

Bernard Boudet

Contract Embedded Software Engineer

Full contact details available upon request.

Location: Cambridge, UK.

Latest update: <http://boudet.net/cv/>

Profile

A highly experienced senior electronics and software engineer with a broad skill base. Enjoys working as part of a team, unsupervised, or providing the technical lead as required by the project. Takes pride in finding innovative and practical solutions to problems with emphasis on simplicity and reliability. Can demonstrate a sound knowledge and intelligent use of software tools and test instrumentation with a methodical approach to problem solving.

Extensive practical experience covering hardware and software design and implementation developed over 25+ years working in consumer, automotive, scientific and engineering applications. Hands-on experience of planning, specification, design and development of various control systems and experience of large projects, mission-critical and safety-critical systems.

Key Skills

Embedded systems design; programming (C/C++, assembly, scripting); software development practices; analogue and digital electronic circuit design; low-power RF applications; serial data communication protocols; hardware interfacing; use of advanced test instrumentation (Oscilloscopes, FFT analysers, etc.); UNIX/Linux systems (Debian); GNU tool-chain; PCB layout.

Career History

March 2010 – present

Director / Contractor Consultant, Gnutronics Ltd.

Freelance design consultancy and project support to a range of small and medium enterprises across several industry sectors. Providing core services of embedded software design, systems engineering, and bespoke systems development.

Using a background in both software and hardware design, control systems and electronic product development able to quickly understand business requirements and to offer practical advice and engineering solutions through each stage of the design cycle.

Areas of expertise and services offered include:

- Embedded real-time software development
- Firmware, bootloader, and device drivers
- C/C++, Asm, scripting, build environments
- Real-time operating systems (RTOS)
- UNIX/Linux systems, Cygwin, GNU toolchain
- Electronic circuit design and evaluation
- Low-power RF, wireless ISM systems
- Multiple embedded platforms:
ARM-Cortex, AVR, PIC, MSP430
- Proof-of-Concept, Product demonstrators
- PCB layout and prototyping
- Production readiness, Design for manufacture
- Hardware / Software integration
- System testing, Code reviews
- Safety-critical applications
- Encryption and authentication methods
- Data communication protocols
- Scientific instrumentation

Career History (Gnutronics Ltd., continued)

Examples of works undertaken:

Home automation – Firmware design of a wireless thermostatic radiator control valve for a home energy management system. Production-ready firmware produced from scratch to an agreed specification and successfully delivered within a tight schedule. During the course of this work a hardware change was also recommended to the client and adopted enabling BOM unit cost savings. The firmware application used and conformed to Sentec's OpenHEMS/OpenThings messaging protocol. Hardware: 8-bit PIC16 microcontroller, HopeRFM69 transceiver. Software tools: MPLAB-X and XC8 compiler.

Telecommunications – Linux systems programming with code written mostly in C. This assignment successfully delivered specified requirements for Operation Administration & Maintenance (OAM) system features including SNMP and command line (Telnet) interfaces for a satellite communications gateway. The software platform was based on Redhat GNU/Linux and the Enea Element middleware framework. Software tools used included the GNU tool-chain, Jira issue tracking, and Coverity static code analysis.

Handheld electronics – Firmware development for a wireless, battery-powered instrument used for procedure compliance in the food industry. This product was already being used in trials and the assignment was to improve its robustness and usability in readiness for final production and market. Firmware improvements were made to keypad and display responsiveness, battery life, and wireless communication reliability. Hardware: 16-bit MSP430 16-bit microcontroller. Software: GNU tool-chain, Git, Bugzilla, and Jenkins CI.

Scientific instrumentation – This assignment delivered a complete firmware design and implementation for a complex application from scratch. The application combined high-speed analogue signal capture, optical sensor control, stepper-motor control, and temperature regulation. The software design encompassed hard real-time elements, asynchronous communication with a PC-based user interface, and a byte-code interpreter for executing user defined scripts directly on the microcontroller. Hardware: Atmel AVR XMEGA 8-bit micro. Software tools: AVRstudio, gcc, Doxygen.

Wireless monitoring – This product was targeted within a medical environment for monitoring the use of hand-wash dispensers, consisting of up to several thousand wireless nodes and a number of gateway/hubs collating data to an internet-based server. The assignment involved bringing up the gateway/hub hardware platform from “bare metal” to a functioning state, producing a low-level library for the RF transceiver, design and implementation of the gateway/hub firmware, and systems engineering support. Hardware: 32-bit ARM-Cortex/STM32, Atmel AT86RF212 RFIC. Software: Rowley CrossWorks Tasking Library.

February 2008 – March 2010

Senior Product Development Engineer, Zimiti Limited.

Designed and developed various low power radio technology products including a battery-powered baby and toddler monitor, and an industrial data telemetry product. Involved in all aspects of product development from concept through to hardware and firmware design, and product manufacture. Responsible for firmware design and development, configuration management, software releases, and elements of the hardware design.

Programming in C and assembler for 8-bit microcontrollers (Microchip PIC, Atmel AVR ATmega and Si. Labs. C8051), using development tools including MPLAB, HI-TECH PICC-18, AVR Studio, GCC, Keil C51 and Cygwin. Source code and documentation administered using Subversion (SVN).

October 2007 – February 2008

Software Engineer, TRW Automotive. (short term contract)

Electrically Power-Assisted Steering (EPAS) – Undertook various software development activities included translating system requirements into code, bench testing of steering column hardware, analysing problem reports, and code reviews following a formal software development process. Software tools used included Keil compiler, Vector CANalyzer/CANape, and AllChange configuration management. Application written in C and assembler, targeting an Infineon C167 (ST10) 16-bit microcontroller.

Career History (continued)

July 2005 – June 2007

Software Engineer, Pi-Shurlok Limited

Electronic Steering Column Lock – Various software design activities to implement new safety-related functionality to an existing ECU application. Tasks included client liaison, maintenance of the software design requirements, code reviews, software test specification, bench testing and hardware integration.

Remote Vehicle Immobiliser – Software development for an existing ECU application to enable stolen vehicles to be immobilised via the GSM network. Complete software module written in C, developed from a software design document. Main responsibilities for design review, module software coding and integration.

Prototype control unit for an aircraft engine – Architecture development for a prototype engine control unit for a major aircraft engine manufacturer. Responsibility for resolving hardware interfacing and electrical issues, as well as organising and maintaining a test bench facility. Worked with software engineers on the team to understand the control strategy and conduct software testing through incremental phases of development in order to identify and document failures and issues of concern.

September 2003 – July 2005

Software Engineer, Observatory Sciences Limited

Developed the primary mirror (M1) control software for the Visible and Infrared Survey Telescope for Astronomy (VISTA). Involved in design, programming, hardware interfacing, creation of user documentation, and testing. As sole programmer, developed code in C using a client supplied software framework and hosted on a HP-UX workstation, created automated test scripts with BASH, and used CVS for version control.

Target hardware was a PowerPC/VME platform running vxWorks, interfacing via CANopen to Beckhoff modules to provide several hundred digital and analogue I/O signals for the actuator and sensor hardware.

November 2001 – June 2003

Software Engineer, Pi Technology Limited

Assisted in the delivery of several projects developing embedded software for automotive engine management systems and powertrain control applications using a wide range of development tools and architectures.

Engine control unit (ECU) for Aston Martin – Software programming in C, following a structured development process, and MISRA coding guidelines. Development under UNIX, using Greenhills cross-compiler and ClearCase version control. Target hardware was based around a Motorola PowerPC CPU.

'OpenECU', Pi commercial product development – Developed low-level software functions in C to support a rapid-prototyping product for auto-coding engine management applications through Matlab/Simulink models. Software tools used include Greenhills, Diab and GNU cross-compilers, Cygwin utilities, SDS debugger, with Microsoft SourceSafe. Target hardware was based around Motorola PowerPC and M68HC12 CPUs.

October 1998 – October 2001

Senior Systems Engineer, Control & Information Section, LUL

As Lead Engineer responsible for the delivery of enhancements to platform information displays, developed and maintained the passenger information software suite whilst ensuring that delivered systems met both project requirements and customer expectations.

October 1989 – October 1998

Equipment Development Section, London Underground Limited

Mostly working closely within a team carrying out fault investigations of railway signalling equipment, design of replacements for ageing system components, and the evaluation of new equipment. Achievements included the development of a software suite for the control of dot-matrix passenger information displays, and detailed circuit designs for signalling equipment replacements.