

**PROJECT HISTORY
OF
JOE MAGEE**

6617 Villa Road
Dallas, Texas 75252
214-202-3142 Mobile
online46@starrider.com
<http://www.rapiddl.com>

Note: This document spans many years of experience in Computer Software and Digital Design. I am confident of my abilities in all these areas. My projects include GUI development, middleware, and embedded software/firmware development for PowerPC and Intel 80386EX systems. Projects also include using Visual Studio, linux development using Codewarrior and text editors including makefiles and kernel modules, and Microsoft Windows development using **Microsoft Visual C++ versions 4.X through Visual Studio 2010**, Microsoft Visual C#, Microsoft Foundation Classes (MFC), Borland C++ versions 3.1, 4.0, 4.02, and 4.5, 5.0, and OWL I and II. Other projects include exposure to linux HTTP server applications, Java, Win32, Windows NT, Windows 95, Windows 95 Device Drivers, Windows 7 UMDF drivers, and technologies such as SQL, PHP, CSS, SVN, Agile.

I currently host more than 20 Internet domains using Linux. One domain achieves an average of 1500 hits a day. X86 based computers in my office or at colocation services act as hosts for these sites. This includes HTTP, FTP, and email servers. Additionally, I maintain Domain Name Servers to handle all these web sites.

07/12 to 10/12: **Aristocrat Technologies** (Hourly): Created an emulation tool to allow the firmware based WindowsXP Embedded application for their Slot Machines to operate on a PC in the Microsoft Visual Studio 2008 IDE debugger on Windows 7. This included a UMDF Windows 7 driver, a stubbed dll to emulate communication with hardware not present, and a GUI front end to monitor communication activity with the emulator. The communication layer consists of Named Pipe client and server architecture. Technologies: WindowsXP embedded, UMDF Windows Device Drivers, C++, Named Pipes, multi-threaded, and Microsoft Visual Studio 2008.

04/11 to 05/12: **Hewlett-Packard** (Hourly) (Project 1) On team creating and maintaining software development processes and strategies. Developing software test code using Microsoft Test environment in Visual Studio 2010. Utilizing managed and unmanaged C++ and C# code in .Net environment. Technologies: Visual Studio 2010, SVN, Agile, C++, and C#.

(Project 2) Linux Protected Boot and Virtual Machine Provisioning: Using Python and C/C++ to create utilities and drivers (kernel modules) to manage boot time parameters for a protected and secure virtual machine loader for a laptop. The software provides a way to read and write to the BIOS to allow the reading and setting of various flags and hashes that are stored in the BIOS. The software utilizes custom functions to communicate directly with the BIOS rather than through the /dev interface used after system boot. Python, C/C++, .so files, kernel modules and other methods for communication via the ACPI-WMI interface.

(Project 3) Created a custom Windows Credential Provider to allow silent logon to a Windows NT domain from boot. (C++, Visual Studio 2010).

(Project 4) Modified existing InstallShield project to utilize new Custom Actions to inspect various install caches and take different actions based on the files present or not present in the cache. (InstallShield, C/C++, Visual Studio 2010)

04/10 to 12/10: **Verizon** (Hourly) Add new features to and modify existing code in the Verizon Media Manager Product. C++, MFC, Visual Studio 2005, ClearCase.

04/09 to 07/09: **Soneticom, Inc.** (Hourly) Add new features to existing real-time software. Create a standalone server for accepting data from the real-time software and creating additional utilities for generating and transmitting test data to the server. **Visual C++**, C# .Net, MFC, Sockets.

01/05 to 04/09: **Banc of America, LLC.** (Salary) Maintained and developed new features for a security trading platform. I worked with the Front End (GUI). The GUI primarily accepted market feed data from back room servers and sent user commands (Buy/Sell orders) to the servers for relay to the proper market. Most of my tasks involved accepting market data real time and manipulating the data and then displaying the data to the user in various formats, tables, and graphs. Most of these real time displays provided input fields for users to specify and execute trades. I dealt with real time and static data. The real time data came directly from the various markets. The static data accumulated overnight. The trading platform software originated with Direct Access Financial Corporation. (**Visual C++**/Windows SDK, Visual Source Safe).

02/04 to 07/04: **Fluke Networks** (Hourly) Update existing software (see below) for new base processor. Develop new test tool for PC platform (**Visual C++/MFC**).

The Test and Verification Tool consisted of a Windows Application utilizing a Multiple Document Interface (MDI). Each window emulated a record and playback interface to the Unit Under Test (UUT), a Fluke Networks CopperPro® test set. Communication with each UUT utilized the one serial port.

Each window captured keystrokes from the UUT and allowed playback to make the test repeatable. The window functioned as an editor to allow editing of the captured keystroke stream.

I developed a parser and interpreter in LEX/YACC to allow for local variables, results processing, and conditional statements for program control.

11/03 to 12/03: **Honeywell, Inc.** (Hourly): Design the top level user interface navigation for a Web (browser) based application. Develop demonstration programs illustrating the resulting UI. Browser based interfaces implemented using Java Applets.

10/02 to 9/03: **Jericho Systems Corporation** (Salary): Design and implement a Graphical User Interface (GUI) for an Enterprise Software Product. Act as Project Manager for the product. This is Jericho Systems inaugural and flagship product providing Real Time decisioning for the security space and targeted marketing. All code written in Java with stand alone application capability and browser based capability in a single executable (jar) file. User Interface code utilizes Java Swing. Browser based interfaces use a combination of Java Server Pages (JSP) and Java Applets served from Tomcat.

12/01 to 9/02: **Rapport Technologies** (hourly): Various projects to support the Rapport product. This included custom boot code to initialize certain network resources to allow downloading of boot time executables in the PXE (pre-execution) environment (BSD Kernel, C/C++).

8/00 to 5/01: **Optical Switch Corporation** (Hourly): Projects included design and implementation of embedded software for 80C386EX processor. Design and implementation of various GUI based test and calibration programs for manufacture of Optical Switches. This included implementing a control and monitoring protocol for communication with switch modules. Windows software written using C++ utilizing Microsoft Visual C++, and MFC. Embedded software written using **Borland C++ 4.52** and 5.0 and DOS based Turbo Debugger.

4/00 to 11/01: **Fluke Networks** (Hourly): Projects include software/firmware running on an embedded Power PC platform. Additional projects involved design and implementation of a graphical front end for controlling a portable telephone line tester. Utilizes **Visual C++**, Win32, serial communications, graphic compression/decompression, customized visual control elements, and development of application level protocols. The embedded projects involved Visual SlickEdit, WindRiver compilers and SDS debuggers.

Specific Projects:

Binary Interface: Designed and wrote target device code and PC code to provide command and control of handheld ADSL tester via an incorporated serial port. This included code in C in the target device and **Visual C++/MFC** on the PC. Developed a DLL to provide command API for handheld device. Utilized a previously developed DLL providing simplified serial port interfacing for PC applications.

Stored Results: Developed “streaming” data storage functionality for the target device. The streaming data layer utilizes an existing Flash File system in the target device firmware. C on PowerPC.

Firmware Updater: A PC based application that splits large binary files into blocks and transfers the blocks to the target device using a protocol layered on XMODEM. **Visual C++/MFC**.

Flash Updater: A target device RAM based utility to request and accept blocks of data intended for storage in Flash memory. Utilizes a protocol layered on XMODEM to accomplish data transfer. Utility also erases and programs flash devices. C on PowerPC.

Checksum Embedder: Utility to calculate a checksum on a large binary file and place the calculated checksum at a specific location in the binary file. **Visual C++**.

Screen Dump: A PC based prototype application to request and accept a pixel by pixel dump of the hand held tester lcd screen. The screen data then appears in a window on the PC. Utilized data compression algorithms to enhance data transfer rates. **Visual C++/MFC**.

Target Device Emulator: A PC based emulation of the target device. This provided an interim development environment for both target device code and PC, or external, controller code. The emulation consisted of graphical representations of the target device face including custom buttons and other controls. One intended application includes a future application allowing a “Virtual Instrument” controllable from a PC or laptop. **Visual C++/MFC**.

8/99 to 4/00: **Netier Technologies** (a division of Wyse Corporation) (Salary): Design new and port existing network management software. Platforms include Windows NT, Windows 95/98, Windows 3.1, and linux 5.2 and 6.0. The linux project consisted of a custom HTTP server which allowed management of client side configuration files using an HTML based protocol. Coding done with Microsoft **Visual C++** on Windows platforms. Linux programming done with Codewarrior.

- 10/98 to 8/99: **Abbott Laboratories** (Hourly): Design, code, test, and maintain portions of a Windows NT application controlling a pSOS based Blood Analyzer (AXSym2). The controlling interface utilized RPC and remote mounted hard drives for interdevice communication. Programming done in 32 bit **Visual C++** using MFC and custom controls.
- 5/98 to 10/98: **Netier Technologies** (Hourly): Design and develop a thin client management utility. This program manages thin client computers on local area networks (LAN). Developed proprietary protocol for communication with thin clients. Program allows operators to view hardware and software configurations on thin clients and to initiate various software upgrades or modifications from management console. Program written in 32 bit **Visual C++** using MFC. Operating system is Windows NT 4.0 and 3.51.
- 11/97 to 4/98: **Cytware Corporation** (Hourly): (Ongoing Contract Assignment concurrent with Third Planet Publishing below): Assist in developing security software for Windows 95 platform. Assisted in developing features in a VxD that controls access to disk files and other resources. This includes an Installable File System Manager. Development utilizes Microsoft **Visual C++**, VToolsD, and Visual Sourcesafe. Project also includes developing test suites using Visual C++ and MFC.
- 5/97 to 2/98: **Third Planet Publishing** (Hourly): Design and produce Windows 95 Device Driver for PCI plug in card. Design includes device level interfacing, mapping physical memory to virtual memory, providing application access to mapped virtual memory, and Plug-n-Play. Using Microsoft **Visual C++** and VToolsD. Project also includes several windows based utilities for manipulating PCI devices and accessing, monitoring, and modifying arbitrary physical memory ranges.
- 10/95 to 5/97: **Motorola, Inc.** (Hourly): Design and produce paging system control interface for WindowsNT workstation. Program also being simultaneously ported to UNIX using Wind/U. Operating System Windows NT 3.51, Windows NT 4.0, Windows 95, UNIX (Solaris 2.4). Target 32 bit Windows platform and UNIX workstations. Also using TCP/IP, Windows Sockets, **Visual C++** 4.X, and MFC. The TCP/IP portion of this project consisted primarily of a set of routines written in traditional Berkeley Sockets and the Microsoft **Visual C++** classes. The resulting applications allowed for a mixture of UNIX and Windows servers and clients to transfer data over specified ports. The designs included multi-threaded design so that any particular server managed multiple clients. I developed the application level protocols.
- 3/95 to 10/95: **Motorola, Inc.** (Hourly): Design and produce paging system control program using Win32 on Windows NT. Operating System: Windows NT 3.51. Target

Windows NT Server. Also using TCP/IP-Windows Sockets, **Visual C++ 4.X**, and MFC.

11/91 to 3/95: **AVO International** (Salary): Software Engineering

PROJECTS

6/93 to 12/94: Designed and produced a Microsoft Windows application using Borland C++ and OWL I and OWL II. The program provides a user interface to an external test set used to test watt-hour meters. Project included conversion from OWL I to OWL II when OWL II became available. **Borland C++ 1.5 years**, OWL 1.5 years.

11/91 to 6/93: Adapted existing DOS program to Microsoft Windows. Used Borland C++ and OWL I. I internationalized the program by insuring that all text presented to the user resided in the resource file. **Borland C++ 1.5 years**, OWL 1.5 years.

6/83 to 11/91: **Self Employed** (Hourly). Major clients included COMPAQ Computer Corporation, Honeywell, Incorporated, and Multi-Amp Corporation (now AVO International).

Note: During this eight year period of self employment, I used Pascal, C, C++, Z-80 assembler and several other specialized languages. I designed digital circuits and systems. I designed and produced embedded software using C, Pascal, and various assembly languages. Exact accumulations of times are difficult to specify because many projects were undertaken in parallel or piecemeal.

PROJECTS

4Q91: **TankMasters** (Speculation): Produced a demonstration program using Turbo Pascal for Windows for a proposed system to monitor the fluid level in remote storage tanks. Used OWL I for Pascal.

3Q91: **Honeywell, Corporation** (Hourly): Wrote new, and modified existing, programs in HVAC controllers at the Phoenician Hotel and Resort in Phoenix, Arizona. These programs are written in *Honeywell Pascal*, a specialized language for Honeywell's Excel controllers.

Red River Resources (Hourly): Selected and installed computer hardware and software for the management of tax roll databases of the owners of oil leases in various counties in Texas. Wrote several programs in dBASE language for the generation of text files used to merge into form letters. Wrote several macros in Word Perfect to accomplish form letter merging.

MindShare, Incorporated (Fixed Price): Presented an eight day course to a client in Mexico City on the *hands on* trouble shooting of 80386SX based PS/2 compatible mother boards.

2Q91: **Multi-Amp Corporation** (Hourly): Designed and produced a software system to run on a PC compatible computer to interface to a motor tester and organize the motor data collected. The user interface is designed using C-scape. The program included the interface to the tester, complete database maintenance, and the ability to graphically compare the results of multiple tests. The entire system is written in **Borland C++** Version 3.1.

1Q91: **Biological Products, Incorporated** (Fixed Price): Designed a Z80 based pulser unit. The unit is designed to interrupt the circuit connected to it in various timed patterns. These patterns range from fixed rates to ramped rates. The duty cycle of the pulse is also variable. The operator controls these patterns via front panel rotary switches. I designed and built the entire hardware and software system. The firmware is written in C.

Cochran Consulting, Incorporated (Hourly): Reverse engineered the firmware in an 8051 based microcontroller. The 8051 is used in the keyboard of a PC/AT compatible computer. Execution of this project required an in-depth understanding of 8051 assembly language.

Honeywell, Incorporated (Hourly): Designed a program to manage the output of data files to multiple plotters. The program runs in a PC connected to a network. Any time a user on the network sends a plotter file to the PC, the software detects the file and sends it to a specified plotter. The objective of the program is to allow any number of users to send plotter files over the network at the same time, but to send only one at a time to a plotter. The program is written in Turbo Pascal Version 6.0 using Object Oriented Programming (OOP) techniques. The user interface is Borland Turbo Vision. Additionally, standard commercial serial port interface libraries are used.

Thoma Construction Company (Hourly): Designed and produced a program to record financial transactions of construction company. Used the following: **Borland C++** through version 4.51, Codebase to interface to Xbase database, Borland Turbo Vision, Borland PowerPak for DOS (including DOS extender), OOP programming techniques. Cumulative time on project approximately nine months through 10/96.

4Q90: **Honeywell, Incorporated** (Hourly): Refined and expanded software to convert information from Honeywell internal database format to dBASEIII format. The new programs convert all information in the Honeywell databases, not just the Fire Alarm related data. The program is written in Turbo Pascal Version 5.5.

Honeywell, Incorporated (Hourly): Assisted in the installation of the Fire Alarm system in the Barry M. Goldwater terminal of the Phoenix Skyharbor Airport. This effort included the creation of a dBASEIII database to track the configuration of each monitor point. The dBASE files also tracked all the internal programming logic for the entire system. (This means the logic used to determine what type of action is to occur when any particular point goes into alarm.) I produced programs that could convert the Honeywell databases into dBASE format. This included dBASE programming, a computer language called AWK, and using the BRIEF programming editor.

3Q90: **Honeywell, Incorporated** (Hourly): Modified existing control software to allow the monitoring of additional data points. The software was written in 80X86 assembler language and run under Concurrent DOS.

Honeywell, Incorporated (Hourly): Modified the firmware in an obsolete product line to allow new operational modes. The product was a damper controller for airflow in large Heating, Ventilating, and Air Conditioning (HVAC) systems.

Specialty Instruments, Corporation (Hourly): Developed an application for a PC/AT compatible computer that monitored light beam units spaced at 1/16 mile intervals around a thoroughbred horse racing track. The system automatically configured itself for the type of race being run and as the horses progressed around the track, it displayed the running time for the event on a scoreboard. The running time was divided into up to five display lines. One for approximately each 1/4 mile portion of the race. The system interfaced to the light beam units through an off the shelf multi-channel parallel adapter. The scoreboard interface was RS232. The entire program was written in Borland Turbo C++ version 1.0 using object oriented programming (OOP) techniques.

2Q90: **MindShare, Incorporated** (Fixed Price): Developed a course that is used by a major manufacturer of personal computer systems. The course teaches manufacturing technicians how to use a specific piece of commercial test equipment to test 80386SX based microprocessor boards (microchannel PC architecture). The course includes a discussion of 80386SX concepts, trouble shooting concepts, and the operation of the specific test equipment. I also teach the course. It requires five classroom days. Developing the course required strong knowledge of the 80386 and microcomputer fundamentals. Ventura Publisher was used to produce the course materials. The course book is approximately 200 pages long.

1Q90: **Cochran Consulting, Incorporated** (Hourly): Cochran Consulting designed a telemetry system used to monitor various parameters pertaining to a scuba diver while the diver is in the water. This information is transmitted to a surface

computer that displays information about all the divers in the water, and each individual diver has a wrist mounted display that shows his parametric information. I designed the control program for the microcomputer in the wrist unit. I also designed the first release of the surface computer software. This software interfaced to a proprietary transceiver that communicated with the monitoring computer on each diver. It gathered information about each diver, stored it in active memory, and formatted and displayed the data on the computer screen per operator specified ordering. It also displayed, in real time, the polling of each underwater unit. The software was written in Microsoft QuickBASIC, 80X86 assembler language (for the interrupt service routine that interfaced to the transceiver), Z80 assembly language (for the portion of the underwater firmware that interfaced to the diver mounted communications hardware), and in the assembly language used by the four bit microcomputer used in the wrist unit.

4Q89: **Honeywell, Incorporated** (Hourly): Produced a program for a PC that lets the PC emulate an Intecolor 8001G color graphics terminal. This program was written in C to run under Concurrent DOS and uses the multitasking capability of Concurrent DOS.

3Q89: **Specialty Instruments Corporation (SIC)** (Hourly): Developed firmware for Z80 based controller used as a Velodrome Event Timing System. The program was written in Z80 assembly language.

Boys Clubs of America (BCA) (Hourly): Developed a complete system in Paradox Application Language (PAL) to allow the monitoring of the entry and exit of individuals from BCA facilities. Paradox is a database system marketed by Borland International. The system produced several reports to characterize the utilization of the facilities over time. The primary input to the system was via a bar code reader used on the membership cards as the individual entered and left the premises.

2Q89: **Tektronix, Incorporated** (Hourly): Wrote an instruction manual for the CAT200 computer program. This is a program that runs on a PC and interfaces to a digital oscilloscope to allow the PC to become a remote display for the oscilloscope. The manual is approximately 15,000 words long.

1Q89: **COMPAQ Computer Corporation** (Hourly): Produced an instruction manual for (and taught) a seminar on the structure of OS/2. Course development included work with Ventura Publisher and development of simple application programs in Microsoft C for the OS/2 protected mode and compatibility box environments.

4Q88: **Specialty Instruments Corporation** (Hourly): Produced and modified software and hardware for existing and new Track and Field Scoreboards. Work

involved assembly language and C programming for the Z80. In addition, an extensive user interface program was produced in C for a PC.

3Q88: **Honeywell, Incorporated** (Hourly): Added additional enhancements to real time graphic presentation software. Project involved on site work in Kenai, Alaska.

2Q88: **Honeywell, Incorporated** (Hourly): Modified existing software to provide additional real time data to graphic representation of an environmental control system. The language used was C and 8086 assembler. The operating system was Concurrent DOS XM 6.X running on an IBM PC/AT.

Specialty Instruments Corporation (SIC) (Hourly): Wrote firmware for a Tennis Match Timer. This project included a basic control program for an SIC standard CPU card and incandescent lamp panel display. The basic program has a software interface for an application that can then compose data for display. The software interface can accept inputs from an external serial link or from other software routines that can be linked prior to programming into EPROM. The hardware is based on a Z80 and the software was written in C and Z80 assembler.

1Q88: **Honeywell, Incorporated** (Hourly): Wrote communications protocol emulator to supply controllable data to a hardware communications interface. This was to test for the existence of a problem in the hardware communications interface firmware. The interface was used in a Hotel Property Management System. The program was written in C on an IBM PC and used interrupt driven serial I/O.

Specialty Instruments Corporation (SIC) (Hourly): Modified an existing hardware circuit to better detect the sound of a gun blast at the start of a track/field event. The circuit is a simple voltage comparator detecting activity on the output of an ac coupled microphone.

4Q87: **Specialty Instruments Corporation (SIC)** (Hourly): Modified a previously written user interface program. This program allows a user to compose messages to be displayed on an outdoor traveling message sign. The modifications were to change the program to a different C compiler. The new C compiler utilizes direct CRT I/O which speeds up the menus considerably. The host computer is an IBM PC.

Honeywell, Incorporated (Hourly): Designed a STD-BUS based automatic test set to test a relay isolation card. The test set employed an a-to-d converter. The test program was written in C for a Z80 and burned into EPROM.

SIC (Hourly): Modified the firmware in a track and field scoreboard to accept inputs from a set of thumbwheels and pushbuttons to allow manual display of scores. The program was written in Pascal.

Honeywell, Incorporated (Hourly): Manufactured fiber optic based voltage surge protectors. These devices are intended to protect computer equipment from voltage surges that occur on outdoor RS422 data cables. The voltage surges are produced by nearby lightning activity.

3Q87: **SIC** (Hourly): Designed electronic hardware and software for a strobe light activated photo finish camera. The camera was required to distinguish the flash of the starting strobe light from random strobe flashes so that a notation could be placed on the finish photographs that the event timer was started from direct detection of the gun blast rather than some random photographer's flash. The camera firmware accomplished this by responding to a known sequence of flashes produced by the gun shot detection hardware. The program is written in Z80 assembler. Also designed the strobe light flash pattern generator and strobe light detector. The detector can detect the flash of a strobe light in daylight to a distance in excess of 700 feet. It can discriminate against most other light flashes such as swinging the detector around or the passage of a reflecting surface through its field of view.

Honeywell, Incorporated (Hourly): Designed a relay signal isolation circuit. This circuit was intended to remedy problems associated with switching small currents with a high current relay. This circuit is used in the New Mexico State Penitentiary in Santa Fe. Additional details of this project can only be released by permission of Honeywell, Incorporated.

2Q87: **SIC** (Hourly): Modified existing firmware used in a Track and Field scoreboard to accommodate a new display matrix. The firmware is written in Pascal and burned into an EPROM. The microprocessor is a Z80.

1Q87: **SIC** (Hourly): Designed firmware for a finish line computer that monitors operator input and finish line light beam inputs to provide split times for each runner in a running event. The information is collected and printed on a printer and finish time data is supplied to a spectator scoreboard. The firmware is written in C for a Z80.

SIC (Hourly): Modified a program that accesses a database of track and field event data to allow the program to support the transmission of data to a SIC designed track and field scoreboard. The program was supplied to SIC in IBM PC BASIC source code.

SIC (Hourly): Added traveling message capability to existing firmware for operation of a track and field scoreboard. This was the marriage of two previously unrelated projects. The firmware is written in Pascal for a Z80 and burned into EPROM.

SIC (Hourly): Modified an existing program to allow the composition of messages to appear on an outdoor traveling message display. The program runs on an IBM PC and is written in C.

Honeywell, Incorporated (Hourly): Modified existing software in security control computers installed in New Mexico State Penitentiary in Santa Fe. The software is written in Pascal and runs in a S-100 computer using CP/M-86.

SIC (Hourly): Developed hardware and software to print an operator specified message on film traveling through a photofinish camera. The program is written in 8080 assembler and runs in a TRS80 Model 100 portable computer.

SIC (Hourly): Developed software to monitor the output of two infrared detectors to calculate the velocity of an object that is sequentially detected by both beams. The program is written in C for a Z80 and is burned into EPROM.

Honeywell, Incorporated (Hourly): Developed a fiber optic based circuit to isolate computer equipment from voltage surges occurring on outside RS422 data lines.

Honeywell, Incorporated (Hourly): Developed software to customer specification to produce detailed reports of energy consumption in a large installation. The installation was being controlled by a Honeywell D5200 Energy Management System. The software was written in D5200 Report Writer and runs on a Honeywell D5200. The central computer in a D5200 is a Honeywell DPS6.

Honeywell, Incorporated (Hourly): Developed firmware for a Z80 based single board microcomputer that connected two half duplex radio modems to a half duplex interface to Honeywell environmental control computers.

Honeywell, Incorporated (Hourly): Modified existing software in security control computers installed in New Mexico State Penitentiary in Santa Fe. The software is written in Pascal and runs in a S-100 computer using CP/M-86.

4Q86: **SIC (Hourly):** Modified existing firmware in a track and field scoreboard. The software is in Z80 Pascal and burned into EPROM.

SIC (Hourly): Developed firmware for a timing computer for use at Rodeo Events. The firmware responds to operator inputs and finish line light beam

inputs. It calculates median and average times for up to three timers. The software is written in Z80 assembler and burned into EPROM.

3Q86: **Honeywell, Incorporated** (Hourly): Modified existing software in security control computers installed in New Mexico State Penitentiary in Santa Fe. The software is written in Pascal and runs in a S-100 computer using CP/M-86.

Honeywell, Incorporated (Hourly): Developed software to graphically display accumulated data in an environmental control computer. The software allowed operator specification of monitor and control points of interest and time scales. The collected data could be converted into DIF format for importation into spreadsheet programs such as Lotus 1-2-3. The software was written in C and ran on a CP/M-86 computer. The graphics terminal was a Tektronix 4105.

2Q86: **SIC** (Hourly): Developed operator interface program for traveling message outdoor display. Software written in BASIC and runs on a TRS80 Model 100.

Honeywell, Incorporated (Hourly): Developed a user interface to allow an operator to organize the data and control points in an environmental control computer into groups. This allowed the operator to specify common actions to an otherwise unrelated group of points. The program was written in C and runs on a S-100 based CP/M-86 computer.

1Q86: **SIC** (Hourly): Developed user interface to allow composition of messages for outdoor traveling message sign. The software was written in C and ran on a TRS80 Model 100 portable computer. This required the development of a run time library for the Model 100 computer. This program was later converted to CP/M-80 and eventually to IBM PC-DOS.

4Q85: **SIC** (Hourly): Developed software to take a data stream from an IBM System 36 and convert it into data files for use by a cutting horse event scoreboard control computer. The software was written in Pascal.

Honeywell, Incorporated (Hourly): Developed hardware and software to drive a pulse generator for the checkout of pulse driven environmental condition monitors. The hardware was connected to the printer port of a Kaypro 2/4 computer and the software was written in Pascal.

3Q85: **SIC** (Hourly): Developed hardware to expand the capability of microcomputer controlling an outdoor traveling message sign. The new hardware had 64K of static ram, a real time clock, and signal distribution amplifiers. The static RAM was configured as an addressable peripheral device and is used for storing message text generated by an external message composition device.

SIC (Hourly): Developed firmware to measure the time between the interruption of two light beams. This time was then used to calculate the velocity of object that broke the beams. The speed was then displayed on an external display. If the speed was greater than any previous display, it was declared the current maximum and a signaling device was activated. The system was installed at a slide at a water park. The software was written in Z80 assembler and burned into EPROM.

2Q85: **SIC** (Hourly): Developed firmware for an outdoor traveling message sign. The sign allowed the display of externally generated message text. Several display modes were supported including static centered, traveling, scrolling, and rolling. The firmware is written in Pascal for the Z80 and burned into EPROM.

SIC (Hourly): Designed hardware and software for a system of lights that were to be placed around an indoor jogging track. The purpose was to allow a person to specify a rate to the system in minutes per mile and then run at that pace. The system would sequentially illuminate a colored lamp on the panels placed around the track. If the runner insured that he was in the same relative position to each panel as the light came on, then he would be pacing the specified minutes per mile. The software is written in Z80 Pascal and burned into EPROM.

1Q85: **SIC** (Hourly): Developed the hardware and software for a photo finish camera that received data to start the event timer over a radio link. The purpose of this setup was to allow the event starter to be able to move around without trailing a cable that went to the camera and timer. This was handy because the camera and timer could be as far as several thousand feet away. The radio link detected the gunshot. Upon gunshot detection it transmitted a sequence of 64 data frames. Each frame was ten bits long and composed of a sequence of synchronization and data pulses. The length of each frame was 10ms. The firmware in the camera monitored a radio receiver and decoded the bits in each frame. Upon successfully decoding two sequential frames of data the camera firmware could then start the event timer and set the event clock to the actual time that had elapsed since the gunshot was detected. The software was written in Z80 assembler and burned into EPROM.

4Q84: **Webb and Associates** (Hourly): Developed prototype circuit to monitor daylight intensities to assist an external circuit in the control of window screens to help regulate sunlight entering a room.

3Q84: **SIC** (Hourly): Developed firmware to monitor a light beam to start a timer to time a bicycle event. At subsequent light beam interruptions, the elapsed time was converted into a text string and transmitted serially to an external monitor computer. The software was written in Z80 assembler and burned into EPROM.

SIC (Hourly): Developed firmware for a Z80 based horse race timing scoreboard. The times were supplied as interruptions of a light beam. As each successive beam was broken, the elapsed time was converted into the format used in horse races and placed in a spectator display.

2Q84: **SIC (Hourly):** Developed software to display data on a scoreboard. The scoreboard was composed of an array of boards that had triacs that illuminated indicated incandescent bulbs. The software accepted data from serial I/O channels and formatted it for display.

Honeywell, Incorporated (Hourly): Began work on software to interface multiple operator positions to security control computers at the New Mexico State Penitentiary in Santa Fe. This is a very complex program, and support and modifications continued until at least 1991. I used Pascal for CP/M-86 as the language. Additional details of this program can be supplied by permission of Honeywell, Incorporated.

3Q83: **Bright Star, Incorporated (Hourly):** Began a seven month contract to supply programming services to E-Systems. The programs developed were used to test various parts of a Tandem Computer system while the system remained on-line. The programs were written in TAL. During the course of the contract, E-Systems sent me to a TAL programming course.

7/81 to 6/83: **Docutel, Incorporated (Salary):** Test Engineering.

PROJECTS

Developed firmware for standalone testers used on the production lines of microcomputer controlled automatic teller machines. Various tests were produced for the different controllers in the machine. The controllers and testers were Z80 based. The software was in Z80 assembler and burned into EPROM. The software was developed on a VAX using the UNIX operating system.

5/77 to 7/81: **Texas Instruments, Incorporated (Salary):** Test Engineering and Test Engineering Management.

PROJECTS

I managed the test equipment group developing test equipment for the Texas Instruments CB Radio. The testers were based on microcomputers using the TMS9900 chip.

Designed test equipment to test various calculator products in the TI consumer product division. These testers were based variously on the 9900 and TI960 computer.

3/77 to 4/77: **Association of Legal CB'ers** (Salary): Technical Staff

PROJECTS

This company was formed by a friend to attempt to ride the popularity of the CB radio hobby of the day. The goal was to provide technical information about the hobby and the FCC. An additional goal was to provide a forum to discuss what was legal operation and offer guidance to achieve legal operation and why this was or was not desirable. My duties were to provide technical information by way of reports or other written communication, and to provide any other type of technical information needed by the staff or members. I also piloted the company airplane. The company went out of business within weeks of my joining the staff as the CB radio craze had already crested and people did not join in sufficient numbers.

3/76 to 3/77: **Motorola, Incorporated** (Salary): Reliability Test Engineering.

DUTIES

I tested or ordered the testing of CMOS devices in the CMOS logic product line of Motorola, Incorporated. I prepared reports from the test data received. These tests included burn-in, and temperature and mechanical stress testing.

8/75 to 6/76: **MRI** (Salary): IBM ALC Programming.

PROJECTS

Developed software to dynamically allocate additional disk space to a parent process. The operating system allocated a process a fixed maximum of disk resources. If a space error occurred (the process used all its space and attempted to get more), the process was terminated. The utility I wrote trapped the error and manipulated the system resources to allocate additional disk space.

4/75 to 8/75: **Collimation, Incorporated** (Salary): Test Engineering and Design

DUTIES

Supported the manufacture of optically encoded ASCII keyboards.

8/74 to 2/75: **Rockwell International, Incorporated** (Salary): Software specification and technical writing on the Space Shuttle Program.

2/74 to 8/74: **KTVV TV-36**, Austin (Salary): Engineering Staff.

6/72 to 2/74: **Self Employed** (Commission Sales).

6/69 to 6/72: **Texas Instruments, Incorporated** (Salary): Test Engineering.

ADDRESS:
6617 Villa Road
Dallas, Texas 75252
214-202-3142
online46@starrider.com

References available on request.