Linux-based onboard "gateway server" remotely pilots model airplane

by Telly Cooper (Sep. 12, 2003)

This whitepaper from Intrinsyc describes how engineers at IBM's Pervasive Computing Advanced Technology Laboratory created an embedded Linux based intelligent, high-level remote control system for a model airplane as a way to showcase Gateway Server technology.

Using an Embedded Gateway Server to Remotely Pilot a Model Plane

Intrinsyc Software has helped pioneer the development of intelligent connected devices with the CerfBoard reference platform. Intrinsyc has developed embedded platforms that target headless and display based applications running on a variety of processors and has developed a custom Linux distribution to take full advantage of the CerfBoard's capabilities. With the addition of IBM's WebSphere Micro Environment's Java Powered Runtime Environment, the CerfBoard has evolved into a powerful, versatile embedded gateway server. This remarkably small gateway server can be used to remotely administer and monitor any connected smart devices.

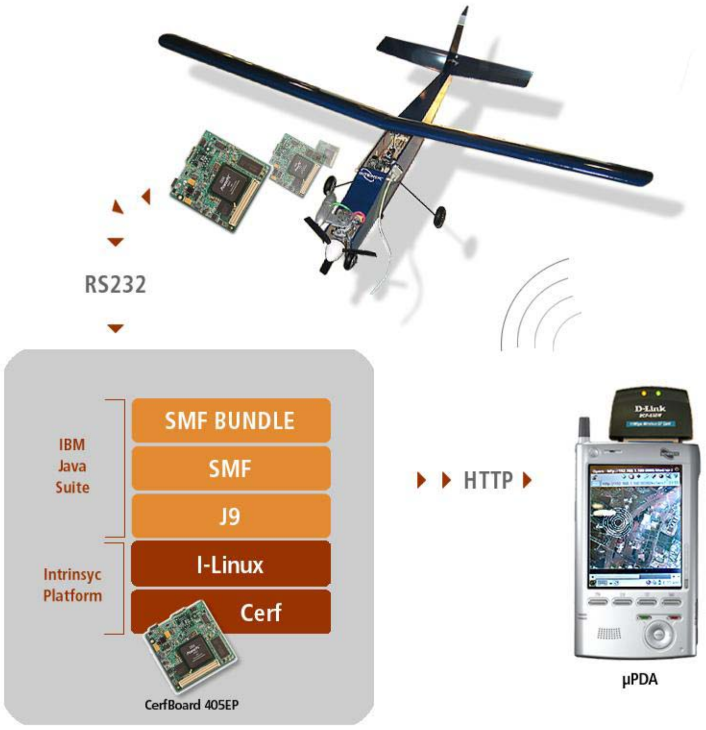
A "gateway server" is a system that acts as a bridge between the Internet and an intranet of local devices, allowing you to talk to and control any connected device. For example, a home gateway server could connect to various devices in the home to allow you to synchronize your lawn sprinklers to weather reports on the Internet. From your wireless PDA, you could turn on lights, set your VCR, monitor your alarm system, or even fly a plane. The range of applications is virtually unlimited.

With the proliferation of embedded smart devices in the home and small office, a standard framework for connection and integration is required. This is achieved with the use of Open Services Gateway Initiative (OSGI) protocols and APIs. OSGi has standardized a collection of Java APIs that define the services gateway. Java technology is used because it is an open technology that runs on a wide variety of platforms.

Pilot a plane with a PDA?

Engineers at IBM's Pervasive Computing Advanced Technology Laboratory have created an application that uses a small Linux based single board computer to replace the standard radio controls in a radio controlled model airplane. The idea was extended to use a Java layer allowing the user to interface with the airplane controls using a standard wireless enabled PDA. The CerfBoard 405EP was chosen for this application because of its ability to showcase the power of gateway server technology.

The Linux based embedded computer that is used is the IBM PowerPC 405EP Embedded Processor based CerfBoard 405EP from Intrinsyc Software. The CerfBoard 405EP runs Intrinsyc's Linux distribution (I-Linux) and IBM's WebSphere Micro Environment Java Virtual Machine and Services Management Framework (SMF) to function as an embedded Services Gateway. An SMF bundle integrates the user interface and the Java layer interaction with the Linux operating system.



Process Diagram

The CerfBoard 405EP is secured in the cockpit of the airplane and integrated into the legacy control system of the radio controlled (RC) model airplane. The CerfBoard 405EP is powered by a 5V regulated battery pack which is also secured in the cockpit.

The legacy radio control module and the CerfBoard 405EP are connected to a wireless switching module on a dedicated frequency. This allows the user to switch between computer controlled flight and manual flight. The airplane's flight path is manipulated by the following components:

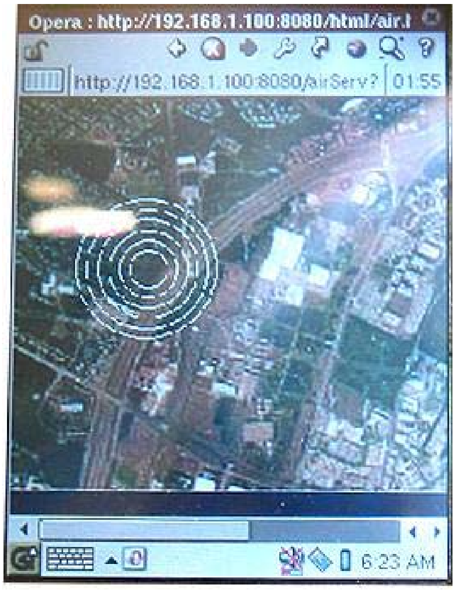
- Ailerons on the airplane's wings change the body roll (rotate the plane body)
- Elevators on the horizontal stabilizer adjust the airplane's height
- The rudder on the vertical stabilizer changes the yaw (side to side motion)

Servo motors connected to the ailerons, elevators and the rudder are controlled by a dedicated integrated circuit that accepts commands via a standard RS232 connection. The Java layer on the CerfBoard 405EP is interfaced with the RS232 port through a native shared library. The communication protocol for manipulating the airplane components in the legacy system was ported to the SMF bundle on the CerfBoard 405EP.

The SMF bundle provides a Java servlet that can be accessed from any Java enabled browser. The servlet displays a LandSAT map of a chosen site and is accessed through the CerfBoard 405EP's URL. When a specific point on the map is selected the target coordinates are compared to the plane's actual position. The bundle then issues the serial commands to guide the plane to the new coordinates. Once the landmark is reached the plane circles it until updated orders are received.

A global positioning system (GPS) and independent gyros are used to maintain synchronization between the virtual and physical positioning of the plane and allow real-time flight path correction. The current position of the plane is determined by the onboard GPS module and the gyros are integrated to indicate horizon and heading information as well as determine when the plane should terminate a turn. The navigation software is implemented as an SMF Bundle with native library support to interface with the GPS module and the gyros. The collected data is plugged into positioning algorithms that compare the actual state of the plane with the intended flight path and actuate the servos based on the result.

IBM's Pervasive Computing Advanced Technology Laboratory has developed a 3-dimensional virtual flight environment to track the plane's flight. This environment is based on LandSAT maps and gives the user a virtual view of the flight from the cockpit of the plane. The position information such as altitude, location, and heading are transmitted to a Linux desktop and are rendered as a 640x480 virtual representation of the terrain. This flight view can be augmented by real-time updates from an onboard camera.



Opera : http://192.168.1.100:8080/html/air.l

http://192.168.1.100:8080/airServ?

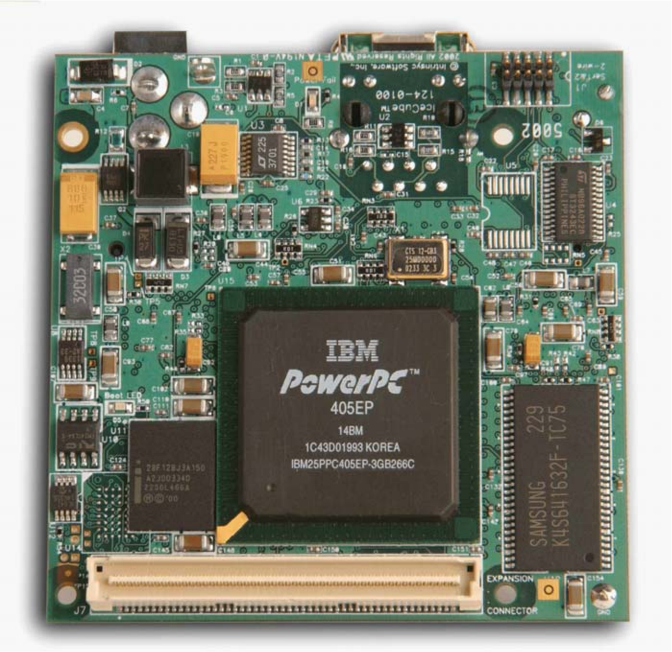
Screenshot from PDA controlling airplane

Communication between the PDA and the onboard CerfBoard 405EP is achieved through cellular communications. The cellular module integrates with the CerfBoard through an RS232 connection. An 802.11b connection is used for near field communications and software updates. The CerfBoard 405EP uses a miniPCI wireless card in 128-bit encrypted infrastructure mode. Alphanumeric connections are also being tested for remote communications with the plane.

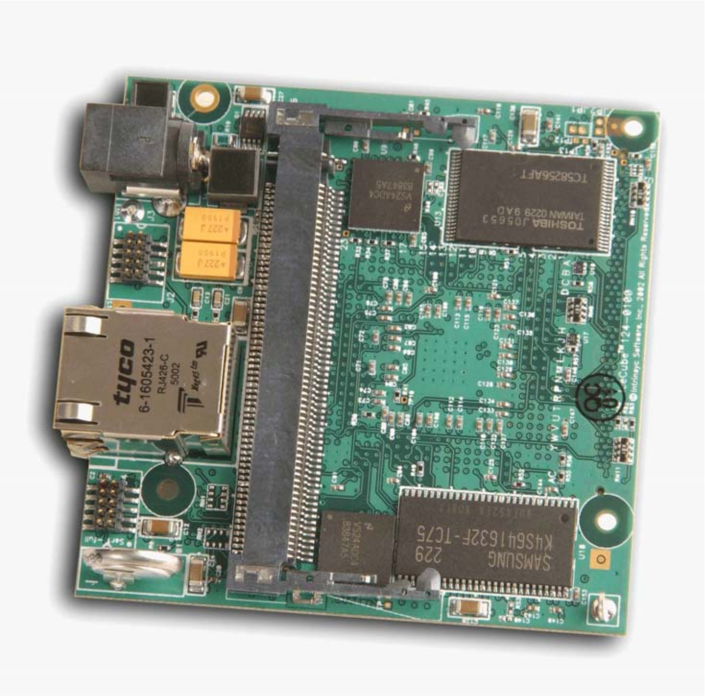
The client side is a PDA with a Java enabled browser, such as Intrinsyc's uPDA reference platform or the Sharp Zaurus. The PDA acts as a thin client connected to the CerfBoard 405EP server through an 802.11b connection and cellular connections. No other applications need to be run on the PDA.

CerfBoard 405EP - A Tiny Server

The CerfBoard 405EP is built around the IBM PowerPC 405EP Embedded Processor. The IBM PowerPC 405EP 32-bit RISC processor is designed to provide a flexible hardware solution to satisfy the demands of high-performance embedded applications. Implemented in the scalable PowerPC Architecture, the 405EP processor maintains code compatibility with other PowerPC processors for ease in migration and fast time-to-market.



Intrinsyc's 405EP-based board, frontside



Intrinsyc's 405EP-based board, backside

An optimized balance of performance, low power, and features makes the PowerPC 405EP an ideal solution for communication, data storage, and pervasive computing applications.

The CerfBoard 405EP includes 32MB of NAND Flash and 32MB SDRAM. The memory configuration provides significant resources for data movement and processing. The I-Linux OS uses less than 5MB and the WebSphere Micro Environment and SMF stack use an additional 5MB of memory. This leaves considerable resources for alternate functions and applications.

Among the exposed peripherals are two 10/100BaseT Ethernet ports driven by an internal Ethernet controller. One is broken out in an RJ45 connector and the other is exposed on expansion pins. Two RS232 UART ports are also exposed through expansion pins.

The 32-Bit PCI V2.2 is exposed through a miniPCI connector. The type III miniPCI form factor ensures compatibility with most laptop Ethernet/Modem and wireless cards.

In addition to the peripherals, the CerfBoard 405EP also exposes the 16-bit address bus and the 16-bit data bus on an expansion connector. This allows for simple integration of alternative function custom daughter cards.

The 3" x 3" x 6/8" CerfBoard 405EP is designed for solutions with size constraints.

Why Linux?

Linux was chosen for the CerfBoard 405EP for its versatility and configurability. It is the ideal operating system for a resource constrained, headless device. In addition, IBM's WebSphere Micro Environment VM and SMF stack integrate seamlessly with Linux to provide a powerful remotely accessible control layer.

I-Linux

Intrinsyc Software has leveraged open source initiatives to create a custom Linux distribution for the CerfBoard. By choosing a freely distributed kernel and filesystem, they have ensured user participation in the development and proliferation of a free, feature rich PowerPC 405EP distribution.

PPCBoot 2.0 was modified to include basic NAND support. Developers integrated an mtd block driver into the kernel and ported it to the bootloader.

The kernel is a 2.4.21-pre2 core that is modified to add in CerfBoard 405EP specific features such as the second Ethernet support and JFFS2 support on NAND Flash. As the CerfBoard 405EP exposes a miniPCI slot, support for Ethernet and Modem cards as well as HostAP support for wireless cards is included in the kernel.

The filesystem is modified from the Familiar Project. This allows the CerfBoard to run a well tested distribution with many key features such as the `ipkg` packaging tools. This packaging tool is converted from the Familiar Project's PXA250 based version and allows for simplified installation of drivers and applications. Debian compatibility provides a native development environment for NFS allowing developers to easily integrate any available Debian packages.

Intrinsyc Software is working toward building a development community to support and distribute PowerPC 405EP based driver and application packages. Intrinsyc hopes to build a strong repository of PowerPC 405EP based applications and drivers through active participation of in-house developers and external users. The key to the success of the Linux based PowerPC 405EP will be the simplified integration and development time for supported platforms.

WebSphere Micro Environment (WME)

The RC model airplane application demonstrates the complementary nature of Linux and Java technologies. While Linux provides the backbone and OS functionality, the Java class libraries provide the remote capabilities and graphical interface with the CerfBoard. The portable nature of Java technology allows an application to be used on any Linux platform regardless of the development machine.

The Java layer that runs on the CerfBoard 405EP consists of a Java Virtual Machine and a gateway service framework called Services Management Framework.

The WebSphere Micro Environment provides a virtual machine that is optimized to run on embedded platforms. It is designed to be a modularized VM to allow a range of feature options. Size constraints are often paramount in embedded platforms so the modular nature of Linux and the WebSphere Micro Environment VM allow the developer to include only necessary components. Configuration of WME can be achieved through a number of methods:

- WME Configuration options
- Native Library selection
- Use of JXE

WebSphere Micro Environment delivers a Java Powered runtime environment for J2ME applications. Configurations define the set of Java programming language features, the set of virtual machine features and a set of supported class libraries and application programming interfaces (APIs). The Connected Device Configuration (CDC) is designed for systems with 2MB or more of total memory and supports the following Java packages:

- java.lang - Java VM system classes
- java.util - underlying Java utilities
- java.net - Universal Datagram Protocol (UDP) datagram and input/output (I/O)
- java.io - Java file I/O
- java.text - very minimal support for internationalization
- java.security - minimal fine-grain security and encryption for object serialization

A configuration combined with a profile creates a J2ME runtime environment. The profile defines the layer that contains the user manipulated APIs. For the airplane application the CDC configuration along with the Foundation profile is used. The Foundation profile includes support for all the core VM features and native shared libraries and has a footprint of 3.7MB.

The native libraries allow the VM applications to access hardware components through Java commands. This eliminates the use of JNI to access the hardware components and simplifies Java application development.

WebSphere Micro Environment JXE files allow the developer to include only the required components of the classes.zip file. If the requirements of the deployed VM are known JXE files are ideal for reducing the VM footprint.

WSDD

WebSphere Studio Device Developer is designed to integrate with the WebSphere Micro Environment runtime and Service Management Framework to simplify development targeting these platforms. Plug-in modules provide the complete environment for developing, testing, and deploying SMF applications.

Why OSGi?

The Open Services Gateway Initiative has created the specifications for the delivery of services over the internet to local networks and devices. The specifications are designed to utilize and manage the growing applications of home automation, residential networks, and entertainment services. The standardized framework for Services Gateways exposes a common backbone for applications to plug into and register their services for other applications to use.

A Services Gateway integrates previously segregated domains such as telephony or data networking, cable services, entertainment, mobile commerce, energy management and Telematics. These devices can be bundled and managed from a common management node exposing a single console or interface.

OSGi leverages Java implementation to ensure cross platform compatibility. The portable nature of Java technology allows multiple devices to be seamlessly integrated into a single uniform network. This means any Java enabled device can be managed from the framework.

OSGi is a powerful tool for remote management of devices over local networks or the internet.

SMF

Services Management Framework (SMF) is IBM's OSGi Service Platform Release 2 compliant implementation of a management server.

SMF is optimized for embedded deployment. It has a componentized implementation to allow user specified configurations dictated by functionality requirements and resource constraints. It can be deployed on systems ranging from the base runtime requirements on WebSphere Micro Environment jclGwp configuration with a footprint of 815KB to a system using a feature rich J2SE configuration.

Resource Management is available to manage runtime constraints. For each bundle the developer can specify maximum memory allocation, the number and priority of threads, socket usage and bundle data area files. Resource management can be integrated into the SMF Runtime.

SMF Runtime

The SMF runtime provides the core functionality of SMF and components to allow integration with IBM WebSphere Studio Device Developer development environment to simplify bundle development, testing, and deployment. The runtime can be accessed through the SMF Bundle Developer, a Graphical User Interface deployed as an SMF bundle, or through the command line. These options allow remote access through SSH, http, and through the WebShpere Device Developer Environment.

SMF Bundle Server

The SMF Bundle Server can be used to manage the storage and deployment of bundles to a network of devices and can be shared by multiple users. Services can be requested by the specific devices runtime or can be initiated by a system administrator to push updates to several devices. The server can interact with a Management Agent to administer unlaunched bundles.

SMF Bundle Developer

SMF Bundle Developer is a plug-in for WebSphere Studio Device Developer. This Integrated Development Environment allows developers to develop, test and deploy OSGi bundles.

SMF Bundles

SMF provides a framework into which an application can be loaded and managed. The applications are encapsulated in a wrapper called a bundle. The bundle contains a Java Archive (JAR) file or a WebSphere Micro Environment executable (JXE) and a manifest file.

SMF Bundle developer is available as part of IBM WebSphere Studio Device Developer 5.5. The Eclipse plug-ins provide a development environment to simplify OSGi bundle creation and management. It provides an SMF Perspective from which an SMF Bundle Server and SMF Runtime can interact.

The Manifest editor exposes a simple graphical interface to edit OSGi headers in the manifest file. These headers are required by the OSGi specification and are used to install and integrate the bundle into SMF. The manifest file describes the bundle and the state of the bundle's dependencies. Once installed, any registered services the bundle provides can then be used by other bundles.

A Residential Gateway -- Other applications of the CerfBoard Gateway Server

There are many real world applications of Services Gateway Servers that employ Intrinsyc's CerfBoard technologies.

- **Washing Machine** -- The washing machines of a high-rise apartment building are all integrated with a CerfBoard 405EP. The computer reports usage information of each of the machines to a Gateway Server that can be accessed with any Java enabled browser.

  When a tenant wishes to do laundry they can access the washing machines site to see how many machines are free and how much time is left for the machines in use. The tenant can then reserve a machine to allow them time to get to the laundry room. When their load is done, the machine sends notification to the tenant's computer.

- **MP3 Streaming** -- A wireless enabled CerfBoard 405EP Gateway Server is attached to a home stereo through an audio expansion daughter board. The CerfBoard 405EP receives streaming MP3s from the user's PC and plays them through the stereo. The entire system is controlled through a PDA browser that accesses the SMF control bundle. From the comfort of your couch you can control your audio experience with minimal investment.

Where can I find a CerfCube 405EP?

For information on this and other applications see Intrinsyc's Cerfcube 405EP page.

About the author: Telly Cooper is currently working as a Field Application Engineer at Intrinsyc Software. He has a B.A.Sc. in Computer Engineering from The University of British Columbia in Vancouver, Canada.