

Resume of Robert Krten

President, Iron Krten Consulting
72 Cambior Crescent, Kanata, ON, K2T 1J4, CANADA
+1 613 271 2936 – rob@krten.com

Background and Objectives

I am a multidisciplinary problem solver with an unusually wide and deep knowledge of software and development processes. I present my ideas clearly, eloquently, and at a level appropriate to audiences spanning from end users, through experienced programmers, right up to C-level executives. I design novel solutions that are often surprisingly elegant while being highly performant.

The key is to find the simplicity in even the most complex problems.

I can do this because of my breadth of experience. Throughout this resume, you'll see the diversification in terms of both the positions I have held (e.g., **technical writer**, **business owner**, **software developer**, **systems architect**) as well as the areas of expertise in which I have practised (e.g., **medical devices**, **realtime control systems**, **telecoms**, and **security**). I have often been recognized as the key contributor to the success of the projects I've been involved in – look at my [LinkedIn](#) profile and recommendations. I have 3 pending, and 6 granted patents ([CA 2.874.277](#), [US 9.407.442](#), [US 9.454.666](#), [US 9.798.867](#), [US 9.934.374](#) and [US 10.459.851](#)).

Challenge me. Give me a difficult problem to solve.

I am interested in a position where I can continue to make real, significant contributions. I feel that the best way for me to accomplish this is as a consultant – someone who can be productive immediately, doesn't get involved in the politics, and can focus on accomplishing the task at hand.

I invite you to visit me at www.ironkrten.com for more details.

Experience History

(overlaps indicate multiple concurrent contracts)

SEP 19 – MAY 20 (projected end date) **<confidential>** (contract)

Company (Kanata/ON/CA): Responsible for creating documentation for a new embedded **microkernel operating system**; duties include getting input from subject matter experts and creating new content (markdown, drawings, and Doxygen). I ask about corner cases, what-if scenarios, and my favourite question, “**why?**” -- this facilitates design and development by forcing developers to think through their designs and articulate the (sometimes subconscious) design choices made. I then capture these, producing documents for both end users (the “how to” type of guide) and internal developers (the “how does it work” type of guide). Once a topic is explained such that I can understand it, I write about it and share the design, best-practices, and sometimes even the “wow!” factor straight from the architect / designer.

MAY 18 – JUN 19 **<confidential x 2>** (contract)

Two concurrent contracts with confidential companies, both based in the Bay Area.

Company 1 (Mountainview/CA/USA):

Responsible for creating documentation for a new embedded **microkernel operating system**; duties include getting input from subject matter experts and creating new content (markdown and drawings). My value add here is that I **worked remotely** and with little supervision; generally I

was pointed at a bunch of code and told to “write something.” The audience level was mostly intermediate to expert, **C/C++ device driver** types. I also **wrote sample drivers** to illustrate concepts being discussed. This contract was 40h/week.

Company 2 (Fremont/CA/USA):

This concurrent contract was on the order of 10h/week, and was a remote consulting role for a **QNX 6.6** based **semiconductor manufacturing product**. I reviewed designs, helped with the system architecture, and coded small parts of the system (e.g., **Simulink QNX driver framework** so that MathWorks models could be run in QNX).

FEB 17 – MAR 18 Best Theratronics Limited (contract)

I was the lead software architect for the Mark I Single Tube Irradiator. This is a **medical device** featuring a Spellman high voltage (160kV / 30mA) power supply and Comet X-Ray tube. I was responsible for generating the **software architecture**, detailed design documents, and producing the code. The architecture involves a **QNX 6.6** kernel (with four custom **drivers**) communicating with an ARM M3 I/O board. I designed the **protocol** between the two CPUs using COBS, and wrote the drivers as well as the control software. Initially, since the hardware wasn't ready yet, I wrote simulators for the Spellman power supply and the ARM control board. Wrote a WebSocket <-> PPS server to speed up the GUI operations. Total of around 27KLOC and 200 pages of documentation.

JUL 16 – JAN 17 Cisco Systems Limited (contract)

Working in the field of model-driven telemetry, I'm currently researching the integration of Google RPC (**gRPC**), Google Protocol Buffers (**gPB**), **openconfig**, (specifically Google's Network Management Interface (**gNMI**)). Major work includes over 3500 lines of C++ **code generator / recursive descent parser** to parse protobuf files and automatically generate gRPC/gPB protobuf C++ -> C and C -> C++ marshalling code. Automatically generates the server class wrappers, two special hierarchical input dump mode handlers, one for text and one for **JSON**, and a JSON input digester. Wrote a **universal test client** that accepts JSON and converts it to protobuf gRPC messages, sends them to the server, and converts the reply from the server to JSON.

JUN 16 – NOV 16 Merizzi Ramsbottom & Forster (contract)

Technical drafting for **patent application** preparation and prosecution, technical consulting therefor, and for **licensing and litigation** activities and other technical consulting tasks.

NOV 14 – JUN 16 QNX Software Systems (contract)

Engaged by QNX to write marketing and technical documentation relating to QNX's latest products. Provided domain expertise in medical, security, automotive and realtime micro-kernel.

AUG 13 – JUL 15 Irdeto Canada (contract)

Formulated the **requirements**, and generated the **architecture** of the next-gen **security framework**. Major requirements include **automatic security injection** into arbitrary customer code, support infrastructure, **modular-yet-entangled security primitives** model, next-gen integrity verification, instrumentation and tracing, **multi-threaded defense architecture**. This framework provides **security as an emergent property** of the supplied code. Filed **4 solo patent** applications. Supplied code for efficient interprocess communications, and implemented novel object storage database. I interfaced to product management, transcoder and binary teams, and internal end-users.

DEC 13 – AUG 14 Broadcom Corporation (contract)

Wrote WICED Development System Reference Manual / Cookbook and related **documentation**.

APR 12 – JUL 13 Cisco Systems Limited (regular, full-time employee)

Technical lead in **IOS XR** team, working on the **QNX kernel** and virtualization. Major projects include a Virtio block **paravirtualized driver** for QNX (hosted on **Linux/QEMU/KVM**). I'm responsible for maintenance of the **QNX kernel** used on the largest **core routers** (e.g., CRS family). I manage QNX

consultants to work on bug fixes, diags, and enhancements. I worked on really difficult, hard-to-debug "big system" **scalability** and interaction problems with geographically diverse teams (e.g., India, China, Japan, California, Europe), exercising effective **time-zone management** during my 16-hour EST/EDT interface window. 1 solo patent (US 9,798,867).

AUG 10 – APR 12 Irdeto Canada (formerly Cloakware) (regular, full-time employee)

Team lead on the binary technologies tools. Responsible for architecting and implementing low-level **security tools**, as well as interfacing with product management, test, and documentation. Gained deep experience with **ELF**, Windows **PE**, and **COFF** file formats. Designed an **integrity verification** system that was substantially smaller and more efficient (2 orders of magnitude faster) than the previous version, and featured many innovations to address customer concerns. 2 joint patent submissions (US 9,454,666 and US 9,934,374).

APR 08 – AUG 10 Semiconductor Insights (regular, full-time employee)

Worked on **software reverse engineering**. Responsible for taking systems apart, extracting the software, and creating a high-level representation of the **assembly language** instructions. Also responsible for some of the R&D tools, most interestingly, the research into a **decompiler** that takes arbitrary assembly language and converts it to C. Additional responsibilities included **patent analysis** in support of matching claims against high level code constructs, and providing claims documentation for end clients. Experience with **ARM**, **PIC**, **PPC**, **x86 processors** and WinCE, REX, and proprietary OSs.

MAR 07 - AUG 07 SynCardia Systems Inc. (contract)

Worked on software components of the SynCardia CardioWest temporary Total Artificial Heart, a class III **medical device**. This was a **QNX/Neutrino**-based project. I participated in the **architecture** of the software components, and **implemented** the Data Manager, Alarm Manager, Watchdog Manager, as well as a system logger. The Data and Alarm Managers are responsible for getting inputs from various parts of the system, and writing the data to persistent storage as well as further distributing data to other registered modules.

NOV 06 - JAN 07 QNX Software Systems (contract)

Worked in two main project areas, Board Support Packages (**BSPs**) and the QNX **Graphics Framework training course**. The BSP work involved verifying functionality, writing support code, building images, and even **fixing hardware**. Worked on BSPs for the KaRO Triton PXA270 (ARM-based eval board with LCD), AMPRO Littleboard (x86), and an e300-based PowerPC custom board. For the training course, I authored just over 300 slides for the QNX Graphics Framework, a separate TDK (Technology Development Kit) that features 2D and 3D graphics for embedded systems featuring an embedded version of **OpenGL**.

APR 04 – JUN 06 MDS Nordion Limited (contract)

Intense design and development contract involving three **medical devices**. I was the **project lead** on the **design** and **implementation** of the software for the Gammacell 40/1000/3000 (a blood irradiator). Responsible for the **software architecture**, **functional specification**, **traceability matrices**, **detailed design docs**, **coding**, **simulator**, **embedded web server**, and **subcontractor management**. I also helped with the software verification **test plan** of the Avanza Table project, and worked on the Avanza Table interface software, performed code reviews, and helped to diagnose and debug the code for the Equinox Radiation Therapy machine. The Gammacell and Equinox were based in **C** on the **QNX RTOS**. Software development lifecycle process was roughly equivalent to what would today be called IEC 62304.

FEB 04 Insight Control Systems (contract)

Helped to **port** over 350k lines of code from QNX 4 to QNX Neutrino 2 on a **process control** system which performs bottle inspection, consisting of several dozen closely cooperating processes. The main challenges were moving from Watcom C to GCC, **rearchitecting** proxy concepts to use Neutrino's pulses, moving the individual drivers to be resource managers, and general "how to" **consulting**.

SEP 03 – DEC 03 **PARSE Software Devices** (owner)

Wrote "[The QNX Cookbook – Recipes for Programmers](#)" **book**, which consists of 6 major projects explained in detail, including a RAM-disk **filesystem**, **data acquisition** driver, **high-availability** component, and others. Performed **publishing, marketing and sales** for the book, finally selling the complete rights to QNX (the company) – it's available for free on their website. Also **wrote, published, marketed, and sold** "[Getting Started with QNX Neutrino 2](#)", 500 pages, with Japanese and Russian translations. Rights sold to Cisco and QNX; book now available on QNX website.

JUN 03 – SEP 03 **Century Aluminum of Kentucky** (contract)

Worked on **data acquisition drivers** for PCL-711, ISO-813, and DIO-144 cards under QNX 6.2. Created a higher level **database server** (ADIOS, for Analog Digital I/O Server) that read the values from all installed drivers and put them into a **shared memory interface** which is organized as a **ring buffer**. Created **API** and **utilities** for managing tag names and ring buffer contents.

JUN 02 – MAY 03 **Cisco Systems** (contract)

Updated Cisco IOS 12 device driver **documentation**. This involved finding my way through literally millions of lines of code, coordinating with various subject-matter experts, and writing and **organizing** the documentation base.

JUL 01 – DEC 01 **Photuris Inc.** (contract)

General consulting to their QNX 6 development team, as well as the **production of miscellaneous tools**, such as "big brother", which is a **system monitoring tool** that detects memory overruns, stack usage, CPU usage, etc., and then takes actions based on a script file. Another interesting tool allowed multiple threads to be identified by **symbolic names** within multi-threaded programs (especially useful when some threads are loaded from DLLs).

FEB 01 - MAY 01 **Olin Chemical Corporation** (contract)

Analyzed proprietary OMNX **industrial control** system for **scalability** and **architecture**, design, and implementation. Analyzed sourcebase for correct design principles, wrote **regression testcases** and **universal testing client** (allowing arbitrary messages to be sent from command line). **Created** a reflective shared memory driver for ARCNet to allow two QNX 4 nodes to **share memory over a network**.

JUN 00 - JAN 01 **Cisco Systems** (contract)

Developed **courseware** for the **CRS** router project (then known as **HFR**). Responsibilities included scoping training materials, rounding up SMEs, parsing design documents, **courseware preparation and presentation**.

OCT 99 - JUN 00 **QNX Software Systems Limited** (contract)

Designed a **high-availability infrastructure** for the QNX Neutrino (QNX 6) operating system (everything from cold/warm/hot standby to **in-service upgrade** strategies to **load-sharing** and **load-balancing**). Wrote three chapters of a **device-drivers** "developers" book (USB, io-net, and the graphics infrastructure).

JUL 98 - AUG 98, JAN 99 - SEP 99 **Cisco Systems** (contract)

Worked on **Cisco GSR 12000 drivers**, including a **driver** for a high-speed (**gigabit** range) **switching fabric**. Also worked on Primary/Standby **arbitration module** for redundant RPs.

NOV 97 - APR 98 **MATROX Electronics Systems** (contract)

Ported MATROX's "Genesis" **frame grabber** software written for Windows/NT to QNX 4, and also made notes of what needs to be done for QNX Neutrino (QNX 6).

NOV 95 - DEC 98 **QNX Software Systems Limited** (contract)

Developed and presented two courses, "Realtime Programming under Neutrino" (4 day **operating systems internals** course) and "Writing a Resource Manager" (3 day course that continues the first

course focusing on **device drivers**, using home-automation X10 and a ".tar" **filesystem** as projects). Also, **developed the prototype** version of the "qnet" native **network manager** for QNX 6, including a simple **transport layer** and **kernel interface**. Finally, wrote a good chunk of the "Building Embedded Systems" manual, which describes how to develop programs and systems for the Neutrino OS.

NOV 96 - JAN 97 DIPIX Technologies (contract)

Ported the DIPIX XPG **frame grabber** device driver from Windows/NT to QNX 4. Contract involved **DMA** and **interrupt handler** development work.

JAN 95 - DEC 96 Bell Northern Research (8144) (contract)

Taught 3 day 'Introduction to Call Processing' course twice per month. This course gives an overview of all aspects of **DMS-100 call processing**, from originator going off hook to digit collection, translation, routing, termination, billing, supervision, etc. Also presented this course in India (Bombay & Bangalore).

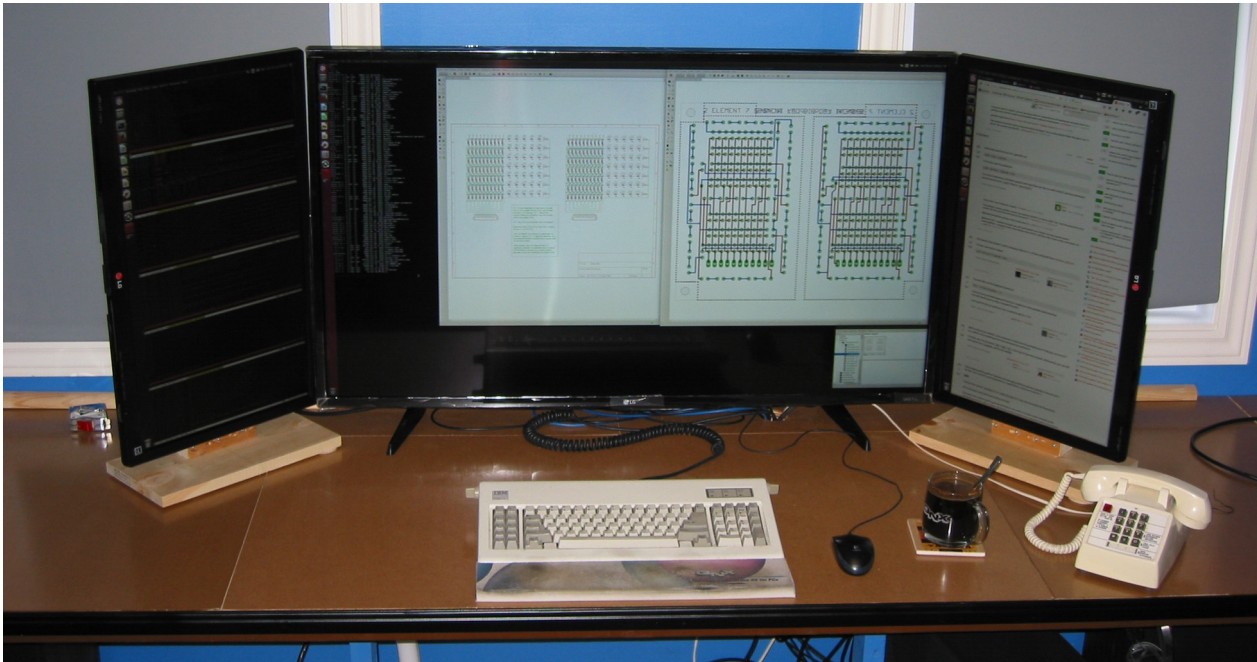
SEP 94 - JUN 95 Bell Northern Research (7P24/7P31) (contract)

Automated **DMS-100** testcases using T language and the AGE **automation package**. Testcases developed under HP/UX with custom BNR tools. **Created** the T-Language Analyzer, a sophisticated 'lint'-like package for the T-Language, consisting of a **preprocessor**, tokenizer, **recursive-descent parser**, and multiple analyzers (wrote over two weekends after management said it would take at least 6 months).

Personal Projects

I've been involved in many personal projects over the years (hyperlinks to project articles):

- [virtual filesystem for the efficient management of USENET news](#),
- **virtual 64-bit filesystem** layered on top of a 32-bit filesystem,
- 8086 instruction set **simulator / disassembler**,
- **disassembly and reverse engineering** of Mitel SX-20 PABX firmware,
- **disassembly and reverse engineering of PDP-8 diagnostics routines**,
- **hardware repair of TTL minicomputers (PDP-8 family) and their peripherals**,
- **synthesizer emulator that simulates music synthesizers for rendering MIDI files**,
- complete **design** of FIFO-based 16-bit stereo A/D & D/A **interface cards and drivers**,
- **Caller ID software modem**
- **design and implementation RGB LED PWM UDP-based light controller H/W & S/W**
- learned **C++** by writing a VI clone from scratch ([part1](#), [part2](#))
- **automatic stock trading AI with options analysis** using [adjacency matrix stock correlation](#)
- [XML/HTML/SGML C++ parser with higher-level analyzer plugins](#)
- custom-designed **printed circuit boards** for data acquisition, display, and control
- video motion sensor software and streaming TCP based server



I'm a hardcore developer; this is my desktop :-). It features a 4k TV used as a monitor, and two 1080p monitors in portrait mode. I built the monitor stands (not terribly impressive), but I also completely refinished the office and built a U-shaped custom bench (more impressive) with LED lighting underneath it for easy tower computer maintenance (slightly impressive, but really useful). I have a real keyboard, and yes, the phone works – it connects to my SX-20 PABX. The monitor shows a PCB layout and schematic.