

Touchscreens mit Web-Technologien umsetzen



Web-basierte grafische Benutzerschnittstellen für Touchscreen-basierte Embedded- Systeme? Vor nicht allzu langer Zeit wäre das mangels entsprechender Technologien nicht möglich gewesen. Dies hat sich dank moderner Mikrocontroller sowie hochoptimierte Software-Komponenten geändert.

Wenn Embedded Systeme mit einer grafischen Benutzerschnittstelle ausgestattet werden sollen, so bieten sich dem Entwickler eine Vielzahl von Möglichkeiten diese zu implementieren. Mehrere kommerzielle, wie freie und quelloffene Toolkits stehen zur Auswahl – von einfachen ressourcenschonenden Bibliotheken bis hin zu hochkomplexen aus dem Desktop-Bereich stammenden Toolkits. Eine interessante Alternative zu diesen traditionellen Toolkits stellen Web-basierte Benutzerschnittstellen dar. Web-basierte Benutzerschnittstellen sind mittlerweile Standard bei netzwerkfähigen Embedded Systemen. Sie werden vor allem zur Konfiguration und Überwachung eines Systems über einen auf einem PC laufenden Web-Browser eingesetzt. In den meisten Fällen werden recht einfache, Formular-basierte statische Seiten verwendet.

Bei den Desktop Applikationen erfreuen sich Rich Internet Applications zunehmender Popularität. Dabei handelt es sich um Applikationen, deren Benutzerschnittstelle mit Hilfe eines Web-Browsers dargestellt wird. Diese Benutzerschnittstelle hat aber dennoch mehr Ähnlichkeit mit einer traditionellen Desktop-Applikation als mit einer Webseite. Dabei wird die Benutzerschnittstelle mit HTML, CSS und JavaScript realisiert. Der zur Darstellung notwendige Browser ist oft direkt in die Applikation oder in die Laufzeitumgebung integriert. Die Applikationslogik kann mit JavaScript betriebssystemunabhängig realisiert werden. Entwicklungs- und Laufzeitumgebungen wie Adobe AIR und Appcelerator Titanium ermöglichen eine einfache Entwicklung von Applikationen, die dann unter Windows, Mac OS X oder Linux ablauffähig sind.

Dieser Trend lässt sich auf Embedded Systeme übertragen. Beispielsweise wurden iPhone-Applikationen in den Anfangstagen des iPhones ausschließlich mit JavaScript, HTML und CSS verwirklicht. Auch Applikationen für das Smartphone Palm Pre werden so realisiert, ebenso wie die Benutzerschnittstelle von IPTV Set-top Boxen. Warum sollte man also nicht Benutzerschnittstellen für Embedded Systeme in dieser Form entwickeln. Dazu ist ein auf dem Gerät laufender Web-

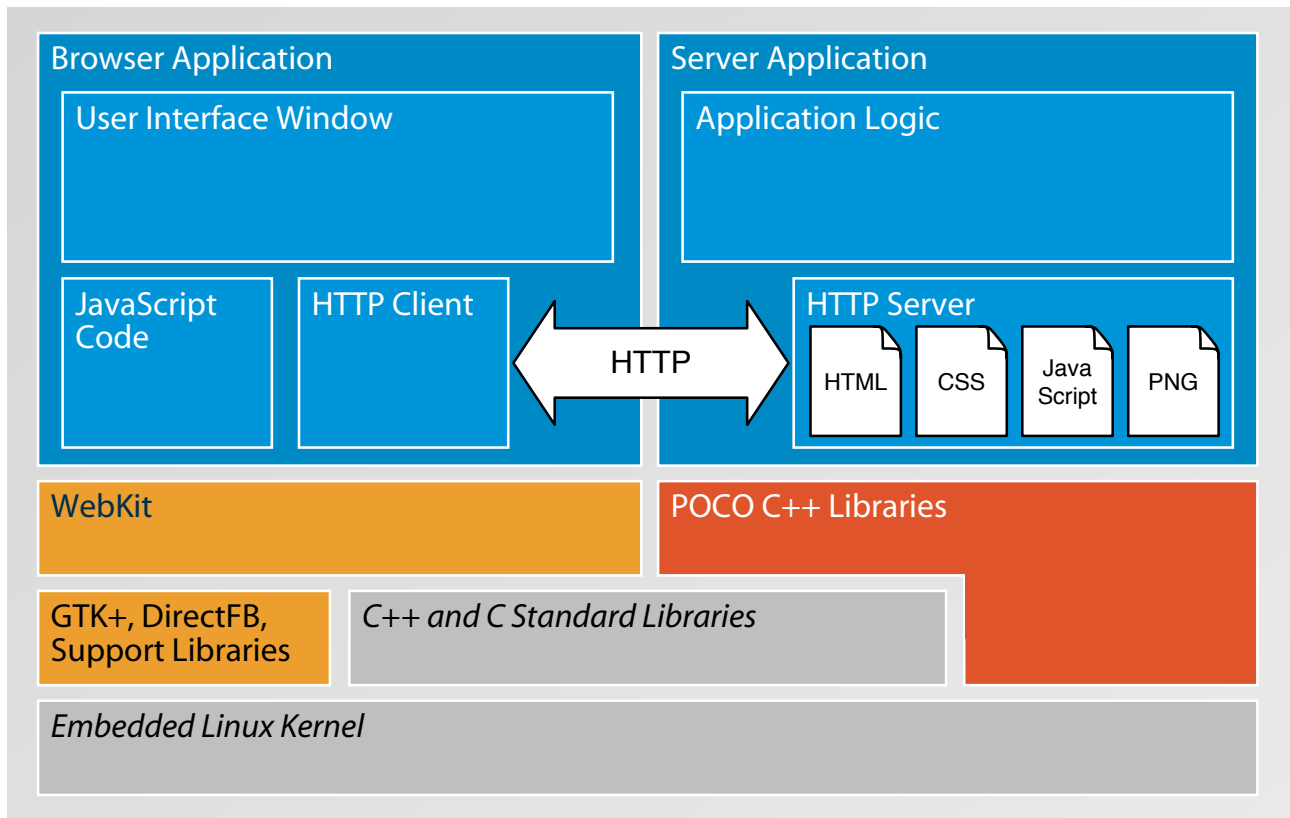
Vorteile Web-basierter Benutzerschnittstellen

- > Ein Prototyping der Benutzerschnittstelle ist sehr einfach am PC möglich – noch lange bevor die eigentliche Zielhardware existiert.
- > Die Benutzerschnittstelle kann von einem darauf spezialisiertem Web-Designer entworfen, sogar teilweise oder vollständig implementiert werden.
- > Mehrsprachigkeit oder die Lokalisierung der Benutzerschnittstelle sind sehr einfach möglich. Ebenso ein Branding oder die Anpassung des Designs ans eigene Corporate Design.
- > Der Entwickler ist bei der Gestaltung der Schnittstelle nicht an ein vom Toolkit vorgegebenes Design gebunden. HTML, CSS und die geschickte Verwendung von Bildern ermöglichen die Entwicklung von sehr ansprechenden Interfaces.
- > Die Benutzerschnittstelle kann lokal am Touch-Screen des Gerätes als auch von einem entfernten PC aus über einen Web-Browser bedient werden. Die Bedieneinheit kann leicht vom eigentlichen System getrennt werden. Somit kann eine Bedieneinheit auch zur Steuerung mehrerer Systeme verwendet werden. Die Kommunikation zwischen Bedieneinheit und System erfolgt über Ethernet oder Wireless LAN.

Server und ein im Vollbildmodus laufender Web-Browser notwendig.

WAS SIE FÜR DIE UMSETZUNG AUF IHREM SYSTEM BENÖTIGEN

Wer eine Benutzerschnittstelle einrichten möchte, braucht zuerst einen leistungsfähigen, JavaScript-fähigen Web-Browser. Hier bietet sich die frei verfügbare WebKit Engine (<http://webkit.org>) an. Dabei handelt es sich allerdings um keinen vollständigen Web-Browser wie Firefox, Safari oder Internet Explorer, sondern lediglich eine Sammlung von Bibliotheken, welche alle für einen modernen Web-Browser notwendigen Funktionen bereitstellt. Basierend auf WebKit kann leicht eine eigene Browser-Applikation gebaut werden, die im Vollbild-Modus läuft. Der Anwender merkt gar nicht, dass die Benutzerschnittstelle des



Gerätes eigentlich eine Webseite ist. Neben der Browser-Komponente braucht man einen Web-Server.

Für die Entwicklung ansprechender Web-basierter Benutzerschnittstellen ist die Verwendung von Ajax Programmieretechniken notwendig. Dies bedeutet, dass Teile der Applikation in JavaScript implementiert sind, und im Browser ausgeführt werden. Der JavaScript Code verwendet das HTTP Protokoll, um Daten im Hintergrund von der Server-Applikation abzuholen oder um Daten an die Server-Applikation zu senden.

Je nach Applikation können dabei sehr viele Anfragen zusammen kommen. Sollen zum Beispiel Prozessparameter in Echtzeit dargestellt werden können, so sind mindestens zwei Updates pro Sekunde erforderlich. Ein CGI-basierter Web Server, wie er bei statischen Web-Interfaces oft eingesetzt wird, bringt ein Embedded System hier schnell an seine Grenzen. Warum ist das so? Weil zum Beispiel auf einem Linux-basiertem System für jede Anfrage vom Client ein neuer Prozess gestartet werden muss. Eine hervorragende Alternative hierzu ist ein Web-Server, der direkt als eigener Thread innerhalb der Embedded Applikation läuft. Dies löst auch gleich das Problem, wie die Prozessdaten von der Applikation zum Web-Server und somit zum Client kommen.

Häufig bei CGI Programmierung verwendete „Krücken“, wie die Verwendung von Dateien in einer RAM-Disk zum Austausch von Daten zwischen Web-Server und Applikation sind nicht notwendig, da der

Web-Server über eine Programmierschnittstelle unmittelbaren Zugriff auf die Applikationsdaten hat.

SCHNELLER MIT DEN POCO C++ LIBRARIES

Für die Implementierung eines solchen Systems sind die freien und quelloffenen POCO C++ Libraries (<http://pocoproject.org>) ideal. Neben einer Vielzahl von anderen nützlichen Features bieten die POCO C++ Libraries einen Web-Server, der direkt in die Applikation eingebunden werden kann und dort als eigener Thread läuft. Ein in den POCO C++ Libraries enthaltenes Tool, der Page Compiler, ermöglicht es, C++ Code direkt in HTML Seiten einzubinden, wie man es von PHP, Java Server Pages (JSP) oder ASP kennt, und daraus C++ Code zu generieren.

Das vereinfacht die Entwicklung dynamischer Webseiten in C++ wesentlich. Diesen Lösungsansatz bestehend aus den POCO C++ Libraries und WebKit hat Applied Informatics bereits in mehreren Embedded Linux-basierten Projekten erfolgreich eingesetzt.

KONTAKT

Applied Informatics Software Engineering GmbH
St. Peter 33
9184 St. Jakob im Rosental
Austria

T +43 4253 32596 F +43 4253 32096
info@appinf.com | www.appinf.com