SCOTT R. BOWDEN **(603) 635-8635**

42 Balcom Road, Pelham, NH 03076 **email: s@sbowden.org**

SUMMARY:

Seeking to apply my Software Engineering skills and experience in the development of software systems. I have worked extensively with C and C++ on a variety of platforms both conventional and embedded.

Specific packages that I have worked with recently include Visual Studio (numerous versions), WinIdea, Tasking compiler, Matlab, svn, LabVIEW, MPLAB, Crestron, SourceSafe, Windows NT DDK.

EXPERIENCE:

Veoneer, Lowell, MA May 2015 - October 2020

Embedded Software Engineer

In early development of the v1.2 sensors (77GHz), reviewed available DSPs for the new 77GHz radar technology; contributed to the development of functional prototypes; on my initiative organized software training for the future v1.2 development teams on competing technologies; reviewed and examined available in-circuit debugging systems. These were key activities in providing the Veoneer (then Autoliv) vendor negotiating team with multiple technical solutions that relied on multiple vendors - and ultimately allowed Veoneer to transition from Texas Instrument DSP parts to those from Infineon.

As part of the DSP evaluation process, it was necessary to specify how we were going to use that chip to process the RADAR signal. I prepared a document describing this process as an algorithm - complete with explanations as to how and what each step accomplished. This became an immediate hit - as it bridged the gap between the signals group and everyone else.

The “Boot Loader” was a completely independent but relatively simple system that allowed for the reprogramming and reconfiguration of the sensor. It was also the first place where many of the DSP features where implemented - providing a baseline for development in the core RADAR system. I was a frequent contributor to the debugger, especially in the initial development and later maintenance of several interfaces - including SPI, CAN, LVDS, and straight DI/DO. As the first user of new circuitry, I commonly acted in direct support of HW development and testing. This was especially true for interfaces to the ST Micro MMIC, a Xilinx FPGA, and LVDS adapter firmware.

Although all of the sensor code was done in C under MISRA rules, there was a continuous requirement for C++ and Matlab for simulations and unit testing and for C++ and Python to provide incidental tools - such as reporting per-module stack usage, data conversions, signal simulators, etc.

Contributed extensively to the sensor blockage monitoring system. The key real-time issue to resolve is whether something is blocking the sensor or we in a RADAR “desert”? This feature required extensive real-world data sets showing sensor responses to RADAR deserts and actual blockage situations. Aside from direct design and code implementation, I developed a server set where this data could be reduced and applied to prototype blockage

algorithms - and directed drive teams in collecting the sensor data while capturing associated environmental parameters (weather, road type, sensor icing, etc.).

Performed a system-wide review to accommodate the Functional Safety requirements related to the setting of every Aurix (DSP) register - roughly 2800 of them.

Provided general development support for the core systems. We were developing the RADAR sensor at the same time that Infineon was developing there Aurix series DSPs. These platform changes, problems uncovered during manufacturing, and special customer requirements provided a continuous flow of changes that were required as v1.2 developed. As a member of the core development group, many of these became my assignments. Three examples follow.

- Toyota wanted our DSP to control their heating element for deicing. I provided experimental code to support this along with documentation on requirements for the heating element and guidance on the selection of a driving chip.
- The CAN transceiver is the interface between the sensor and the rest of the vehicle. Some vendors required special energy-saving modes that required more advanced CAN interface chips. I provided the code for determining the how the sensor was configured and driving these variant chips.
- Provide MS Win32 simulation code for the multitasking support provided by the sensor multi-tasking services. This was based on “WaitForMultipleObjects”.

Henschel, an L-3 Company, Ayer, MA March 2005 – March 2015
Senior Software Engineer

While assigned to the Software Develop Group, I developed, tested, generated full deliverable documentation, and supported final software audits for several Navy software components for life-safety or mission-critical systems. I am often comfortable at the hardware level and have, at times, designed or modified simple circuitry with full manufacturing documentation.

My worked primarily involved developing and delivering software components for systems and devices used in outfitting Navy ships and submarines. These include helm and lee helm consoles, navigation systems, local telephone systems, announcing systems, alarms systems, and battery monitoring and reporting systems. My contributions included developing software components for embedded processors and microprocessors, developing unit test procedures and tools, developing full build procedures (source code extraction, verification, and metrics, build development environment, build components, build target disk image, ghosting, software manufacturing drawing), developing system test procedures, first article testing, audit documentation, direct audit (PCA) support, and developing manufacturing test procedures and tools. I provided tools and procedures to support environmental testing (heat, humidity, etc), shock testing, and EMI testing of ship-bound systems.

Some of my specific software development contributions include:

- For a Navy submarine battery monitoring and reporting system: Software for two microprocessor (both PIC parts) for collecting information from each cell in the battery cell arrays, a Windows 2K device driver to provide an interface to that collection

system, a queued reporting system that controlled processor loading while saving, retrieving and reporting battery cell data.

- For a navigation console: provided GUI support in an MFC environment for graphically reporting assorted navigation information (Rudder Order, Rudder Answer, Deck/runway wind/crosswind components, etc).
- For a RADAR power and array controller: Test tools under LabVIEW (for the Siemens controllers) and 'C' and assemble (for the VMEbus-based processors).
- For a VoIP research project: Numerous several demonstration VoIP "stacks" and end-user devices, mostly with 'C' or C++ under Linux or proprietary versions of UNIX.
- For a British Naval announcing system: device controller and tools for an interruptible power supply (a UNIX variant).
- For several US and foreign Navy ships, developed full software for announcing systems including manufacturing and field installation procedures, user documentation, and software design flow documentation. In the case of the Australian system, this system (and software) was required to meet SOLAS requirements.
- For CVN78 (the Ford class aircraft carrier) wrote the backup steering component to the steering system (a microprocessor) which included RS232, RS422, and I^2C interfaces, configured the XP Embedded and implemented the IA requirements for the main steering software components and developed all software upgrades required for integration to the ship systems (Windows XPe).
- Supported the overall integrations of the steering system with all other ship systems with changes to SNMP, Profibus, and other interfaces. This program operated under 12207 rules.

I was 4 and 0 for successfully completing PCAs (Physical Configuration Audits) for Electric Boat - by far the best record at Henschel.

Avid Technology, Tewksbury, MA Feb. 2001 – Feb. 2004
Principal Software Engineer

Worked as a Principal Software Engineer in the hardware diagnostics group first in the maintenance of the diagnostics for their Meridian product and in the development of the Adrenaline and Nitris product line. These are high-end products used in the television and movie industries for non-linear video editing. My major challenge during the two-year development crunch for Nitris diagnostics was keeping myself off the critical product development and manufacturing paths while being outnumbered roughly 20-to-1 by the hardware engineers - which I was 98% successful in doing. My designs and code are not only modular and object-oriented but, in the terms of database designers, well-normalized. The type of hardware elements I have coded for include flash memories, other programmable devices, the I^2C bus, and JTAG.

- Developed a core library, in the form of a DLL, to provide common access to the Avid hardware for all diagnostic applications.
- Developed a test engine UI which supports a scripts which language elements specific to the Avid hardware.

- Developed numerous hardware tests for testing both the design and manufacture of the Avid hardware. For modularity, these tests were implemented so that once linked into the test DLL, they would appear on the user interface without further integration.
- Applications for maintaining PCI VPD structures, reprogramming FPGAs and such, MPEG key management, and assorted data conversion tools.
- Designed and participated in the development of several automatic hardware register interface tools. This includes include files (*.h) that are automatically scanned to create a register library that is then used in support of the script language and a register tree debugging tool.

PerkinElmer, Inc. Detection Systems Division, Woburn, MA May 1999 – Feb. 2001
(This division was sold to L-3 Communications in 2002.)
Principal Software Engineer

Participated in the development of airport luggage x-ray systems. My contributions there included:

- Completing the design and implementation of a specialized object-oriented multitasking event management system.
- Tracking down and repairing a multitude of FDDI network messaging problems. This included restructuring and recoding much of the low level application code to use the Winsock 2.0 "WSA" functions in place of the mis-implemented Berkley set.
- Surveyed a handful of multitasking applications for resource sharing conflicts - specifically, potential deadly-embrace conditions. I established a general resource-claiming protocol and specific resource rules for each application. This work eliminated many mysterious application "hangs" and established quick methods for finding deadly-embrace vulnerabilities.
- Maintained the Management Information System network bridge - a real-time GUI-based system for monitoring the x-ray systems and operators performance.
- Replaced the FDDI-based network with a 100Base Ethernet system. This included work with SNMP, the Cisco 2900XL MIB, the Cabletron ELS-100 MIB, and AG Group's Etherpeek.

IMAGRAPH Corporation, Chelmsford, MA March 1994 – March 1999
(their assets were purchased by Foresight Imaging in May 1999.)
Principal Software Engineer

At Imagraph, my primary contribution was the development of the drivers and SDK's for their HI*DEF and I-Series lines of video frame grabber boards. The primary development environments for their most recent SDK releases had been Windows 95/98/NT using MFC. I was the lead developer of these SDKs. Among my major contributions have been:

- The design and development of "Auto-SYNC" which uses the frame grabber board to survey all the video signals at each connections to the board and to measure each signal that it finds. The software automatically recognizes sync patterns and configures

the board to capture the video. This product is recognized as a tour-d-force in automated signal processing and remains a unique capability of Foresight prducts.

- The design and implementation of a field support strategy for use in connection with Auto-SYNC. The HI*DEF boards are most commonly used to capture from medical imaging devices such as MRI's. Our customers would install their systems at these sites hoping to capture perfect images before days end - or in time to make their next flight. Reports generated by Auto-SYNC could be faxed or e-mailed to IMAGRAPH where problems such as bad cabling could be rapidly diagnosed and a remedy determined. The combination of Auto-SYNC and this support strategy allowed most customers to make high-quality captures within two hours of the time they opened the box. In contrast, competing products commonly took weeks to set up properly and required specially trained personnel on site.
- The design and development of library and SDKs which convert a common video description (so-called "CHP files") to a HI*DEF board configuration. This completely isolates the application programmer from arcane board characteristics. It also allows any HI*DEF or I-Series board model to work with a CHP file generated from any other HI*DEF model introduced during my employment there.
- The redesign and development of manufacturing QC software tools, procedures, and pass/fail criteria.
- Assisting in the design and specification of the frame grabber hardware and testing of the prototypes.
- During that time I also contributed to their ICE-YC project (for JPEG compression of motion video) and the IMASCAN project (combination video capture and display cards).

My development experience before 1994 includes work in these application areas:

- Image processing.
- Signal processing.
- Process control.
- CAD/CAM systems.
- A word processing system for a technical periodical.
- Design of two cartographic data bases.
- Database normalization of the Air Force procurement process.

EDUCATION:

B.S. Computer Science, GPS 3.84
Franklin Pierce College, 1994