

Howerd Paul Oakford

Curriculum Vitae, 2018 Jun 22

Born : 28th October 1954

Married, two children, two step-children.

1965 - 1972 Lowestoft Grammar School, Lowestoft, Suffolk. A-levels in Physics, Chemistry, Applied Maths, Pure Maths.

1973 - 1976 [Trinity](#) College, [Oxford](#) University, MA in Physics, specialising in Nuclear Physics and Electronics.

1976 - 1977 Test/Design Engineer for Computer Electronics Ltd., Saffron Walden, Essex. Repair/servicing of the CE6000 range, design of Prom Programmer board and 8 channel "intelligent" serial comms. board for Texas mini-computer.

1977 - 1981 Chief designer/programmer for Rofin/Sinar Technology on the Moisture Computer. The design won the British Microprocessor Award (1980) and is mentioned in "Starting Forth" (page 4). Programming in microForth on the CDP1802. Specialised hardware includes patented vibrating spring mass transducer and very low drift capacitance circuit.

From 1981 Self-employed consultant in software and hardware development and system design of specialist instrumentation. Trading as [Inventio Software](#) since 1985, Director of [Inventio Software Ltd](#) since 7th July 2000.

Participant and contributor to many [EuroForth](#) conferences.

Skills Summary

Protocols : Modbus , OBDII/CAN , HTTP, FTP, TFTP, TCP, UDP, IP, PPP, HDLC also GPRS (GSM) Sndcp and LLC/MAC layers using Anite scripts (Racal 6103), Unix, Unix shell scripts, Lauterbach and Anite tests and scripting. Linux (Ubuntu) and Windows based test software. Proprietary PPP.com Internet Protocol test program.

Programming Active Server Pages (ASP) using JavaScript and VB script, together with SQL database access.

Programming in Forth : ANS Forth, chipForth, polyForth, HolonForth. HolonJ Forth-like Java compiler. Embedded systems programming where performance and reliability are important.

Programming in C, C++ , perl, VB : Eclipse, ARM DS-5, Keil C51, MicroWare UltraC, IAR, MS VC++ Developer Studio, Borland C++ V5.02 + TASM , PLM, PVCS, Paradigm Locate, RT Debug + PDRemote, SMX V3.3.2 RTOS, EBS RTIP TCP/IP Embedded Network Software, Nucleus RTOS, SMX RTOS, Hitop5, D.A.v.E., Atollic, GNU ARM, GCC, Go.

Programming in C,C++, Assembler using : Intel 80x86, Infineon XC161 and XE164, ARM 7/9/Cortex M0+/ M4/A5/Vybrid, MSP430 and 8051 family, Renesas SH7254x and V850e , Motorola 6303, 68X0X0, 683XX, PIC series, Harris CDP1802, Yamaha YSS205B DSP, Texas DSPs. PC-based and embedded systems, chip-level programming. FFT implementation.

Hardware design using Futurenet, PADS, Easy PC/Pro, Ranger2, Proteus, especially microprocessor based instrumentation.

Special skills: low level drivers, STemWin , SECS/GEM II , Networking – USB, TCP/IP, UDP, Smartcards, secure serial communications, wireless communications. Apache HTTP Web Server configuration. Multi-tasking and multi-user programming. Virtual Machines. WAP, XML, WML, HTML. System design, including secure communications protocols, encryption, decryption and user interfaces. Multimedia / graphics systems. Smart Card Open Terminal Architecture including TLVs, cryptography, modem communications, and databases. GBCS/COSEM/DLMS/ASN1 Smart meter protocols. MSDOS, Windows, Linux, Cygwin, Unix and OS9 operating systems.

[Forth, Inc.](#) Advanced polyForth course, 11th January 1985.

Hobbies : Piano/keyboard, guitar, walking/cycling, OS development in Forth.

Howerd Oakford
Schlossstrasse 37
38448 Wolfsburg
Germany

UK Company Address
Inventio Software Ltd
Loudwater Farm, Loudwater Lane
Rickmansworth, WD3 4 HG
United Kingdom

Landline Germany	+49 5361 892 5421
Mobile Germany	+49 1762 262 3235
Mobile UK	+44 7954 580833
Skype	howerdo
LinkedIn	www.linkedin.com/in/howerd-oakford-5a69891
Email	howerd@inventio.co.uk
Website	www.inventio.co.uk

Patents

IMPROVEMENTS RELATING TO WEIGHING DEVICES (#EP0041967 B1)

<https://google.com/patents/EP0041967B1?cl=zh> February 1985

Vibrating mass balance for portable moisture measurement device. A weighing device uses frequency measurement to determine weight. One body (3; 23; 68) is of known mass and another body (8; 28; 66) also of known mass is spring (9; 29; 83) connected to it. This body carries the load whose mass is to be determined. Both bodies can oscillate in one mode in an effectively isolated system, being carried by springs (2, 13; 22, 33; 65, 67) from a support (1; 21; 60, 62). The oscillations are electromagnetically generated, but when a certain amplitude is attained the drive is cut off and the system resonates freely. The frequency varies with the mass of the bodies, and so a load on one can be weighed by reference to the frequency, which can be detected through the electromagnetic means (12, 32, 76) which provided the initial drive pulses.

Wireless remote control and position detecting system (#US5435573 A)

<http://www.google.tl/patents/US5435573> July 1995

Computer video game console. A wireless remote control system is provided. The remote control system includes a remote unit and a receiver unit, and may be utilized in connection with a video game system or other controllable system. The receiver unit includes a

plurality of detectors for detecting a signal transmitted by a remote unit. An angle-limiting device is coupled to each detector for limiting the signal which may be received by the particular detector to that signal which is transmitted from a particular location. In one embodiment, the angle-limiting device may limit the signal received by the detector to that signal transmitted from within a specified angular range relative to the detector. In a preferred embodiment, the angle-limiting devices coupled to the plurality of detectors may be arranged so that each angle-limiting device allows a signal to be received by the respective detector from a unique angular region. In another embodiment, the angular regions may be overlapping. In operation, a signal transmitted from the remote unit is received by one or more of the detectors, and based upon the strength of the signal received by each detector, the angular location of the remote unit relative to the receiver unit may be determined.

Design Experience

13th August 2018 to present

[ESE GmbH](#) / [Siemens AG](#), Braunschweig, Germany

January 2017 to July 2018

[Brunel GmbH](#) / [Siemens GmbH](#), Braunschweig, Germany (ST ARM, C/C++, Assembler, SwiftForth, Python, QT)

Developing an ARM and Altera FPGA based dual redundant train safety system at Siemens, Braunschweig, via Brunel.

IAR ARM C/C++ compiler and assembler (Windows), low level hardware-software interface on ST32F417 (Cortex-M4 core), including cold/warm start detection, FSMC interfacing. MD5 using Hash module, dual channel synchronisation and secure clock distribution.

Altera FPGA with NIOS 2 operating system, programming TCP/IP and UDP/IP servers. GCC compiler (Ubuntu 16.04) and ARM-GCC cross-compiler, using SSH and SCP to upload software to target.

SIL4 safety critical development. Display and monitor test tools written in SwiftForth, Python and QT.

October 2016 to January 2017

[Slock.it UG](#), [Mittweida, Germany](#) (Ubuntu Linux, ARM cross-compiler, Go)

Building Ubuntu Linux systems and ARM cross compiler toolchain for ARM-based industrial computers :

Openmatic's Bach-box, BeagleBone Black and NXP iMS6UL.

Compiling Go compiler for the target system, compiling Geth (Go ETHereum) client as part of feasibility study to show an Ethereum (crypto-currency) light client running on commercially available ARM-based hardware.

August 2016 to September 2016

[Sinar Technology Ltd](#) (Hardware development)

Development of Bluetooth Low Energy (BLE) / LoRa serial communications link, using BlueGiga and Semtek modules.

June 2016 to July 2016

[Xecnet](#) (Ubuntu Linux, Solidity, SwiftForth, Go, C++, Python)

Writing a proposal for a Distributed Autonomous Organisation (DAO) running on the Ethereum Blockchain system, for the [You-Me Drive](#).

Investigating the Solidity compiler and Ethereum Virtual Machine (EVM) bytecode.

Creating an EVM bytecode parser, to view DAO Smart Contracts.

15th February to 27th May 2016

[EDMI Limited](#) (ST ARM, MSP430, C/C++ , SwiftForth, ASN1-DER, Labview/TestStand)

Development of the ESME and GSME GBCS SMETS 2 compliant Smart Electricity and Gas meters. Responsible for the implementation and testing of GBCS Use Cases, in parallel with the EDM I test team, updating the meter firmware accordingly.

Development of PC based test tools : GBCS HTML document to source code converter, GBCS/COSEM/DLMS packet constructor and parser. Real Time Clock read/write simple GUI, using the meters' console test port, for testing clock drift.

Updating test process and system overview documentation.

NXP LPC Xpresso (Eclipse) ARM development environment, SVN version control, JIRA and JAMA project/bug control systems.

4th January 2016 to 18th January 2016

[Microprocessor Electronics \(MPE\) Ltd](#) (ST ARM , C/C++ , VFX Forth, Assembler)

Porting a "C library to Forth interface" to the STM32F429ZI-Discovery board.

Porting code from ARM Cortex M3 to M4 exception handler routines, and from an Eclipse IDE to GNU ARM command line compiler interface.

Creating a user friendly command line environment to compile ST supplied development board libraries using GNU ARM embedded C, to compile the Forth system using MPE's proprietary VFX xArmCortex Forth compiler, and to download the resulting Intel Hex files to the chip using the ST-Link Command Line utility.

Integration of STemWin library to make it accessible to the VFX Forth system.

2nd June 2015 to 1st January 2016

[Hanover Displays Ltd](#) (ST ARM , C/C++ , SwiftForth, Python)

Development of the Hanover Display's Electronic Route Indication Controller (ERIC) Generation 3, a destination and information display controller for transport systems.

Keil uVision ARM C/C++ compiler for the STM32F207 ARM Cortex M3 chip.

Modifying the FTP-based configuration and data update system of the ERIC G3 (EG3) for the Hanover Display's Wireless Configuration Management system. The requirement is to allow automatic updates of configuration, database and information files when an EG3 unit is replaced, and to inform the Wireless Configuration Manager of the current status of the EG3. This was achieved by uploading a set of status files, each one having a file name based on the EG3's ARM chip's Unique Identifier.

Rationalising the handling of configuration data, by defining one Configuration Parser that

takes a pointer to a file buffer or string as its input. The data source for the Configuration Parser can be from

1. a file downloaded over FTP (using the lwIP light-weight TCP/IP stack over Ethernet or WiFi),
2. a command string typed in over Telnet (again using the lwIP stack)
3. a command string or file sent over Hanover Display's proprietary Functional Test interface (used in production and test of the units), based on the RS485 port used to connect to the Hanover Display's range of LED signs
4. a command string or file as part of the ERIC.bin proprietary Database file.

Expanding Hanover Display's proprietary Functional Test interface to include tests for new EG3 features, including : Real Time Clock calibration, RTC time set and get, readout of the EG3's ARM chip's Unique Device ID etc...

Collation and updating of documentation for the Hanover Display's proprietary ERIC.bin Database file format, and Functional Test and FTP interfaces.

Porting of the Telnet interface code from Hanover Display's DERIC G3 system.

Porting of the Hanover Display's proprietary Hanauto interface code from the DERIC G3 system. This allows a test PC system to control the EG3 by simulated keypad button presses and analysis of the text displayed on the EG3's LCD. Installing and running a Python script to perform a defined subset of the possible Hanauto tests.

Applying a script-based code prettifier to address legacy file formatting issues (mixed Unix/PC line endings, use of variable size tabs, code indentation etc)..

Removal of many (~200 of ~1020) compiler warnings by adding appropriate type casts and other “cosmetic” code changes.

Developing a script-based GUI interface in SwiftForth, to allow custom test strings to be sent to the EG3 with a single click or key press.

April/May 2015

[Application Solutions \(Safety and Security\) Ltd](#) (Freescale ARM Vybrid, C/C++)

Porting of existing ARM Cortex M4 C/C++ Vikipedia serial driver software to the Freescale Vybrid VF65GS10 dual core (A5 and M4) ARM chip.

Configuration of Freescale MQX based program for the Vybrid A5 core to enable the M4 core.

Modifying interrupt-driven UART drivers in both an MQX RTOS queued environment and using direct register-level access.

The original Cortex-M4 chip used a different interrupt module to either of those in the Vybrid, changes to the hardware to use a different selection of on-chip and external UARTs and the addition of the MSCM (which routes interrupts to one or both of the A5 and M4 cores), made this project particularly challenging.

The Freescale TWR-VF65GS10-PRO Tower system was used as a test bed, pending new hardware, with the SER2 IO adapter connected to three of the Vybrid on-chip UARTs.

Debugging at a hardware level, right up to posting of MQX messages to tasks running a context-based common handler routine.

Eclipse development IDE with ARM DS-5 compiler, also using nmake and cmake from a command window, GIT version control system.

July 2014 to end of March 2015

[Landis & Gyr \(Toshiba \)](#) (ARM, C/C++ 9 months)

Development of DLMS/COSEM interfaces on the E470 Smart Electricity meter, using Code Warrior/Eclipse, IAR debugger and Segger USB JTAG.

Using the proprietary Vader C-based Object Oriented architecture.

Using DLMStool (proprietary DLMS tester), developing PC-based tests and GUI. Interfacing to test team using National Instrument's TestStand.

Analysing HDLC format 3 DLMS/COSM packets.

Code base Tortoise SVN version control.

Implementing GBCS specification for parsing of ANS.1 DER packets for secure key exchange, using ST Crypto Library interface, specifically the ASE-128-GMC used in GMAC authentication-only mode.

Running Python scripts to verify ASE-128-GMC packets using the PyCrypto, AES and GMAC libraries (Python 2.7 and 3.2 versions).

Development of SMETS2 compliant firmware using ST-Link, IAR debugger and Eclipse GUI and Cygwin.

Parsing Authentication Certificates using ASN.1 / BER / DER encoding/decoding rules using SwiftForth test scripts to scan 416 certificates.

May/Jul 2014

[AMIHO Technology Ltd](#) (Freescale ARM Cortex-M0+ C/C++/ Assembler 9 weeks)

Adding Semtech LoRa (Long Range) mode operation to the Amiho Wireless Meter-Bus EN13757-4 software communications stack. Working extensively on low level drivers for the Freescale/Kinetis MKL26Z128VLH4 ARM Cortex-M0+ microcontroller on the KL26Z Freedom board, using Code Warrior/Eclipse.

April/May 2014

[Sinar Technology Ltd](#) (MSP430 Assembler, SwiftX)

USB CDC device support programmed in Forth , with interrupts and an interrupt event queue written in MSP430 assembler. This allows USB events to be processed in high-level Forth in a low-power aware RTOS, with only a minimal loss in response time.

Extensive use of interrupts, timers and the MSP430 on-chip USB module, MQP Packet Master USB analyser and the USBclarify Windows USB display tool.

February 2014 – April 2014

[AMIHO Technology Ltd](#) (Freescale ARM Cortex-M0+ C/C++/ Assembler 8 weeks)

Porting the Amiho Wireless Meter-Bus EN13757-4 software communications stack to a variety of new Wireless transceiver chips and host processors, including the Semtech SX1231 and SX1276 LoRa series Long Range wireless chips.

Working extensively on low level drivers for the Freescale/Kinetis KL25Z128VLK and MKL26Z128VLH4 ARM Cortex-M0+ microcontrollers on the KL25Z and KL26Z Freedom boards, using Code Warrior/Eclipse, and optimizing the interface to Amiho's proprietary low-power radio stack.

November 2013 – February 2014

[Sinar Technology Ltd](#) (MSP430 Assembler, SwiftX)

Hardware and software design, PCB layout and testing of an MSP430 based upgrade of the Sinar range of Moisture Meters.

MSP430F5529 low power chip with USB support. The software for this project was ported from the existing 8051 based code base, with low level drivers written in MSP430 assembler.

Extensive use of interrupts, timers and the on-chip UCS, ADC and USB modules.

May 2012 – October 2013 (18 months)

[Suss Microtec AG](#) (C++, C#.Net, Visual Studio, Team Explorer)

Software support for the Suss Microtec range of semiconductor industry products – coaters, mask aligners, photomaskers etc.

Writing/debugging low-level device drivers for interfaces to third party equipment – Cybor pumps, Sensirion flowmeters, Trinamic stepper motor controllers etc.

Writing simulators, data capture loggers and display scripts in SwiftForth for the above devices.

RS232/RS485, TCP/IP, Ethernet, NetBEUI, SECS/GEM II protocols. NDIS promiscuous Ethernet port driver GUI.

RTMT Real Time Multi Threader C macro based task scheduler.

Software field support, located in Lowestoft, working with a team based in Sternenfels, Germany.

[Westweigh Ltd](#) (C/C++, assembler, 2)

Evaluation of ARM and MSP430 and 8051 based boards for use in Westweigh's CBC-02 updated Conveyor Belt Computer.

Addition of OBDII/SAE J1962 and ISO15765 CAN hardware to measure fuel levels in a mobile rock-crusher environment.

Using SparkFun - Arduino CAN-BUS OBDII Kit Arduino shields for development.

Using CANopen as source for Forth driver.

December 2011 – April 2012

[NXP Semiconductors Germany GmbH \(Hamburg\)](#), (C++, C#.Net, NUnit, Visual Studio)

Development of test environment and test programs for NXP contact-based and contact-less smart cards in C++ and C#/.Net

C++ is used for low-level interfaces, C# for the test harness, with MS batch files and some perl programs for glue logic.

Use and integration of NXP proprietary programs PathFinder, MiFareWnd and others to interface to Ashling SmartIce Emulator, NXP Pegoda and MicroPross MP300 TCL-2 USB card readers, and an NXP proprietary FPGA simulator.

TrueCrypt secure vault with Enovia DesignSync version control system.

March 2011 – November 2011 (8 months)

[Itron Metering Solutions UK Ltd \(Felixstowe, UK\)](#), (C, C++ ,Visual Studio , Renesas/NEC Multi)

Agile programming methodology (Sprint/Scrum). ClearCase Version control system, Enterprise Architect UML design environment, Eclipse and Multi IDE.

Programming of HMI and low-level device drivers for Renesas V850e 32 bit microcontroller in the new Itron Electricity meter.

EmBOS Real Time Operating System, COSEM/HDLC protocol.

COSEM demand averaging object drivers.

LCD driver, including scrolling. Flashing of text and icons

Token key low level driver using bit-banging of V850e port pins.

Table-driven HMI menu system (GUI), with exception handling using a state-machine in a single thread.

Font editor and formatting program for HMI display graphics.

March 2011

[Sinar Technology Ltd](#) (PIC16 Assembler – 2 weeks)

Hardware and software design using MPLAB, PCB layout and prototype build and testing of a PIC16F676 based voltage to frequency converter.

The PCB is the electronic part of a load-cell based replacement for the vibrating spring balance in the Sinar range of Moisture Meters.

The original spring balance produced a sine-wave signal with frequency varying with the mass of the grain sample.

The new design uses a strain gauge based load-cell to measure the weight of the sample, which is then converted to a varying frequency signal, compatible with the original.

The design features unusually high stability, with a chopper stabilized op-amp, and crystal controlled PIC clock.

August 2010 – February 2011

[Infineon Technologies Austria AG \(Graz\)](#) (C, C++ ,Visual Studio – 6 months)

Development of test environment and test scripts for Infineon contact-based smart cards in C and C++, using Infineon proprietary CCTop and CCHiTs test environments. Agile programming methodology. ClearCase Version control system. Linux terminal script

based low-level testing. ARM microcontroller with custom hardware peripherals. Configuration and interfacing to Smart Card readers through Visual Studio C++ executables invoked by high level scripts from CCHiTs. Testing of all aspects of the Smart Card, including security-critical components.

March 2010 – May 2010

[Markem-Image Ltd](#) (C, Visual Studio, Forth and assembler – 3 months)

Port of SmartDate5 embedded software from Infineon XC161 to XE164. Driver for XE164 SPI interface to Altera FPGA. Forth test harness and scripts to debug the embedded system. Modification to binary to run-length encoded C-source file in Microsoft Visual Studio. Configuration of embedded peripherals using the [D.A.v.E](#) tool from Infineon. Writing a source file comparison script to compare files created by D.A.v.E while ignoring data stamps (which always change). Hitop5 embedded development system, Tantino USB JTAG debugger. Debugging of FPGA generated timing signals using LeCroy digital storage oscilloscope.

March 2010

[Sinar Technology Ltd](#) (6303 Assembler)

Modifications to legacy software for the Sinar Grainspear, to merge Fahrenheit/Celsius functionality from V1.4 with averaging function from V1.5 to create V.6 software. Also source file archiving and “tidy up”.

July 2008 – February 2010

[Eltek Ltd](#) (Forth, C/C++ and MSP430 assembler)

Hardware design of an upgrade to the Eltek 1802-based Squirrel data-logger, based on the Texas MSP430FG4619 and MSP430F5438 micro-controllers using the FETU430IF USB JTAG debugger.

Writing low-level drivers for the new hardware, including LCD, keypad, RTC, buffered low-level Flash driver, SDcard reader, FRAM EEPROM interface, serial port and USB module in SwiftX for the MSP430.

Upgrading SwiftX for the MSP430 to include the MSP430X instruction set, and extending the compiler to cover 128K of program space.

Writing a 16 bit FAT12/16/32 file system for the SDcard.

Programming the Squirrel data-logger application in SwiftX Forth for the MSP430. Conversion of the original polyForth 1802 application (stored in Forth blocks) to files.

Extending the SwiftX Forth to use the MSP430X extended instruction set, and adding an 8 bit token threaded Forth layer to allow programs to reside in high memory.

Setting up development board using the IAR C compiler.

Evaluation of PIC18X development board and on-the-go USB drivers.

Programming PC based Windows USB driver software for the PIC18X (MPLAB) and FTDI USB module, using GUIDs and the Windows API.

May 2008 – June 2008

[Westweigh Ltd](#) (Forth and 8051 assembler, 6)

Programming of RS485 multidrop network protocol using the 8051's 9 bit UART on the IS5b processor board. Upgrading the software to the same functionality as the original 8051 hardware. Designing a custom RS485 multidrop protocol for interconnecting Conveyor Belt Computer units and other peripherals.

September 2007 – April 2008

[Renesas \(UK\) Ltd](#) (C and Assembler, 6 months)

Debugging low level Flash drivers for the Renesas SH7254x series of RISC processors, in particular for the on-chip Flash, MCU, ICU, GPT, CAN and SPI peripherals, using the Renesas HEW development environment with the E10A USB JTAG debug port interface. Creation of a test harness and test scripts using SwiftForth, and the conversion of the scripts into C.

Writing Requirements and Design Specifications and coding for a customer API using the AutoSar (Automotive Open System Architecture) interface layer at the Basic Software and Runtime Layers.

June 2007 – August 2007

[Thales \(UK\) Ltd](#) (C and Assembler, 3 months)

Debugging low level drivers for a Texas DSP.

April 2007 – 11th May 2007

[Penlon Ltd](#) (Forth – 3 weeks)

Writing test software for the Penlon AVS unit, as part of the software verification system for Penlon's anaesthetic ventilators.

January 2007 – March 2007

[Westweigh Ltd](#) (Forth and 8051 assembler, 3 months)

Development of the Conveyor Belt Computer (CBC). Upgrade to the Q.C. Technology product to use a more powerful processor, with added functionality such as RS485 multidrop, bigger LCD etc. Translation to Russian, including embedded font support, using the Forth Language Translation system (please see <http://www.inventio.co.uk/Multiple%20Language%20Programs%20Made%20Easy.htm>).

July 2006 – December 2006

[Reach Technology Inc](#) (C and Forth)

Development of the “PI” (Program Interpreter) graphics library interface. Based on the Atlast dialect of Forth, written in C.

PI provides an extensible scripting language and interpreter as an extension to the Reach Tech SLCD quarter VGA graphics modules.

Graphics widgets such as bar graphics, dials, and soft keypads can be defined and imported into user programs.

Also RTPC, the Reach Tech PC program, an interactive test environment that allows files edited on a PC to be downloaded to the SLCD unit.

October 2005 – June 2006

[Metric Group Ltd](#) (9 months, C , Assembler and Forth)

Accent/S4 project, development of an LCD device driver for an Atmel 89M55800 ARM based car park ticket vending machine.

Encryption for car park ticket time extension code using the TEAN algorithm, modified for 26 bit operation.

Font conversion programs, from vertical pixels per byte to horizontal, output as a C file etc written in Forth.

Hierarchical, extensible data file format, which can be converted to/from XML. This allows backwards and forwards compatibility between data files and embedded programs that read them. Nested “schema” definitions allow record data formats to be specified in a compact way.

June 2005 – September 2005

[Oxford Semiconductor Ltd](#) (3.5 months, SwiftForth)

Specification and development of OST, a script-based test environment to test device driver calls. ANS Forth was used because of its rapid development time, ability to interface to hardware / OS API's and its availability on most platforms. The same environment that tests device driver API calls can also be used to test the hardware directly.

SwiftForth was used initially as the ANS Forth compiler for the Windows XP/2000 environment, with Gforth (GNU) available for Linux, and dsForth available for WinCE for future ports.

A user-friendly GUI is provided, so that tests can be run by opening files using the standard Windows interface.

The test scripts can be edited with a standard text editor, and use a simple set of platform-independent function calls to test all aspects of the device's functionality. This project required internal knowledge of Windows Device Drivers, using the DDK, especially relating to USB and serial drivers for the Oxford Semiconductor USB950, USB954, PCI952 and PCI954 chips.

July 2004 – May 2005

[Marconi Selenia Secure Systems / Selenia Communications Ltd](#) (11 months, SwiftForth, SwiftX ARM, PIC C and Assembler, polyForth 8086, 8085)

Software and hardware development for a secure ISDN telephone. Modifications to

polyForth 8086 and 8085 (Z80) programs. PIC12F683, 12F675 and 16F684 program design, implementation and test using PICClite C compiler and MPLAB PIC Assembler for a CRC-protected non-volatile storage device. Test rig for the PIC device using an ARM EB40 evaluation board with SwiftX ARM compiler. CRC verification programs using SwiftForth on a PC. Investigation of ISDN protocols. Use of Agilent Logic Analyser. Integration of a new FPGA into the existing design. Use of Swiki to log progress, and the Tortoise SVN version control system. Note : Marconi Selenia Secure Systems changed its name to Selenia Communications in January 2005 and again to Selex Communications in August 2005.

February 2003- April 2004

[Alaris Medical](#) (14 months, ARM Assembler, C, C++, Rhapsody, UML)

Asena "Amigo" project. Port of the SMX RTOS to GNU C/C++ and the Motorola Dragonball M9328 ARM9 microcontroller. Port of Rhapsody IDF layer to GNU C/C++. Writing of device drivers for UARTs, timers, monochrome and colour LCDs, keypad, EEPROM, I2C, PWM sound generator, low level embedded FLASH driver, SDcard, boot protocol. The development environment included Lauterbach USB ICE, Cygwin/Bash, GNU C/C++ ARM cross compiler, Rhapsody UML to C++ state machine code generator, Rhapsody queue-based scheduler, UML documentation, PCVS (initially) and CVS (currently) version control systems, Object Oriented design.

October 2002 - February 2003

[Richmond EEI Ltd](#) (~16, F, A)

Abrasive Cutting Equipment (ACE) project. A four axis (x , y , extension and rotation) manipulator used to position a high pressure abrasive/water jet remotely over an unexploded bomb, to cut a hole and so remove the detonator and/or explosive. The system comprises three modules, located up to 500 m apart (for safety!), using an RS422 serial protocol based on HDLC for communication. The ACE protocol was designed to be backwards compatible with a "dumb terminal", so that each module can be tested without the other modules using a standard serial terminal. The two modules (Power and Service modules) use the Sharp ARM processor, programmed using the VFXForth ARM cross compiler and some assembler, and the Graphical User interface on the Command and Control module is programmed using VFXforth for the PC. Software for the modules is downloaded by Ethernet using a Flash management system, allowing simple field upgrades. The project created a fully functioning prototype to be submitted to a major customer for approval.

July 2002 - September 2002 (ongoing, part time)

[SME Paramedics Ltd](#) (HTML, ASP, JavaScript, VBscript, SQL)

Modifications to the [Nikon Fotoshare](#) Websites. general technical support and upgrades. Conversion of &-encoded characters to 8 bit ASCII and Unicode formats for the Russian site. Changes to ASP files which access the SQL server. Use of Terminal Services Client (Win NT) to upload modified ASP, HTML, FLASH and SQL database files to the server using HTTP and FTP. Use of FrontPage to edit ASP and HTML files. System documentation.

10 April 2002 - September 2002 (part time)

[Peak Performance Training Ltd](#) (~6 , FP, F VB A)

Programming of PC sound card and parallel port in real time to give sound and light display. User friendly MMI.

A prototype program was written in Visual Basic 6, but this did not allow easy access to real-time timer and I/O port DLL functions, so the code was converted to Win32Forth. An exercise in getting around Window's limitations in real-time programming.

8,9 April 2002

[Running Communications](#) (2 days, FP, ASP)

Installation and operation of a 45-seat ALPS system at the Hilton Hotel, Park Lane London (see below for details).

Eleven laptops and 34 HP Jornadas connected together in a wired/wireless Ethernet, accessing the ALPS database manager running on a server laptop. More than 240 questions were received from the conference delegates. Two Ethernet hubs (one 24 port , one 4 port), 34 Proxim wireless Ethernet PCMCIA cards and one Proxim Access Point were used. The whole system ran perfectly.

15 October 2001 - 29 March 2002

[Matsushita Mobile Communication Development of Europe Ltd](#) (Panasonic) (6 months, Ag, C, perl, Unix)

Design of the API for the Cursors used in the MMI section of the new GD67 mobile phone.

Detailed design and coding in C of the API and Cursor drawing using graphics primitives. Cursors are used by many of the modules and applications in the phone. There are a variety of different cursors, including "I-beam", left and right line cursors and inverse video. Some of the cursors may flash, so the Cursor API must work alongside the Timer API in the RTOS to ensure synchronisation.

Writing perl scripts to run under the Panasonic HMTE Host Mobile Test Environment, to test both the Cursor and Font modules.

28 August 2001 - 12 October 2001

[De La Rue Cash Systems](#) (3 months, Ag, C)

Porting of Teller Deposit Unit C source from Whitesmiths C compiler to the Mentor Graphics/Microtec C compiler.

Modification of C source to interface to the new 2520 Note Counter. Involved four different proprietary serial protocols - TDU to host computer, TDU to original counter, TDU to new counter, TDU to escrow unit. Each unit uses a different protocol. The job involved converting the old counter protocol to the new counter protocol.

The [Hitex](#) 68000 ICE was used to speed up the software development.

23 July 2000 - 29 June 2001

[Matsushita Communication UK Ltd](#) (Panasonic) (11 months, Ag, C, perl, Unix)

Design of software test scripts in C and Perl under Unix and Windows to test the [USNet IP stack](#) as used in the Panasonic GD35, GD36, GD93P and GD95 WAP mobile phones. This project involves an in-depth understanding of the Internet protocols used in the Wireless Application Protocol : PPP, LCP, IPCP, IP, UDP, TCP, ICMP, XML. The test environment is the proprietary Host Mobile Test Environment, which allows perl scripts to be run under Unix and the Nucleus RTOS.

Also testing of GPRS SNDCP protocol and LLC/MAC layers using the Anite scripting environment and customised Unix scripts, Unix and Linux based packet routing and modifying programs.

April 2001

[Running Communications](#) (2, FP, ASP)

ALPS - Advanced Lecture Presentation System. An ASP, HTML, VBscript and JavaScript conference "chat room" application, based on a Windows laptops and up to 36 Windows CE Hewlett Packard Jornada palmtops. Messages entered at the Jornada are sent using a Proxim Wireless LAN to the master laptop where they may be reviewed, edited and printed. The VBscript was used on the Laptop to access an SQL database, and the JavaScript on the Jornadas, with an ASP (Active Server Page) file on the Laptop which constructed an HTML file to be sent to the Jornadas. One installation at the ExCel centre in Docklands, London, used eight Laptop PC's and one Proxim Wireless Ethernet LAN port connected to a 16 port Ethernet hub, plus 36 HP Jornadas with Proxim Wireless LAN cards, all running ASP files from one of the laptops which supported the Microsoft Personal Web Server.

November 1999 to March 2000

[Hybaid UK Ltd](#) (5 months, Ag, C&PLM)

Programming of the MultiBlock System (MBS) in PIC C. The MBS is a programmable thermal cyler which can be networked on an RS485 line. It uses a PIC 16C77 to control the temperature of up to 384 samples, either to the same temperature or to a defined temperature gradient. The original program did not support the gradient mode of operation, and was extensively re-written to save program space (8K x 14 bit words) and to make it easier to understand, as the original was written as a state machine. To assist in the programming, reference was made to Hybaid's Express unit, which supports gradient operation, and was written in PLM. The methods used to make the program more compact were to factor all functions as much as possible, and to redesign the structure of the program to isolate the three areas of functionality : The RS485 serial interface, the temperature control PID loops and the user defined program.

March 1999 to mid-September 1999

[Chase Communications](#) Ltd (6 months, Ag, C)

Programming of the Chase SMR receiver using 68332 GNU C PC based cross compiler and GNU assembler. This is a dual 68332 processor system, with dual port RAM used to interface the sampling processor with the communications processor. Extensive use is made of the 68332's TPU (Timer Processor Unit) to provide interrupts from a variety of

distance and time sources, which are used to trigger programmable data capture events. The SMR also communicates over an Ethernet link using a subset of the TCP/IP protocol. The TCP/IP code is based on the NCSA public domain sources. The interrupt routines and much of the fast inner loops of the sampling processor are coded in assembler for maximum speed. The SMR unit interfaces to a PC program being written simultaneously by other members of the team, in Visual Basic.

January 1999

[Computer Solutions Ltd](#) (3, FP, C)

Programming, configuration and testing of TCP/IP networking software for the AMD Net186 single board computer. The Net186 is based on the AMD186ES chip, with an Ethernet controller and two serial ports.

The project included compilation of html files into a virtual filing system which allows the Net186 to be a Web Server, with html pages embedded in the Flash memory, and configuring the IP addresses for the network.

The tools used were: Borland C++ V5.02 + TASM + PVCS, Paradigm Locate, RT Debug + PDRemote, SMX V3.3.2 RTOS, EBS RTIP TCP/IP Embedded Network Software, all under Windows / DOS

December 1998 / January 1999

[MultiElectronics Ltd](#) (2, FP, F)

Audio control board based on the Yamaha YSS205B DSP chip and the Atmel 80C2051. The program was written in 8051 chipForth and controlled the 180 or so DSP control registers in the YSS205B. Four buttons, two LEDs and one potentiometer set the required functions, which were then processed into the DSP parameters to give key-shift, voice cancel and echo effects.

November 1998

[Quality Controls Technology Ltd](#) (4, FP, F)

The Titrette - portable Carl Fischer titrator. An 8051 based project based around the IS5 processor board, with additional analog hardware. The analog board was based on the MAX1241 12 bit ADC, and features scanning at 2400 Hz and averaging over 16 seconds to reduce noise. The hardware interfaces include a DC motor controller (constant speed using back EMF monitoring) , variable voltage PSU (digitally controlled), PID feedback control, AC conductance meter, and two RS232 interfaces. The software includes two tasks (background ADC monitor and user interface) , print spooler, and extensive user interface and test facilities.

September - October 1998

[RACAL Radio Ltd](#) (8, Ag, -)

An eight week contract working with a RACAL team on an encrypted transceiver project. The work includes verifying the operation of hardware and software across the product range : Mobile (for use in vehicles), Personal (hand held). and Covert (compact size), testing using the Tariff UDP (subset of TCP/IP) protocol, and designing test rigs to assist in the programming of the units. Also loading of encryption keys, repair and design

improvements to "key fill devices" (which load the keys), and testing the OTAC (Over The Air Control) system.

November 1997 to January 1998

[Hi-Gain Electronic Designs Ltd.](#) (3 months, Ag, C&A)

User interface and FLASH memory driver for a Portable Appliance Tester, programmed in Keil C51 embedded C .The user interface features a Finite State Machine architecture for maximum reliability in a safety-critical area, and handles storage of results, and reviewing, printing and downloading of stored data. The project also included a serial port and keypad interface, and a graphics LCD driver for T6963 controller chip, and assembler and C code for LCD driver for Toshiba TLCS870 series chip.

February 1996 to April 1997

EMV / [MicroProcessor Engineering](#) (13 months, Ag, F&C&A)

Forth implementation of V25/80186 version of Europay/MasterCard/Visa Open Terminal Architecture Virtual Machine for Smart Card terminals. Development of 32 bit Forth cross-compiler and 32 bit operating system to run on 16 bit processor, including "umbilical Forth" environment. Development of low level Flash drive code, including database and module support and live update of programs held in Flash. The OTA interpreter operates as a "virtual processor", consisting of approximately 300 instructions. The instruction set ranges from simple address/data access and computational instructions written in assembler, to support for Tag-Length-Value encoded data packets (including parsing of Data Object Lists of TLVs), volatile and non-volatile linked-list databases, cryptography (including RSA, SHA and Modulo Exponentiation), modem communications (X-Modem protocol) and executable module support, written in high level Forth and assembler. Several system functions including a system heap, databases and TLV code were translated from C into Forth. OTA instructions are provided to support different programming languages, including Forth and C. The OTA concept allows a program written in OTA token code to be validated once for execution on many different platforms.

October 1994 to December 1995

[Philips](#) / [MicroProcessor Engineering](#)(15 months, Ag, F&C).

LYNX project for CDI-players C and Forth interface program for Philips CDi multimedia system. C programs written using UltraC/PC Bridge, PC hosted cross compiler for OS9/68000. Forth programs written using ForthMacs on OS9/68000 system. Multimedia/CD ROM graphics application with extensive 68000 assembly language programming in OS9 environment, some Unix. The system comprises a platform-dependent B-wrapper layer, a platform- dependent C-Wrapper layer (both written in C) and a Forth scripting language interpreter/compiler layer. The B-Wrapper made extensive use of the Balboa graphics library, with syscalls and callbacks connecting the Forth text interpreter and Balboa multitasking O/S.

Philips LYNX project for PC . PC / CDI compatible multimedia graphics scripting language. C and Forth interface program for Philips Video CD+ PC-based multimedia system. C programs written using Microsoft Visual C++ and Developer Studio. Forth programs written using ForthMacs on PC. Multimedia/CD ROM graphics application with extensive 80x86 assembly language programming in a Windows95 environment.

Projects before October 1994

Key to data :

Each project has a description and contains a key as follows :

Date

Company name (approximate length (weeks) , Agency (Ag) or Fixed Price (FP), Forth (F) ,C (C) and/or Assembler (A)).

Projects in Forth will always use the Forth Assembler in part - A indicates a non-Forth assembler.

For example : LYNX project for CDI-players : (15 months, Ag, F&C) indicates 15 months, through an "agency" (in this case MPE) - that is paid daily or hourly, and written in Forth and C). The project time is in weeks unless otherwise stated.