



1855 Hamilton Ave. #203, San Jose, CA 95125
408-371-3060 (voice) 408-371-3061 (fax)
www.OctaveSW.com

Michael Grischy

Software/Firmware Consultant
Michael@OctaveSW.com
(408) 371-3060

SUMMARY

Experience: More than 25 years experience covering all aspects of software development: helping clients define project requirements, defining application architecture, writing specifications, designing, coding, and testing software and firmware, porting software to different platforms, reviewing digital hardware designs for firmware implementation feasibility, debugging prototype hardware, and writing internal and user documentation. I have also done work trouble-shooting complex software systems and enhancing existing products to improve performance, improve GUI usability, and reduce memory footprint size. I am comfortable working closely with hardware and with using such tools as network analyzers, in-circuit emulators, logic analyzers, and oscilloscopes. I am adept at quickly learning new technologies and applications. I also have experience providing expert consultant litigation support.

Technologies: Projects worked on cover a broad array of applications, technologies, tools, and platforms including: 802.11 wireless LAN (WiFi) networks and communications systems; embedded linux kernel drivers and applications; embedded real-time microprocessor systems and communications devices, with or without a real-time operating system (RTOS) such as VxWorks, AMX, or MultiTask!, running on processors such as IXP425 (ARM XScale), Power PC PPC405EP, 8051, 80186, Z8018x, and MIPS; web-based applications using Java Swing, Java Servlets, and Java RMI; network drivers and LAN and WAN protocols, such as TCP/IP and Apple's LLAP and AppleTalk protocols; Microsoft Windows technology, including GUI applications, DLLs, services, NDIS drivers, and other device drivers for most versions of Windows, including Windows CE and Windows Mobile, developed with tools such as the WDK, WinDbg, Visual C++, MFC, eVC++, and Platform Builder; serial communications; and PCB CAD applications.

Languages: C, C++, Java, Pascal, Fortran, Z80 and x86 assembler.

EXPERIENCE

(see [Consulting Projects Addendum](#) for more details)

**Consulting Engineer, Co-Founder
President (since 2005)**

2001 – 2010

Octave Software Group, Inc., San Jose, CA.
A software/firmware consulting firm.

A sampling of projects:

- Assisted in porting CCXv5 features from one code base to another for an NDIS 6.20 ExtSTA 802.11 WLAN Native Wifi miniport driver for Windows 7. ([details 33](#))
- Worked extensively with a suite of 802.11 wireless LAN (WiFi) control software composed of applications, services, and NDIS drivers running under Windows CE and Windows Mobile on a handheld device, including:
 - Architecting and implementing a solution to replace a proprietary 802.1x authentication supplicant with an open source based supplicant. ([details 32](#))
 - Re-architecting a key component of the suite to resolve chronic design problems, improve product reliability, and improve user responsiveness. ([details 31](#))
 - Specifying and implementing RETAIL build run-time debug logging to reduce time and effort needed to debug complex problems found during formal test. ([details 30](#))
 - Leading a team to re-merge parallel branches of the software suite to correct hundreds of problems introduced by a faulty merge methodology. ([details 26](#))
 - Leading the effort to draw up a formal high-level architecture document describing the software suite, as well as writing the bulk of the document. ([details 25](#))
 - Architecting and implementing support for EAP/FAST and PEAP/EAP-GTC 802.1x authentication protocols. ([details 23](#))
 - Re-designing the code that interfaces to the (old) 802.1x authentication supplicant to accommodate major changes that had been made to the supplicant's API. ([details 22](#))
 - General trouble shooting of various complex and subtle problems in the software suite, including finding bugs in industry-standard infrastructure devices. ([details 29](#))
- Modified the firmware in an 8051-based handheld device to accommodate the replacement of old CPU and LCD chips that were facing end-of-life. ([details 28](#))
- As an expert consultant, performed research and wrote reports on certain aspects of IBM OS/2 installation procedures and associated BACKUP/RESTORE utilities. ([details 27](#))
- Wrote glue logic and a GUI utility to allow a developer to configure the diagnostic interface of a CDMA radio in a Windows Mobile based SmartPhone product. ([details 24](#))
- Designed and wrote software to marry two linux products from different parties: a standard GSM cell phone and an 802.11 wireless VoIP phone. ([details 21](#))
- Helped client in extreme time crunch effort to integrate a third party's lightweight SSH Server into an existing embedded 802.11 wireless Access Point application. ([details 20](#))
- Designed and coded both a sockets-based kernel-space/user-space communications driver and a packet-switching kernel driver for a linux based 802.11 Access Point. ([details 19](#))
- As an expert consultant, assisted in court-mandated review of technical documentation in order to evaluate it for usability by third parties. ([details 18](#))
- Re-architected and re-wrote ethernet driver firmware for a Power PC-based multi-threaded real-time embedded system running in an 802.11 wireless RF Port. ([details 17](#))
- Wrote flash management API for a 186-based multi-threaded real-time embedded system. As follow-on work, diagnosed a bug in an FPGA-implemented UART. ([details 16](#))
- Implemented a sockets-based client/server control protocol for an embedded ATE application, running under WinCE, for testing 802.11 wireless radio cards. ([details 15](#))

Consulting Engineer

2001

GVO, Inc., Mountain View, CA.

A product development consulting firm.

- Reworked existing software to create diagnostics firmware for testing a new hardware

revision of an existing 80186-based 802.11 wireless LAN telephone. ([details 14](#))

- Partially implemented the SIP call-setup protocol for use in an 802.11 wireless LAN telephone. (The project was cancelled.) ([details 13](#))

Consulting Engineer

1992 - 2001

Probitas Corporation, Mountain View, CA.

A computer engineering consulting firm.

A sampling of projects:

- As an expert consultant, assisted with forensic work on several hard disk drives searching for evidence of illegal copies of source code. ([details 12](#))
- Architected, designed, and coded portions of a multi-tiered web-based data gathering system, based on Java Applets and a multi-threaded Java Servlet. ([details 11](#))
- As an expert consultant, assisted with the investigation of the possible infringement of patents covering advanced video display technology. ([details 10](#))
- Designed and coded a multi-threaded Java Swing dialog-box application that uses the Java Media Framework to play WAV files of foreign language phrases. ([details 9](#))
- Designed and coded a multi-threaded Java program that tests a set of Java Servlets that comprise part of a browser-based employee review application. ([details 8](#))
- Redesigned and enhanced the performance of embedded real-time firmware running on an 8051-based handheld device that reports the fluorescence of algae in water. ([details 7](#))
- Designed and coded networking firmware in printer-related products to utilize Probitas' AppleTalk network stack. Also implemented Apple's ADSP protocol. ([details 6](#))
- Designed and coded firmware to implement Apple's LLAP network communications protocol for use in a RISC-based printer. ([details 5](#))
- Ported Probitas' AppleTalk stack to run on a proprietary ARM/networking processor under the VxWorks RTOS. ([details 4](#))
- Helped customer set up and configure his VxWorks environment. Designed and coded a Windows program to update the product's flash image in the field. ([details 3](#))
- Designed and coded a Windows driver, layered over the standard serial driver, that permits user authentication transparently to standard serial comm apps. ([details 2](#))
- Designed and coded an object-oriented DLL in C++ that encapsulates a set of robotic wafer-fabrication tools and the protocol used to communicate with them. ([details 1](#))

Software Engineer, Software Manager

1986 - 1992

Teradyne Inc., Santa Clara, CA.

Case Technology Inc, Santa Clara, CA (bought by Teradyne).

Producers of CAE, CAD, and other software.

- **PCB Software Manager:** Managed group responsible for interfaces to OEMed PCB layout product, gate assignment and back-annotation software, netlist-to-schematic generation software, and software-porting to Japanese-language-based NEC-PC platform. Also was responsible for relationships with outside third parties developing netlist-to-schematic generation software and PCB router software.
- **PCB Software Architect:** Was a key member in team responsible for overall specification, architecture, design, and implementation of PCB layout software, including PCB router interfaces, auto-placement software, DRC software, and other supporting

software. Contributed in areas of employee training and internal software development standards.

Software Engineer, Software Manager

1982 - 1986

Vectron Graphic Systems, Inc., Santa Clara, CA.

A Producer of CAE and CAD software.

- ***PCB Router Software Manager:*** Managed team responsible for development of PCB router software. Projects included designing and implementing several PCB routers, a database sub-system for PCB drawings, a schematic back-annotator, and a netlist capture program. Contributed in areas of software control, software documentation, and coding standards.

PROFESSIONAL ACTIVITIES

Speaker at Embedded Systems Conference: Gave a tutorial class as part of the Embedded Training Program at the Embedded Systems Conference held in San Francisco in March 2005. The class, entitled *Design Patterns for Tasks in Real-Time Systems*, was designed to help developers new to embedded real-time systems learn how to break down an embedded application into a set of tasks.

PUBLICATIONS

“Using Design Patterns to Identify and Partition RTOS Tasks”, embedded.com (an on-line magazine associated with the Embedded Systems Conference), February 6, 2006 in three parts, co-authored with David E. Simon. The article was based on the paper written for the tutorial class given at the Embedded Systems Conference, 2005 (see above).

PROFESSIONAL ORGANIZATIONS

PATCA (Professional And Technical Consultants Association)

EDUCATION

BSEE, San Jose State University, 1984, summa cum laude.
Course work emphasized digital design.

CONSULTING PROJECTS ADDENDUM

This addendum gives additional details about selected consulting projects.

Octave Software Group, Inc.

2001 – 2010

- ***Port CCXv5 Features to Windows 7 WLAN miniport driver:*** Assisted client in porting those portions of an existing WiFi driver that implement version 5 of the Cisco Client eXtensions (CCXv5) from one code base to another with a different architecture. The driver is an NDIS 6.20 Extensible Station (ExtSTA) 802.11 WLAN Native Wifi miniport driver for Windows 7 (win7). Work included running CCX test cases manually and with an Azimuth test bed. The work was performed in C using Windows tools such as the WDK (Windows Driver Kit), WinDbg, and DebugView. ([summary 33](#))
- ***802.1x Authentication Supplicant Replacement:*** Architected solution for replacing a proprietary third party 802.1x authentication supplicant (from Meetinghouse) with an open source based third-party supplicant (from Devicescape). Lead small team in integrating the new supplicant into an existing suite of 802.11 WLAN control software. Work included writing a programmatic front end to encapsulate the communications-specific and data-translation-specific attributes of the supplicant's text-based command and control interface. Challenges included, in the face of inadequate documentation, analyzing the supplicant's source code and behavior to determine what exact commands to send to it to elicit the desired result. Work was performed in C using Platform Builder and targeting Windows CE 6.0 and Windows Mobile 6.1. ([summary 32](#))
- ***Re-architect Key State Machine:*** Advocated for, designed, and implemented the re-architecting of a key state machine in a key component of a suite of 802.11 WLAN control software. The re-architecting resolved chronic design problems - misuse of mutexes, improperly handled race conditions, proliferation of global variables, unserialized event handling, and other flaws - greatly increasing product reliability and greatly improving user-responsiveness. ([summary 31](#))
- ***Dynamic Debug Logging:*** Advocated for, specified, and implemented dynamically controllable run-time debug logging capability for key components of a suite of 802.11 WLAN control software. The design allows debug logging to be enabled in RETAIL builds, greatly reducing the time and effort needed to debug complex problems found both during product formal test and in the field. ([summary 30](#))
- ***Trouble Shooter for 802.11 WLAN Software Suite:*** Was instrumental as a general troubleshooter through several releases of client's suite of 802.11 WLAN control software. Tasks of interest included: diagnosing driver code-space corruption caused by the DMA controller, finding and fixing problems with freeing allocated memory twice, and extensive analysis of wireless packet captures to identify issues in the software suite's radio driver/firmware and, on occasion, in brand-name industry-standard wireless infrastructure devices (i.e., wireless switches and access points/ports). ([summary 29](#))
- ***Modify Firmware for New 8051 and LCD Chips:*** Modified existing firmware in an 8051 based handheld fluorometer to support new variants of the CPU chip and LCD controller chip as the old chips went end-of-life. Work included reviewing the new hardware design and

working with the client to correct errors, checking all 8051 function register values and correcting incompatibilities, and writing firmware to emulate the old CPU chip's built in I2C controller used for writing configuration values and log information to EEPROM. Work was performed in C using the Keil development tool chain. ([summary 28](#))

- **Expert Consultant for IBM OS/2 Installation Research:** Obtained vintage PC equipment, performed experiments, and wrote reports giving detailed analyses of various aspects of installing the IBM OS/2 operating system and of the operation of the accompanying BACKUP and RESTORE utilities. ([summary 27](#))
- **Corrective Code Re-merge:** Detected and informed client of problems introduced into client's code base due to code-merging activity that used a fundamentally flawed methodology. Created team plan and lead the effort to re-perform the code merge correctly. The merge encompassed two parallel branches of an entire suite of 802.11 WLAN control software (approx. 500K lines of code) and corrected hundreds of defects resulting in a stable code base on which new development could proceed. Also wrote code-merging best practices document adopted by client. ([summary 26](#))
- **Software Architecture Documentation:** Lead effort to draw up a formal high-level architecture document describing client's entire software suite for managing 802.11 WLAN communications on a handheld Mobile Unit (MU). Wrote 80% of the document. Presented my portion to the development team as part of internal training. ([summary 25](#))
- **SmartPhone Radio Diagnostics Enabler:** Wrote glue logic and utility code to configure the debug interface of a CDMA radio in a Windows Mobile based SmartPhone product, including a Windows GUI utility to allow a developer to enable the radio's various diagnostic modes. Also performed general trouble shooting of radio and telephony interface layers (RIL/TIL). Challenges included clarifying and reconciling different interface specifications from the different groups working on the product. The work was performed in C++, using eVC++ and Platform Builder. ([summary 24](#))
- **EAP-FAST and PEAP/EAP-GTC:** Acted as architect and technical lead working with client's software team to upgrade a suite of 802.11 wireless LAN control software to support the EAP-FAST and PEAP/EAP-GTC 802.1x authentication protocols. Work included writing requirements documents, technical specifications, formal test plans, schedule estimates, sales force training presentations, and user documentation. Challenges included creating a functional specification for EAP-FAST and its related PAC management requirements based on analysis of the operation of Cisco's ACS Server and Wireless Client. The work was performed in C and C++ using eVC++ and Platform Builder and targeted both Windows CE and Windows Mobile platforms. ([summary 23](#))
- **802.1x Authentication Supplicant Upgrade:** Redesigned and reworked a portion of client's existing 802.11 WLAN control software suite to interface to a newer version of a third-party's 802.1x authentication supplicant (from Meetinghouse). The third-party vendor changed the supplicant's architecture from one of requiring all information prior to authentication to one of requiring that its information be supplied asynchronously bit by bit, via a callback, as the authentication process proceeds. The client's code was in the form of an NDIS protocol driver running under Windows CE 5.0 and Windows Mobile 5.0. Challenges included diagnosing and avoiding flaws in the third-party's threading model, as well as diagnosing and correcting existing flaws in the client's code's use of mutexes. ([summary 22](#))

- ***Linux GSM/802.11 VoIP Phone:*** Worked with a client and a third party to “marry” their two products together into a proof-of-concept prototype: Designed and coded multi-threaded communications “glue” software, based on linux named pipes, that allows the two parties’ software applications to communicate. The project also required modifications to both parties’ software in order to interface to the “glue” software, including extensive work with one party’s SIP protocol stack. The end goal was to effect a seamless transition from a 802.11 wireless VoIP phone call to a standard GSM cell phone call automatically without user intervention. The prototype was demonstrated at the CeBIT trade show. ([summary 21](#))
- ***Time Crunch Embedded SSH Server:*** Helped client in extreme time crunch effort to integrate a third party's lightweight SSH Server into an existing embedded 802.11 Access Point application running on an MPC860 Power PC processor under the SuperTask RTOS and with the USNet TCP/IP stack. Work included writing a memory allocation sub-system to support malloc and free, designing and executing formal test plans to validate operation, and writing user documentation. The firmware was delivered on-time. Challenges included finding and fixing bugs in the TCP/IP stack. ([summary 20](#))
- ***Embedded Linux Kernel Drivers:*** Designed and coded a sockets-based kernel-space/user-space communications driver and a network packet-switching driver, both for an embedded linux system running on an Intel IXP425 processor. The packet-switching driver switches ethernet packets between the local subnet – which includes a LAN port and two 802.11 radios – and either the linux networking stack, or proprietary software running in user-space. The packet-switching driver uses the communications driver to exchange packets with the user-space application software. The kernel drivers are used in an Access Point device for 802.11 wireless networks. Development was done under linux using the MonteVista toolset. Deliverables included test scaffolding code with an extensive set of regression test cases as well as documentation on the packet-switching driver’s design and implementation for use by the engineers that inherited the firmware. ([summary 19](#))
- ***Expert Consultant for Documentation Review:*** Assisted in court-mandated review of technical documentation produced by a major software corporation in order to evaluate it for usability by third parties. Wrote an expert report on both the usability and completeness of the documentation and on the corporation’s proprietary software system used for viewing and navigating through the documentation. ([summary 18](#))
- ***Power PC Ethernet Driver:*** Re-architected and re-wrote ethernet driver firmware used in an RF Port device for 802.11 wireless networks. The firmware runs on an IBM PPC405EP Power-PC processor as part of a multi-threaded real-time embedded application running under the MultiTask! RTOS. Development was performed using GNU C++ tools on Sun Solaris. Work included configuring and controlling the processor’s EMAC (Ethernet Media Access Controller), using the processor’s MAL (Memory Access Layer) to transfer packets between the EMAC and memory, and controlling an external PHY chip via an 802.3u standard MII (Media Independent Interface) implemented in the EMAC. To simplify any future porting effort, the object-oriented driver design is partitioned into hardware-dependent and hardware-independent objects, with an interface between the two parts that is enforced using an abstract base class. Work also included testing link auto-negotiation and driver throughput with SmartBits test equipment. Deliverables included training material for the engineers that inherited the firmware. Challenges included: writing work-around code to detect whether the PHY chip had “crashed” due to electrostatic discharge and to reset it if it

had; and controlling the PHY chip in such a way that its link state would survive across an intentional reset of the processor. ([summary 17](#))

- **802.11 WLAN Radio Configuration Application:** Helped in a client's time-crunch bug-fixing effort for an 802.11 Wireless LAN mobile unit configuration and control program, developed using VC++ and MFC, that runs on Windows 98, ME, 2K, XP and on Windows CE. Tasks included adding multi-threaded capability for IP address acquisition, finding and fixing subtle memory leaks and memory corruption bugs, and making enhancements and bug fixes to the graphic user interface.
- **Trouble-Shooter for 802.11 WLAN Software Suite (previous version):** Was instrumental as a general trouble-shooter through two release cycles of a suite of software that runs on standard and proprietary hardware platforms running Windows XP, 2K, CE 4.1, CE 4.2, PocketPC 2002, and PocketPC 2003. The software suite is used for configuring and controlling an 802.11 wireless network stack running on a mobile unit and is comprised of a complex scheme of layered software that includes radio card firmware, an NDIS driver, network stack drivers, middleware, and a GUI application. Tasks included: debugging memory leaks, MFC bugs, and networking bugs all found in Microsoft's CE operating system; debugging system timer and performance problems in a third party's custom CE implementation; debugging subtle problems in the interaction between the mobile unit and various 802.11 Access Points and RF Ports; programmatically interfacing to Microsoft's Wireless Zero Config functionality in order to use Microsoft supplicants for various 802.1x authentication schemes, such as EAP/TLS, PEAP, WPA-PSK, and WPA-PEAP; and specifying and implementing test utilities for use in automating the testing effort.
- **Flash Management API and UART Debugging for Embedded System:** Wrote a "flash management" task that runs under the KADAC AMX RTOS in an AM186ER-based multi-threaded real-time embedded system for a proprietary biomedical device. The task serializes accesses to system flash by other tasks so as to protect the flash during write and erase operations. The task also provides a generalized API that maps independently-erasable sectors in the flash to "pages" that can be grouped together and used for a single purpose by higher layer code. Also wrote higher layer code that uses the flash API to store and manage calibration parameters and system log data. Follow-on work included troubleshooting an intermittent RS-232 communications failure between a host PC and the device. Diagnosed a bug in the device's UART – implemented in FPGA logic – which would occasionally miss the first bit of the first character transmitted to it after the RX line had been idle. Devised a workaround that required no changes to the device or its firmware, thereby avoiding FDA re-certification. ([summary 16](#))
- **802.11 Radio Test System:** Worked on many aspects of an automated test equipment (ATE) system used for testing 802.11 wireless LAN radio cards. The system is composed of a "Host" running Windows 2000 Server and a PC104-based radio test "Fixture" running Windows CE. Initially implemented a sockets-based client/server protocol over TCP/IP in an embedded multi-threaded real-time VxWorks environment for Host-Fixture communications. Follow-on work included porting the Fixture code to Windows CE, configuring the Windows CE kernel to support additional features such as multiple ethernet cards, reworking the Host and Fixture code architecture using object oriented design techniques so that multiple radio types could be supported with a minimum of code changes, improving the performance of the Host-Fixture communications by a factor of 10 or more, adding support for interfacing to a bar code scanner and a controllable power supply, simplifying the system

startup process by using industry-standard protocols such as DHCP, FTP, and DNS, and reworking existing internal documentation to accurately reflect system design, setup, and operation. ([summary 15](#))

GVO, Inc.

2001

- **802.11 Wireless Phone Diagnostics:** Combined and reworked existing software to create diagnostics firmware for testing a new hardware revision of an existing 802.11 wireless LAN telephone. The firmware runs on a 186-based processor without a real-time kernel. The program presents a menu-driven user interface, via the phone's keypad and LCD display, through which the user can invoke commands to exercise various portions of the hardware. Modified various device drivers to conform to the new hardware design. ([summary 14](#))
- **SIP Protocol for 802.11 Wireless Phones:** Implemented the SIP call-setup protocol for use in an 802.11 wireless LAN telephone system integrated with a local PBX. The project entailed modifying existing telephone firmware by removing the existing H.323 protocol code and replacing it with SIP code. Challenges included interpreting the sketchy SIP RFC and inventing a SIP URL scheme that would be compatible with a SIP ethernet-LAN-to-PBX Gateway supplied by a third vendor. (This project was left unfinished because customer demand for the product declined.) ([summary 13](#))

Probitas Corporation

1992 - 2001

- **Expert Consultant for Infringement of Intellectual Property:** Assisted with an investigation into the possible infringement of a client's intellectual property rights by a competitor. Carried out forensic work on several of the competitor's hard disk drives searching for evidence of illegal copies of source code belonging to the client. ([summary 12](#))
- **Web-Based Configuration for Call Distribution Center:** Architected, designed, and coded the intermediate and front-end portions of a multi-tiered web-based system for configuring, monitoring, and controlling a remote Call Distribution Center data-gathering application. The system provides a browser-based user interface that allows multiple users to simultaneously access the data-gathering application via an intermediate web server. The web server serves up HTML and Java applets, and runs a multi-threaded servlet that provides an API to the data-gathering application running on a remote machine. The browser-based UI is comprised of a suite of tightly coupled applets, running in multiple frames, that communicate with each other and with the servlet. The remote data-gathering application provides an RMI-based command interface accessed by the servlet. ([summary 11](#))
- **Expert Consultant for Video Patent Infringement:** Assisted with the investigation of the possible infringement of patents covering advanced video display technology. Work included evaluating the patent claims, reviewing schematics and other technical documentation, and reviewing depositions. ([summary 10](#))
- **Java Foreign Language Phrase Player:** Designed and coded a multi-threaded Java Swing application that uses the Java Media Framework to play WAV files. The program, designed to play a series of foreign language phrases, presents a dialog-box-like user interface that allows the user to control various aspects of how the phrases are played. ([summary 9](#))

- ***Servlet Test Application:*** Designed and coded a Java program that tests a set of Java Servlets that provide the user interface for a browser-based employee review application. The test program, written in Java, provides the capability to simulate multiple simultaneous users through the use of multiple threads. ([summary 8](#))
- ***Embedded Fluorometer:*** Redesigned and coded an existing embedded real-time program that periodically samples the fluorescence of algae in water and then calculates and reports algae concentration values. The firmware, written in C using the Keil IDE, runs on an 8051 processor without a real-time kernel in a power-conscious environment. My work included enhancing the firmware's performance, allowing it to meet its target design criteria. ([summary 7](#))
- ***AppleTalk Network Protocols:*** Worked extensively with Probitas' AppleTalk network protocol stack on many embedded multi-threaded real-time projects. These projects entailed designing and implementing firmware for various printer-related z8018x-based AppleTalk network communications products, including one in which I designed and coded firmware to implement Apple's ADSP network communications protocol. All of these projects included helping the customer debug prototype hardware. ([summary 6](#))
- ***LLAP Network Protocol for RISC-Based Printer:*** Designed and coded firmware to implement Apple's LLAP network communications protocol for use in a RISC-based printer. LLAP is the lowest level in Apple's LocalTalk networking protocol specification and is the data-link access protocol used to control access to the physical network wire. Thoroughly tested the implementation on the hardware to ensure that the timing requirements of the protocol were met. Challenges included analyzing and mastering the subtle behaviors of the Zilog 8530 serial communications controller (SCC) which communicates with other network nodes using the SDLC protocol over an RS-422 high-speed serial connection. ([summary 5](#))
- ***AppleTalk Port to ARM/VxWorks:*** Ported Probitas' AppleTalk stack to run on a proprietary ARM/networking processor using the VxWorks RTOS and the Tornado tools. Interfaced the bottom of the stack to VxWorks' MUX layer. Confirmed operation on the customer's hardware. ([summary 4](#))
- ***LocalTalk Packet Transfer Protocol:*** Designed and coded firmware that receives and transmits LocalTalk network frames, using Probitas' LLAP driver, and transfers the frames to and from a separate host system via a PCI bridge. The firmware runs on a Z80185 processor and uses the MultiTask! RTOS. Designed the protocol used to exchange commands and data across the PCI bridge.
- ***VxWorks-Based Field Update Firmware:*** Helped a customer to set up and configure his VxWorks environment for developing a custom x-ray imaging system. This included finding and reporting bugs in a BSP written by a third party. Also designed and coded a Windows program that allows a field technician to download either configuration data or a firmware update and write it into the system's flash. Designed the serial communications protocol used for downloading. Designed the Windows program's user interface for ease of use and fool-proof operation. ([summary 3](#))
- ***Windows Device Drivers for User Authentication:*** Designed and coded three versions of a custom device driver – one each for Windows 95, NT, and CE – that layers itself on top of

the standard Windows Serial Port Driver. The purpose of the driver is to monitor, capture, and insert data into the RS-232 serial data stream flowing between a remote host and a local standard communications application in a way transparent to the standard communications application. The driver is controlled by a custom monitoring application running on the local machine whose purpose is to perform user authentication with the remote host. Challenges included thoroughly analyzing the rich feature set and capabilities of the Windows Serial Port Driver so that the custom driver could precisely mimic it in order to remain transparent to the standard communications application. As part of this project, also ported to Windows CE several MFC Visual C++ Windows applications, including the monitoring application already mentioned. ([summary 2](#))

- ***Object-Oriented Robotic Control System:*** Designed and coded a Windows DLL that was used by application software to communicate with and control a suite of robotic wafer-fabrication tools. The DLL, written in C++, encapsulates the tools and the protocol used to communicate with them. ([summary 1](#))
- ***DOS Radio-Spot Capture:*** Redesigned and coded an existing DOS-based “radio-spot” capture program written in Pascal that captures data from an audio card and writes it to disk for later distribution to radio stations. Reworked the program’s GUI to improve its usability, upgraded it to create .WAV files, and fixed Y2K problems.
- ***Serial Communications Monitor:*** Designed and coded various parts of an embedded application that monitors and controls serial communication between hosts and modems through a bank of 8530 USARTs. The application incorporates a real-time kernel running on an 80186 processor. Wrote boot-up and initialization code for the 80186 and for the 8530s. Brought firmware up on prototype hardware and completed debugging and testing.
- ***Network Protocol Interpreters:*** Designed and coded protocol interpreters for EGP (Exterior Gateway Protocol routing protocol), IS-IS (Intermediate System to Intermediate System routing exchange protocol), NFS (Sun’s Network File System), RPCBind (Sun’s Remote Procedure Call Bind protocol), and others. Upgraded an existing protocol interpreter suite for Banyan VINES (Virtual Integrated Network Services) to interpret VINES version 5.5. These interpreters all run within the framework of an existing DOS-based network analyzer tool, translating data captured from a local area network into human-readable form.