Brian Bennett

Engineer

Software design for embedded firmware and hardware testing. Development of automated test tools to improve repeat ability and to reduce operator error. Excellent communication skills, able to guide teams through development processes with focus on customer experience. Strict adherence to time critical schedules with exceptional depth of testing and development. Experience interfacing with internal and external customers to identify and address defects or improvements to implement economical solutions.

TECHNICAL SKILLS

C, C++
C#, .NET
Assembly
Python, Perl
MicroChip PIC
Atmel ATMega
ARM Cortex M4
FreeRTOS
Ruby
Linux / Bash scripting
CADKey, VersaCad, Draftsight
PostgreSQL database
Access databases
Xen VM server

EMPLOYMENT

SW Engineer TriLynx Systems, Fort Collins, CO *2015-2017*

- Wrote programs for field technician support, embedded field data collection units.

Wrote Java script to parse raw data coming in from the field.

Wrote Linux system scripts to monitor PostgreSQL connections and to add table partitions when required.

Legacy C code maintenance. Utilized VMs for code building and testing.

Developed Atmel AVR C code for embedded field data collection units

Updated client website Perl scripts to work with NOAA SFTP requirements.

Wrote numerous log file parsing scripts in Bash and Ruby to facilitate data recovery and maintenance.

SW Test Engineer (Contractor), Intel, Longmont, CO - *2014*

Wrote test scripts in Python for testing software using PyWBEM and CIM technologies.

Worked in an Agile environment.

Sr. Test Engineer (Contractor), Pico Digital, San Diego, CA - *2013-2014*

Production factory testing of new headend audio products, audio encapsulator and audio encoder.

Developed the test framework in C# for testing hardware interfaces to ensure compliance with product specifications. Updated C# test framework to simplify integration of test suite in to the framework UI.

Developed a networked control box for powering the test stand and for automated data input.

Wrote Ruby scripts for initial factory operating system loads.

Captured transport stream for analysis using a DekTec DTA-145 ASI/SDI IO adapter.

Automated hardware production testing on audio platforms – **reduced cost, tech interaction, test time and improved test quality and reliability**

Test Engineer, Pico Digital, San Diego, CA - *2011-2012*

- Developed and built an audio encoder test fixture for legacy production and RMA units.

Wrote control and data gathering in Ruby. Equipment integrated in the system included an audio encapsulator, IF modulator, L-band upconverter, satellite receiver, Audio Precision audio tester, and a GPIB-ETHERNET module for Audio Precision control.

Updated legacy satellite receiver production test system. Upgraded system from a hardwired IF modulator to a programmable IF modulator. Updated the C++ control software, fixing bugs, streamlining loops, and adding controls for the IF modulator and noise generator.

Developed and built the test fixture for the watermark receiver. Working with the design engineers to ensure the system tested the product to design specifications. Developed the initial release of the C# test framework.

Developed a C# class for testing audio with a digital Fourier Transform. Used a Rhode-Schwartz signal generator for the required AM/FM modulated RF input to the watermark receiver.

Developed test software framework – **simplifying changes to test software and reducing costs**

Worked design engineers to ensure production units were tested to system specifications

Applications Engineer, Entropic Communications, San Diego, CA - *2008 - 2011*

- Specified the API for the silicon tuner product. Integrated the API into the evaluation software kit. Evaluation software was written in C#. Maintained tuner documentation which included datasheets and product bulletins. Direct customer contact to debug integration problems.

Moved into the Multimedia over Cable Access, MoCA, and applications group. Involved with product integration certification. Leading summer interns built and developed two (2) certification test stations including control and logging software.

- Worked with design engineering to specify and develop tuner API

Wrote evaluation software
Working with management, led a small team to build certification test stations

Sr. Applications Engineer, Vativ Technologies, San Diego, CA - *2003 - 2008*

- Wrote and maintained documentation for the T3/E3 LIU product including datasheets and product bulletins. Worked with the design engineers in defining the default power up settings for the LIU. Worked with the digital designer in testing the eval board. Worked with customers answering technical questions pertaining to implementation. Worked at an offsite lab to do product verification, ensuring the LIU met or exceeded industry specification.

Help develop the 10Gbe MCM platform. Embedded and programmed a PIC micro controller used in the final product. Wrote the control loop to handle firmware calls to the 10Gbe device and MDIO registers. Also included methods of modifying core code via MDIO thus eliminating the need to use a JTAG or other device for code mods.
Developed the USB interface to the HDMI evaluation system. As part of the development, wrote the control loop used to program and modify device registers. Worked with design engineers to test the HDMI receiver. Maintained eval Ruby scripts adding new functionality as required.

Working with management and engineering, specified and implemented a documentation control system to centralize and maintain all the company's documents. This was used a supplement to the normal SVN source code repository. Released compiled or script eval code maintained in document control to simplify marketing and sales access.
Product evaluation and verification
Embedded system control software
Document Control

EDUCATION

Bachelor of Science, Mechanical Engineering, University of Missouri, Columbia, MO, 1984

TECHNOLOGIES:

Java - Developed a script to parse raw Alert2 data into a spreadsheet. This was then passed along to the engineer in charge for further processing APDUID sequences.

C# - Created a small application to be used in the field by technicians to maintain and update field data collection units. Previous to this, the only method to maintain the field units was to use a serial terminal and manually input the changes. With FieldTech, the technician is presented with a GUI with windows for viewing the data coming in from the DCU and one to edit scripts to send over to the DCU. There is also offline modes to allow for building scripts back at the office before heading out into the field.

Xen - Converted several idle machines into Xen virtual machine servers. This allows the developers to spin up virtual machines as needed for software development, testing and building. Scripts were written to create the images required by the developers. Scripts were also written to provision the images with a base set of Linux packages.

C/C++ - Developed the code for an Atmel embedded system used in a field data collection unit (DCU). The system was fully interrupt driven. Inputs triggered routines that were used to wake up the main processor and establish communications and measurements. As part of the testing process, I developed code on a small ARM base microcontroller to exercise the inputs and monitor the outputs on the embedded processor.

C# - Developed and maintained test framework. The framework is a generalized UI that was originally written for a watermark receiver. The UI was divided into three regions, test selection, status / non-modal feedback, and test progress / results. When used for audio encapsulator testing, the tests included ASI, serial, reference PLL, monitor audio, LAN connection, micro SD, etc. By just rewriting the test class, a new set of tests are easily implemented. Using a simple hashtable, the UI test names and database inputs are controlled with a text file. The program reads in the test names as the key portion of the hash and assigns a value to be used as control indices. The system then only needs to do a generalized loop to call the tests. The final iteration of the framework will have the test class moved into to a dynamically linked library, thereby reducing the compile to only the tests library.

Classes developed for the system include logging, networking, Digital Fourier Transform, ini file reader, audio tone evaluation.

C/C++ - Developed a Linux library extension for Ruby. The library contains system semaphore, mailbox, and mutex functionality. This allows Ruby to interact with other programs written in different languages and make use of the Linux system not supported directly with Ruby, i.e. serial port, parallel port, audio system, etc.

Ruby – Created scripts for testing audio encoders. The test system consisted of integrating an audio encapsulator, audio digitizer, RF modulator, RF upconverter, a satellite receiver, an Audio Precision audio tester. The system was controlled via Ethernet. The Audio Precision required control with GPIB and was accomplished using an Ethernet GPIB module. This reduced all the communications down to TCP/IP and telnetting. Easily handled with Ruby. The audio tests consisted of THD+N, crosstalk, amplitude, noise, digital THD+N, and digital amplitude. The Ruby scripts

included a small internal scripters to control the AP via the GPIB, i.e. the GPIB commands were maintained in separate text files. Raw audio data sweeps were captured from the AP and logged. The scripts then sorted through the raw data and filled a spreadsheet, and finally the spreadsheet results were parsed into pass/fail.

Ruby – Created scripts for controlling a Nikon DSLR. The system utilized a small Linux library extension written in C to read serial data from a Garmin GPS module. Then with the aid of the semaphore extension, the script interacted with a separate camera control program written in C. The system read in the GPS coordinates and applied the Vincenty Formulae to calculate the distance since the last camera shot. Instead of time lapse photography the system is now space lapse photography.

Ruby – Created scripts for controlling streaming audio. The script reads in a playlist in the form of a text file and feeds this information to an audio streaming program. The streaming program is written in C using an open source library. The communication link between the script and the streamer is the semaphore extension. Semaphores control the playlist data to the streamer with a mailbox holding the path and title to the next song.

C – Developed a timeslice / preemptive RTOS for controlling paving equipment. This was a replacement system for one implemented with a PLC. Specifically, the program controlled the mechanism for loading “tie” bars into an insertion rack, the insertion phase, and finally the recovery of the rack to the initial position. Sensor inputs included inductive proximity sensors and paired encoders to track the paving machine centerline / progress.

C – Wrote a round robin OS for controlling underwater paving system. The system controlled the forward plate attitude to sequester water channel blow outs in front of the main underwater paving pan. Inputs included magneto restrictive sensors to measure the plate height. User interface was implemented with an RS232 I/O terminal.

API – Specified the API data structures and procedures for a silicon tuner.

Office tools – Daily usage of Microsoft Word – or OpenOffice Writer, Microsoft Excel, and Access. Other tools include Gnuplot, GIMP, text editors, IDEs, etc.

Lab equipment – oscilloscopes, signal generators, network analyzers, IF and RF modulators, satellite receivers, spectrum analyzers, thermostreams, multimeters, video test generators, data generators, soldering equipment, hand tools.

Patents – crown height control for slipform pavers, front accessible Printer / Fax / Scanner, keyboard snooper, self-identifying remote control device, context sensitive remote control groups.