umwege über die WinAPI-Schnittstelle gehen zu müssen) und der Nutzung von Gerätetreibern als Standardschnittstelle für Treiberschichten. Zudem bieten sie ein mögliches Monitoring-Interface zum Debuggen der Schnittstellen-Kommunikation.

Portierung zu Windows CE leicht gemacht

Neben den Funktionen der Mikrocontroller müssen die notwendigen Anwendungen von den veralteten Workstation-PCs übernommen werden. Da Windows CE auf der Win32-API basiert, ist es möglich, desktopbasierte Applikationen, die für ein anderes Windows Betriebssystem entwickelt wurden, ohne große Änderungen auf eine mobile Plattform zu portieren. Allerdings sollte man einige Besonderheiten von WinCE beachten, um ein optimales Ergebnis zu erhalten:

Windows CE unterstützt nur einen Teil der 32-Bit- und keine 16-Bit APIs. Es kann demnach passieren, dass einige APIs durch neue ersetzt oder mithilfe der Funktionen in Windows CE umdefiniert werden müssen. Wieder andere Features wurden an die kleinere Benutzeroberfläche der Windows CE Plattformen angepasst. So wurden etwa die Werkzeug- und Menüleiste zu einer Befehlsleiste zusammengelegt. Zugleich bietet das CE-Betriebssystem eine Reihe spezieller Features (z. B. für Touchscreens oder Barcodescanner), die eine Anpassung an eine mobile Plattform wie Pocket-PC oder Smartphone vereinfachen.

Windows CE nutzt Unicode-Zeichensätze, weshalb alle nicht in Unicode programmierten Applikationen zur Portierung modifiziert werden müssen. Hierbei müssen alle Strings der Chars zu *WIDECHARS* konvertiert werden (etwa zum

variablen C-Typ *wchar_t*). Man kann für bestimmte Befehle auch das *TCHAR*-Format nutzen, um die Programmierung sowohl in Unicode als auch in ASCII kompilieren zu können.

Zudem ist Windows CE ein ROM/RAM-basiertes Betriebssystem und unterstützt standardmäßig keine internen Festplatten. Daher sollte bei der Portierung der Applikationen darauf geachtet werden, dass die einzelnen Applikationen ohne hohe RAM- und Hardwareanforderungen funktionieren, denn die Hardware mobiler Geräte wie Pocket-PCs oder Smartphones, ist gegenüber Desktop-PCs begrenzt.

Auch wenn Windows CE nicht den vollen Umfang von Win32-APIs unterstützt, ist die Portierung von vorhandenen Applikationen mit verhältnismäßig geringem Aufwand verbunden. Die Entwickler-Tools sind von den Win32-Plattformen her bekannt und ebenso komfortabel. In Kombination mit den erwähnten Treiberbibliotheken kann demnach der Aufwand bei Integrationsvorhaben und der Portierung von Legacy Applications erheblich reduziert werden.

Dieser Weg zur Vereinheitlichung des internen Netzwerkes ist für die überwiegende Mehrheit der Betriebe die sinnvollste Lösung. Die Abschaffung redundanter Prozesse und der Umstieg auf ein schlankeres System bringen deutliche Zeit- und Aufwandsersparnis. Die Umsetzung eines solchen Integrationsprojektes ist in kurzer Zeit möglich.

Zudem benötigen die Mitarbeiter nur wenig Nachschulung, weil sich die Grundlagen des Betriebssystems und der beibehaltenen Abläufe nur geringfügig ändern. So sollten auch Betriebe, die bisher vor dem vermeintlich hohen Aufwand der Systemoptimierung zurückschreckten, ernsthaft über den Umstieg nachdenken.

* Jens Schneeweiss ist Diplom-Ingenieur Elektrotechnik. Er leitet das Ingenieurbüro Technische Systemprogrammierung Jens Schneeweiss in Herten.

ERGÄNZENDES ZUM THEMA

- ▼ Windows CE im Profil
- ▶ Windows CE im Profil

Windows CE ist das Betriebssystem von Microsoft, das speziell für mobile und Embedded-Geräte entwickelt wurde. Zur Windows-CE-Familie zählten und zählen verschiedene Windows-Mobile-Versionen, die bei Smartphones oder PDAs mit integrierter Telefonie zum Einsatz kommen. Das aktuelle Mobil-Betriebssystem von Microsoft, Windows Phone 8, basiert allerdings auf dem Windows-NT-Kernel. Das System unterstützt Prozessoren mit Intel x86-, MIPS-, ARM -und SuperH-Architektur und ist seit der Version 6.0 ein Echtzeitbetriebssystem.

Copyright © 2014 - Vogel Business Media

Dieser Beitrag ist urheberrechtlich geschützt.
Sie wollen ihn für Ihre Zwecke verwenden?
Infos finden Sie unter www.mycontentfactory.de.

Dieses PDF wurde Ihnen bereitgestellt von <http://www.elektronikpraxis.vogel.de>

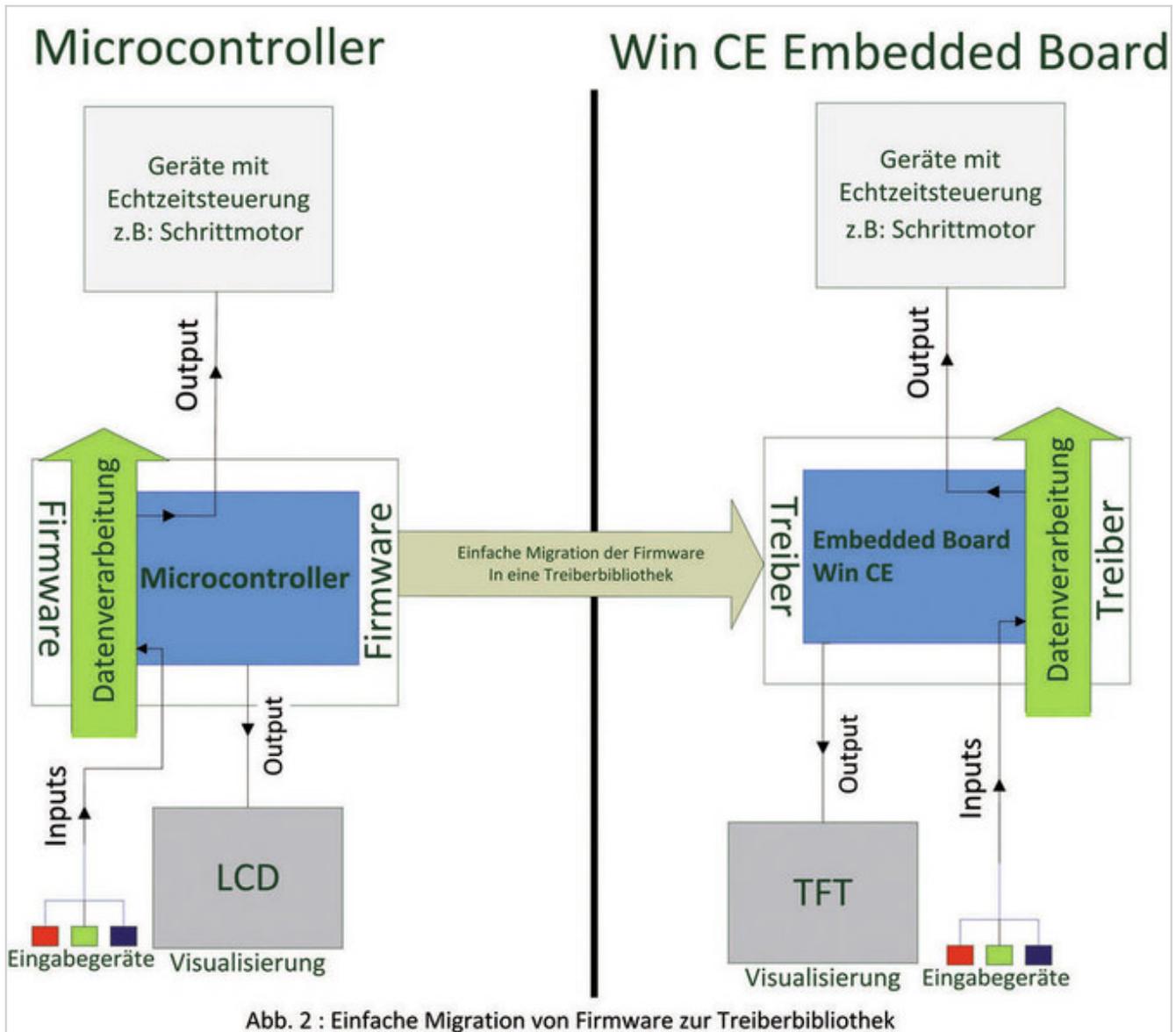


Abb. 2 : Einfache Migration von Firmware zur Treiberbibliothek

Von C zu CE: Einfache Migration von Firmware zur Treiberbibliothek (Grafik: Jens Schneeweiss)