Resume for:

Mark R. Hahn
212 Sawdust Ave
Cottekill, NY 12419

email: mhahn@hvc.rr.com
Phone: 845-687-9945

Profile:

I am a real-time embedded systems contractor with a proven ability to deliver high quality hardware and firmware under tight deadlines. I have an Electrical Engineering background that gives me a unique perspective on HW/SW interface issues. I am experienced with embedded systems ranging from 8 bit processor based ASICs (Application Specific Integrated Circuit) to designs involving multiple 32-bit processors.

Technical skills:

Languages - C++, C, Visual Basic, Python, PHP, JavaScript, Perl, Pascal, PL/I, various assembly languages (Atmel AVR, Microchip PIC, PowerPC, 68hc08, 68hc05, 8051, ColdFire, 68000, 65C02)

Operating Systems - Microsoft Windows, MS-DOS, Unix, VxWorks

SW Tools - Visual C++, Visual Basic, Microchip MPLAB, WinAVR, Microchip C18, CCS PICC, HiTech C, SubVersion, CVS, CodeWarrior, Tornado II, Make, PVCS, Cognex Vision Systems, Eagle Layout Editor

HW Tools - oscilloscopes, logic analyzers, in circuit emulators, device programmers, motion control systems, vision systems, protocol analyzers, and soldering iron.

Education:

BS Electrical Engineering, Columbia University

From 1996 till now I’ve worked as an embedded systems contractor. Here are some projects I've worked on:

Industrial Boiler Controller -20011 to present

I’m currently contracting for a company that makes industrial on demand boilers and water heaters. I’m involved in a few different projects.

I’ve developed a new application for the existing dsPIC30F based used in water heater products. The application allows up to 8 water heaters to be networked together into a system. The system responds dynamically to changes in demand. As demand goes up the number of heaters firing increases and as demand goes down the number of heaters firing decreases. This project involved learning and modifying the existing code base of approximately 25,000 lines of C code.

I’m also working on developing firmware for a next generation boiler controller. I’ve been involved in the design of the hardware and have ported some of the existing firmware so that it can be used on the new hardware.

Fuel Cell Controller -2004 to 2010

I helped a small Colorado company develop a controller used to run small stand-alone fuel cell systems. I was originally hired to consult on some of the hardware design for the system and went on to write most of the firmware.

The original controller was based on the Microchip PIC18F8720 microcontroller. The controller firmware is written mostly in C, with a small amount written in assembler. When finished the firmware was about ten thousand lines of code

The latest version of the controller was redesigned to use an Atmel ATmega2560. It does pretty much the same thing as the older version, but with several hardware changes. I ported the code to the new processor, and modified it to support the new hardware.

Video Camera Protocol Converter-2009

I developed a small interface board that accepts VISCA protocol commands over a RS232 serial interface and converts them to the commands used by an OEM camera module.

Using a prototype developed for a previous project, I was able to develop and test the firmware for this project and deliver a working prototype system to my client’s customer in 2 weeks.

Video Camera User Interface-2008

I developed an interface board and firmware for it that accepts commands to control a video camera over a RS485 serial interface and also accepts user input from a built in keypad. The RS485 interface accepts Pelco protocol commands. The keypad interface let the user modify camera settings as well as change the camera’s zoom and focus.

Network Switch Statistics System – 1999 to 2001

Network switches are used in telephone networks, typically as an interface to the rest of the network. The switch I worked on used an ATM based architecture that allowed the user to mix voice and data within an existing network. It was a high reliability / high availability, message based, multi-processor system running the VxWorks real time operating system.

I was the lead programmer of a group of three programmers that implemented the statistics subsystem. We designed, wrote, debugged, and tested over one hundred thousand lines of code in less than a year. The statistics subsystem code was written in C. Some of the subsystem configuration code was written in XML. The diagnostic and test code was written in C and JavaScript.

Pacemaker Firmware Upgrade – 1998-1999

I consulted for a year for a company that designs and produces pacemakers and other high reliability medical electronic devices.

The bulk of the work I did was on an older generation of pacemaker. It had very limited code space and very little RAM. I compressed the existing code and added new features. This involved learning a bit about a number of different subjects; cardiac physiology, pacemaker hardware, the client's build process and test tools, and FDA approval requirements for electronic medical devices.

Pacemaker Test System - 1998

While consulting for the pacemaker manufacturer, I also developed a system used to test pacemaker firmware. I researched and designed a method of collecting high-speed data from a pacemaker prototype. The data was saved on a PC hard drive and dumped for later examination by a software engineer. This project involved finding an off the shelf high-speed IO card for use in a PC, designing a small interface board to connect the pacemaker to the IO card, doing the PCB layout for the card, and writing a couple of PC applications to capture, and interpret the recorded data.

FM Subcarrier Data Receiver - 1998

The pager watch company brought me back a year later to develop a packet based data network receiver chip and a prototype data receiver. I helped design a hardware interface between the receiver chip and a Microchip PIC microcontroller. I wrote code for the PIC, in C and assembly.

Hard Disk Read/Write Head Assembly Robot - 1998

Until 1999 all read/write heads for hard disks were assembled by hand. I consulted for a robotics firm that developed one of the first commercial robotic systems used to assemble read/write heads.

I was part of a small team that designed and built the robot in less than 3 months. I worked mainly on the software, though I did design some of the circuitry that went into the final hardware. My hardware background came in handy when we had to debug or repair the robot.

Pager Watch - 1996 - 1997

An American subsidiary, of a Japanese watch manufacturer, designed pager watches and pager systems for the parent company. The watches incorporated a text message pager in a wristwatch.

I was brought in to work on a second generation of the pager watch. This was an ambitious system on a chip (SOC) project. The delivered firmware had to be as close to bug free as possible, since there was no way to upgrade it. I worked as part of a 4-person team that developed the firmware for the watch. I designed and implemented a display driver, sound driver, and a message assembly, storage, and retrieval system. All the firmware was written in 65C02 assembly language.