c-horsesoftware.com

Real-world, usable engineering

* *RIP software development for all industries*
* *Color control/conversion/separation.*
* *Custom software for just about any purpose.*
* *IoT solutions.*
* *Custom circuit boards*
* *Purpose built small microcontroller designs.*
* *Windows, Windows CE and embedded programming*
* *Motion control. Steppers / Servos*

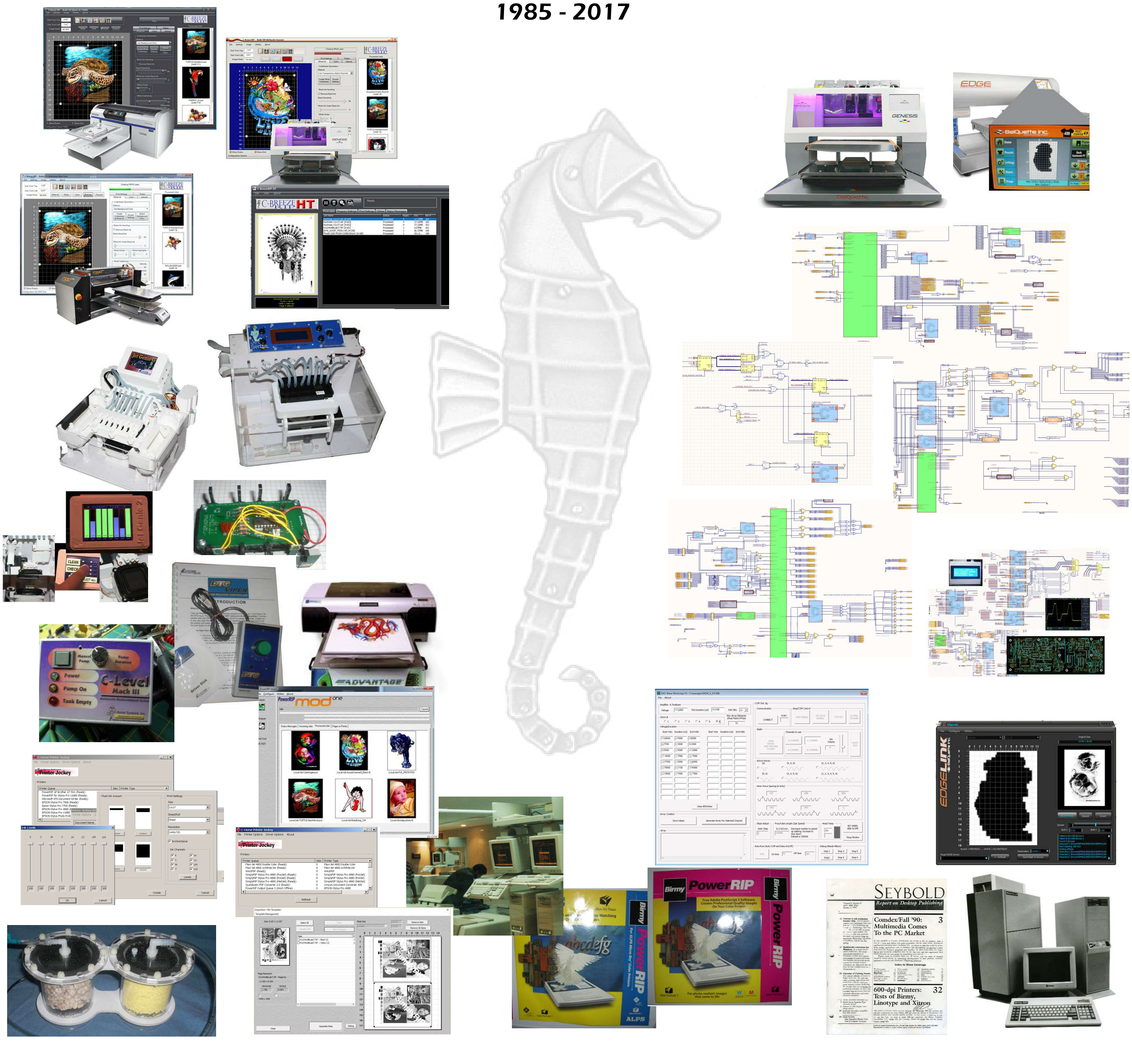
C, C++, Embedded C/C++, Assembly, PostScript, HTML, XML, FPGA development, PLC, EPLC, ASIC, Motion Control, Color Profiling, Color Separation, Color Conversion, Halftoning, Error Diffusion/Stochastic Screening and more.

Software and hardware design solutions from concept to finished product and all steps in between. We deliver quality, innovative and intuitive products that work. Setting industry standards since 1985. Industry leading direct-to-garment RIP development since 2005.

PROJECT PORTFOLIO

Version 3.0.1

1985 - 2017



Preface

Over 100,000 users worldwide. From the dawn of Desktop Publishing to present day. Code and design that has helped shape the pre-press, proofing and direct-to-garment industries.

As head programmer for Birmy Graphics (which later became iProof Systems, Inc.) I have designed and written PostScript RIP implementations and printer utilities for over 30 years. Driving all kinds of output devices. From laser imagesetters that sell for hudreds of thousands to ninety dollar inkjets and everything in between. All with unparalleled conectivity and ease of use. Many of the innovations I designed into those products have become household words in the RIP business. In 1998 I founded C-Horse Software, Inc. Under the C-Horse name I continued to write RIP implementations as well as other utilities. I also developed several hardware and firmware products.

During the last several years, I worked closely with another engineering firm (Belquette, Inc.) on the development of the a fabric pretreater and a direct-to-garment an inkjet printer. The Genesis direct-to-garment printer was developed and built completely from the ground up, using in-house designed hardware exclusively. The only component which was not engineered by our development team was the actual print head which is a Ricoh Gen 5. I wrote all the firmware, software and FPGA fabric for the Genesis. Everything from the RIP, all the way through the system to the FPGA logic which vibrates the piezo crystals. This includes motion control for the X (carriage), Y (platen advance) and Z (platen height). Also the software/FPGA which controls all the pumps, valves, switches. Touch scree, IoT, etc.

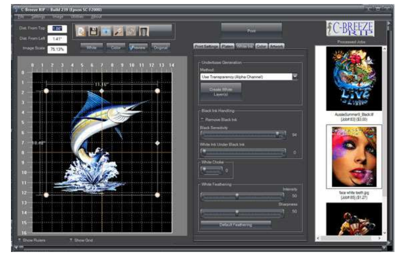
As a side-effect of the Genesis project, emerged a line of RIPs named the C-Breeze. This RIP is now available for the Epson SC-F2000 as well and has been well received.

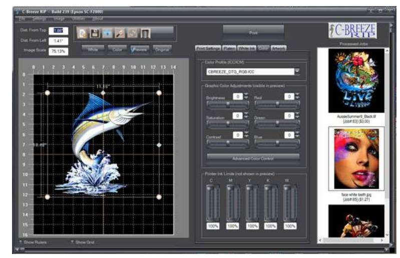
This portfolio lists projects in reverse chronological order. Not all projects are listed and not all projects are described in detail as the disciplines used are either the same or similar enough that they don't illustrate any new or unique skill set that hasn't already been presented.

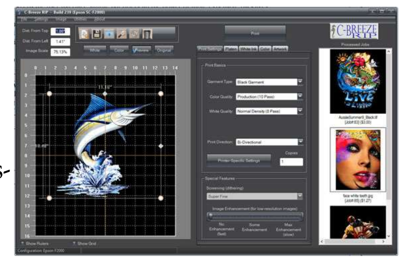
Thank you for your time. I hope C-Horse can offer development services to your organization.

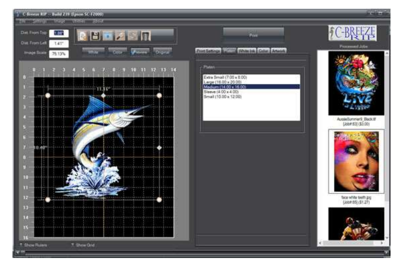
Fred Padilla
President
C-Horse Software, Inc.

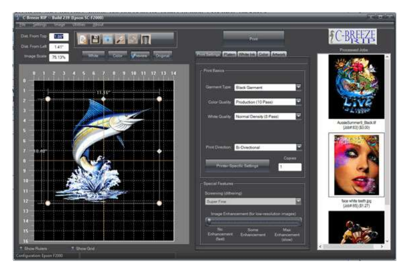
C-Breeze RIP Series

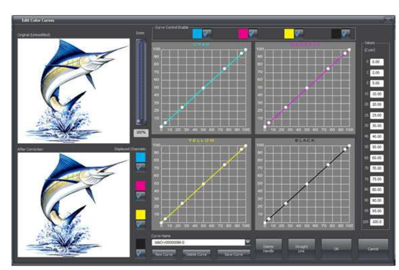












C-Breeze RIP is the culmination of over 12 years of direct to garment RIP development experience. A lot of RIP vendors have taken their existing product, initially designed for other uses, and added the ability to produce direct-to-garment jobs. C-Horse was designed from the ground up to be a direct-to-garment RIP. No unnecessary features to trip over. It keeps the user interface simple but offers all the power of more expensive packages. Arguably the best white-ink generation in the industry.

Written in C++ using Microsoft Foundation Classes, the program has a modern look and a small footprint. It does not depend on large memory-hogging frameworks like .net. The graphics handling is fast and the color is true. The copy protection is achieved with a web-based validation system.

For more info, see www.c-breezerip.com

Intuitive User Interface

Flexibility and power should not mean complexity. C-Breeze has an intuitive user interface that's easy to learn and navigate.

Workflow systems support and other unique features

Although easy to use, C-Breeze RIP includes support for high-production workflow systems. This makes it ideal for unattended or "headless" operation. Cafe Press is using this system both on C-Breeze RIP and on C-Breeze RIP's predecessor, PowerRIP (sold for years by iProof Systems). It also includes specialty features like special effects and the ability to export the individual layers.

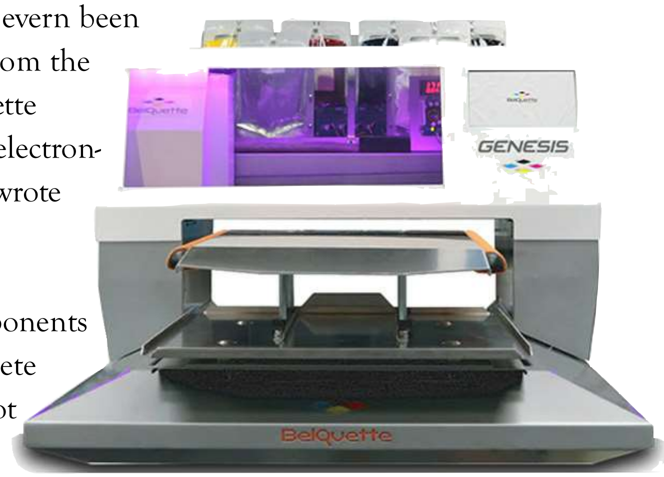
C-Horse Software Inc. is a licensed Epson ISV and therefore has access to the Epson development libraries. These ensure complete compatibility and quality of output. The F2000 version uses the Epson HTM (halftone module libraries).

The Genesis version, however, uses an entiredly home-grown language called BelOse which I designed from scratch for the Genesis. It supports both hardware and software weave (weaving done by the printer's hardware or the RIP). BelOse language specification is on a separate document.

Unlike printer languages typically developed by the "smaller" engineering forms, BelOse is a full-featured language, closely resembling languages from large corporations such as Brother, HP and Epson.

The Genesis

By far the most challenging and educational project I've evern been involved in. This printer was designed and built from scratch, from the ground up by Belquette, Inc. and C-Horse Software, Inc. Belquette (specifically Mark Mombourquette) designed the hardware and electronics while C-Horse Software, Inc. (specifically Me) designed and wrote all the firmware, software and FPGA fabric.



This portfolio explains the various subsystems and components of the Genesis in order to illustrate the skillset needed to complete the project. However, please understand that some details cannot be divulged due to a non disclosure agreement that is still in-place between C-Horse Software, Inc. and Belquette, Inc.

Printer Particulars

The Genesis employs two Ricoh Gen 5 print heads. One 4 color and one 2 color. They both have the same number of nozzles. The 4-color head is used for 4-color the CMYK layer, the 2-color is used for the white. The printer and the platen move simultaneously in a reciprocating motion. This allows maximum print area with minimum footprint. One of the main printer components is a very small single-board PC running Windows embedded. This allows for a very sophisticated touch-screen driven user interface. It also allows it to be easily networked in nearly any way imaginable.

Subsystems

The Genesis has many subsystems. This section briefly describes the function of each subsystem and whatever other details are necessary to illustrate the complexity of the work performed by C-Horse Software.

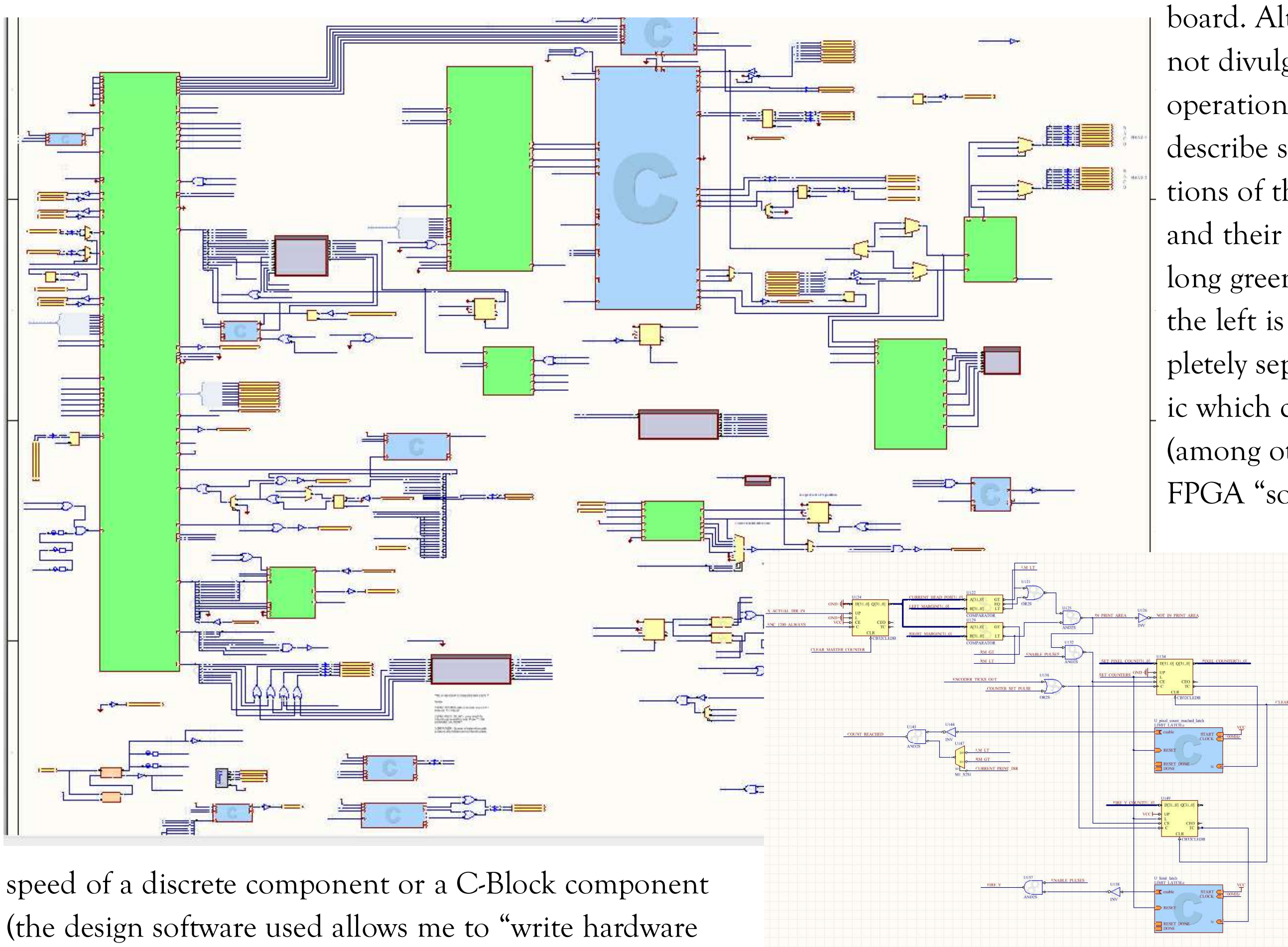
The Genesis (continued)

FPGA Board Set

There are three circuit board components in the Genesis with on-board FPGA devices. Each has an area of control they are responsible for. One handles the pixel pipeline, communications with the onboard controller PC, the X motion (moving the carriage) and the Y motion (platen movement). One handles all the I/O between the main FPGA and the Ricoh Gen 5 heads. The last one handles all the peripheral motion such as capping station, platen height, ink valves, sensors and drying lamps. The 3 circuit boards were designed and created by Mark M. at Belquette, Inc. All the FPGA fabric was designed and created by Fred Padilla, C-Horse Software. Obviously there was a lot of discussion between the two as to function and form which influenced both sides of the design.

The main FPGA

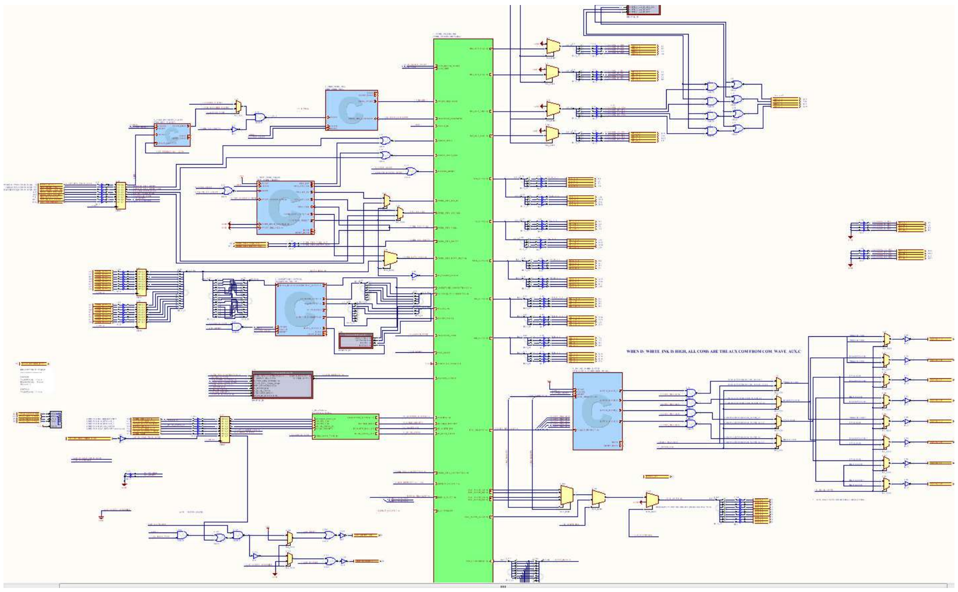
This component handles all carriage movement, pixel alignment, and platen Y movement. It also orchestrates the action of the other two FPGA devices. Below is the top-level schematic I created for the main FPGA circuit board. Although I cannot divulge too many operational details, I will describe some of the sections of the schematic and their purpose. The long green rectangle on the left is actually a completely separate schematic which contains (among other things) an FPGA "soft processor". This soft processor, similar to an 851, handles all the mundane tasks which do not require the speed of a discrete component or a C-Block component (the design software used allows me to "write hardware devices in C"). Towards the middle-bottom, is the circuitry which controls when the pixels actually fire (the actual pixel sequence actions are explained later). Basicall, when all signals are in a certain state, whenever the head is inside the "window", each encoder pulse triggers the pixel sequencer section which sends the necessary digital and analog signals to the Gen 5 to fire a row of pixels. The signal to initiage a platen advance is also triggered from this logic block.



The Genesis (continued)

The Multiplexer FPGA

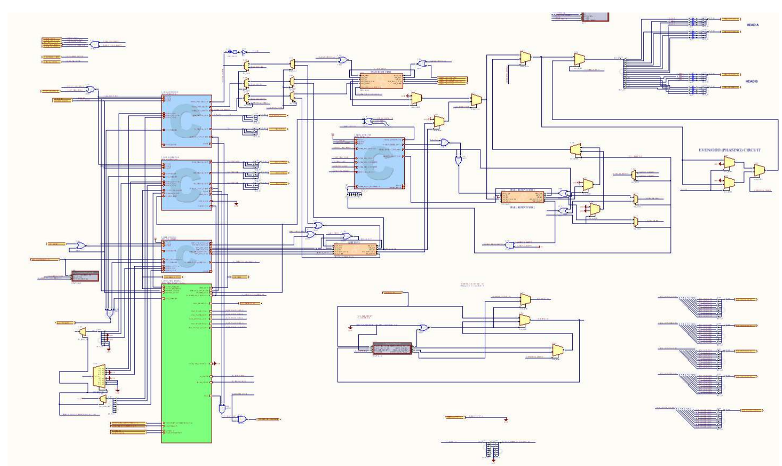
Top-Level circuitry. This component handles the generation of signals for the print head. The Gen 5 requires several digital and analog signals to fire pixels. This FPGA component generates all the signals necessary for this to occur. It also generates the nozzle check pattern data and the on-demand ink-spit data (spits selected channels while cleaning). There are also some blocks (gray in color) which are used for debugging.



One of my favorite features in the FPGA design package used for the Genesis is the ability to define hardware logic components using the C programming language. As a long-time C programmer, this cut development time considerably. The blue blocks with a “C” in the middle represent these hardware logic components written in C.

The large green rectangle in the middle represents the “pixel sequence generator” schematic. That circuit controls all the other pixel-generator circuits such as the different digital signal generators and the analog (COM) signal generators, the pixel data FIFO circuitry, etc.

The Multiplexer FPGA

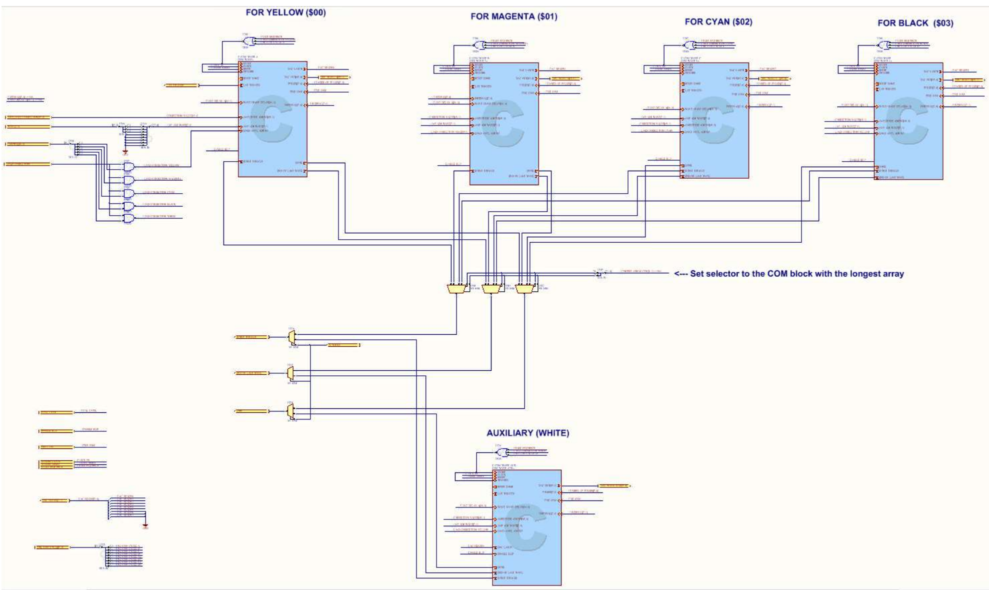


The Pixel Sequence genrator Circuit. The pixel sequence generator orchestrates all the different logic components included within, which generate all the necessary digital and analog signals necessary for the firing of a column of pixels. All the appropriate pixels fire at once throughout the entire head. At this point, all the pixel data has been formatted properly. All the weaving, etc. has been performed. This happens very, very rapidly so there is no time for formatting at this point. The details of how this is done I cannot divulge without an NDA. This circuit also includes a subsystem which can automatically repeat the pixel columns to achieve double the density. This is now achieved in different ways but the mechanism to repeat the columns is still present. The pixel data chain also doubles as a databus in certain modes, such as selecting which channels to spit during a cleaning. This mechanism also contains the ability to alter the amplitude of the COM waves (the pulses that actually vibrate the piezo elements) without changing their shape. This allows us to accomodate different head ranks.

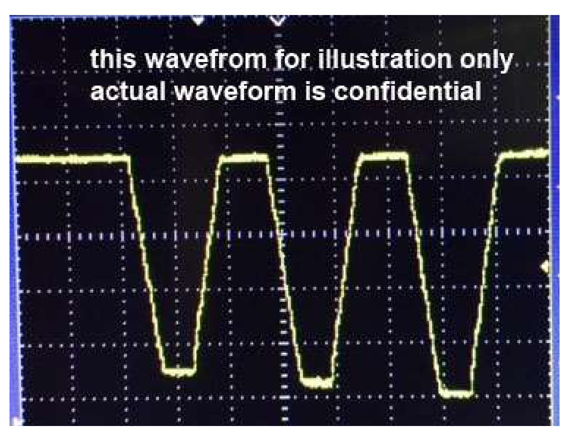
The Genesis (continued)

The Multiplexer FPGA (continued)

The Pixel Sequence genrator Circuit (continued). One of the more challenging aspects of creating circuitry which succesfully drives the Ricoh Gen 5 print head is the shape of the COM waveform. The COM is a high-voltage, high-amplitude analog signal which is used to pulse the actual piezo crystals in the print head. This pulsing is what causes the ink to eject from the nozzles. The COM signal must be closely synchronized with an additional set of digital signals which define excatly “what” will happen when the COM wave fires.



The documentation for the print head defines the maximum and minimum timings for the COM wave. It also depicts the “suggested” waveform (which is formed by using the mean of these limits). This default wave is a simple inverted triangular wave. This waveform must be changed significantly, however, depending on the ink being used. After many months of tuning, the actual waveform used by the Genesis looks nothing like the default waveform described in the head documentation. Ricoh does offer a waveform editor system to its OEMs. However, while their software allows for the tuning of triangular waveforms, it is lacking the ability to create composite waveforms. So, I also developed the software tool which, used in conjunction with the high-speed camera included in the Ricoh tool, allows very fine control over the shape of a composite COM wave. This software is mentioned again later in this porfolio. Although I wrote the tool for waveform editing, the actual shape ultimately used by the Genesis was developed over many months by Mark M. and Brett W. of Belquette.



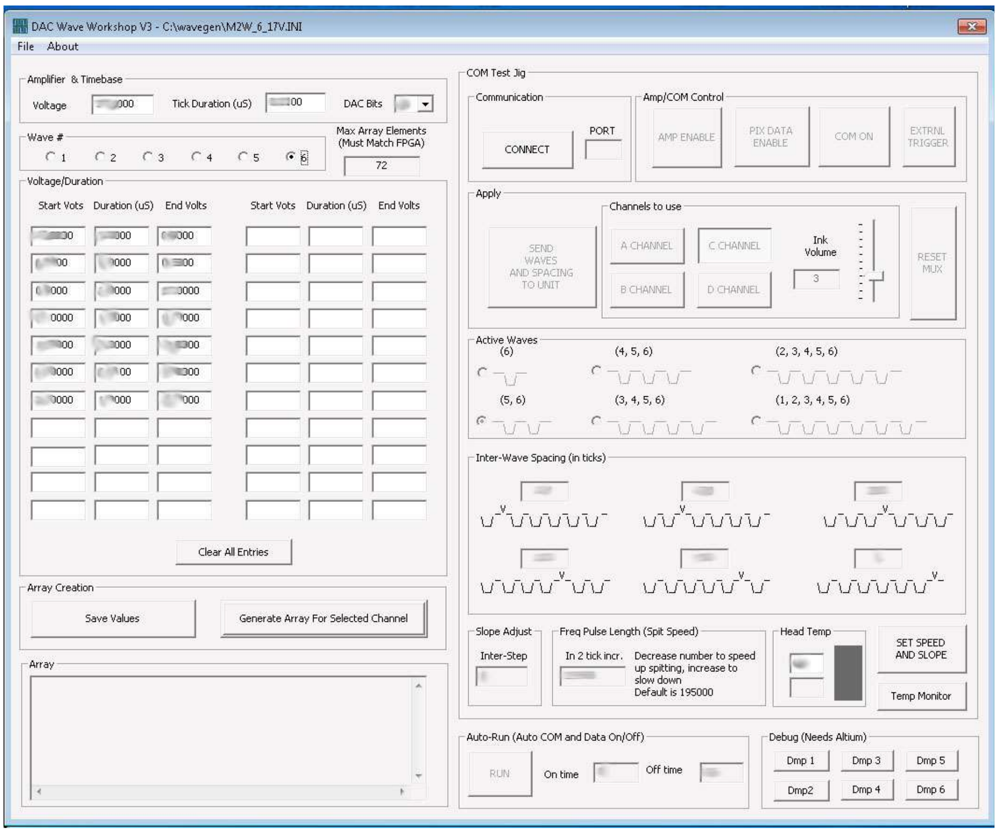
Most of the “magic” part of the COM generator system is inside the 5 “COM Generator” logic blocks shown in the schematic above. It allows for different wave shapes per color as well as different amplitude per-color, per-waveform and even per-peak withing a given waveform. This mechanism cannot be publicly discussed, however, without an NDA in place.

For a brief description of the waveform editor I wrote, see the “waveform workshop” section later in this portfolio.

The Genesis (continued)

The Waveform Workshop

One of the more challenging aspects of creating circuitry which succesfully drives the Ricoh Gen 5 print head is the shape of the COM waveform. The COM is a high-voltage, high-amplitude analog signal which is used to pulse the actual piezo crystals in the print head. This pulsing is what causes the ink to eject from the nozzles. The COM signal must be closely synchronized with an additional set of digital signals which define excatly “what” will happen when the COM wave fires.



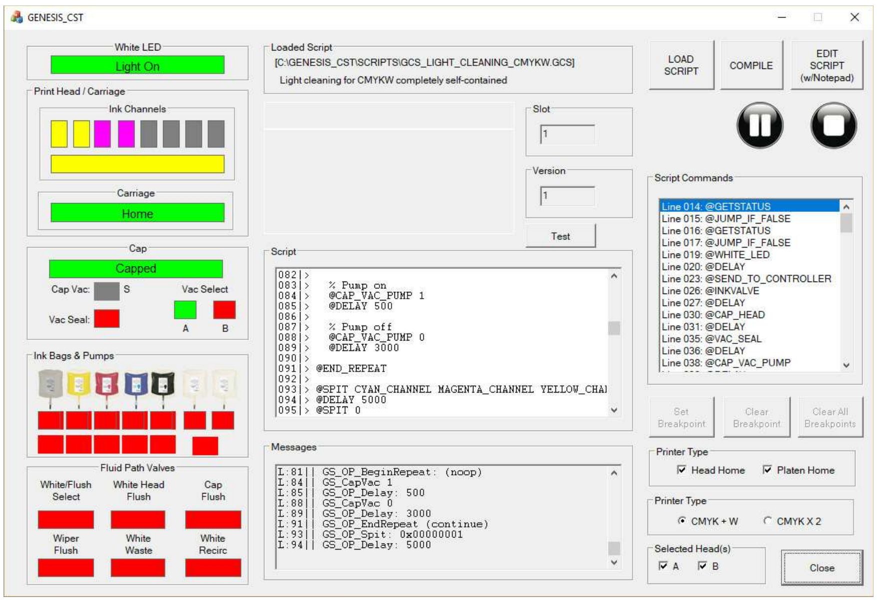
The documentation for the print head defines the maximum and minimum timings for the COM wave. It also depicts the “suggested” waveform (which is formed by using the mean of these limits). This default wave is a simple inverted triangular wave. This waveform must be changed significantly, however, depending on the ink being used. After many months of tuning, the actual waveform used by the Genesis looks nothing like the default waveform described in the head documentation. Ricoh does offer a waveform editor system to its OEMs. However, while their software allows for the tuning of triangular waveforms, it is lacking the ability to create composite waveforms. So, I also developed the software tool which, used in conjunction with the high-speed camera included in the Ricoh tool, allows very fine control over the shape of a composite COM wave. This software is called the Waveform Workshop. It allows for the definition of different wave shapes by defining the start/end voltages and durations for an up to 6-wave burst. It also allows for adjusting the inter-wave spacing, inter-sequence spacing, etc. Very complex waveforms can be defined.

Using this tool along with a hybrid version of the Ricoh test jig which includes Belquette hardware and the high-speed camera, the engineer can see the effect of the waveform change instantly on the screen. He/she can see the spot size, shape, formation and speed instantly as changes are made to the waveform using the software. Obviously the resulting waveform can be seen on an oscilloscope as changes are made with the workshop.

The Genesis (continued)

The Cleaning Routine Script Editor/Emulator

By far one of the most callenging things about designing an inkjet printer is defining the cleaning routines. How to apply positive or negative pressure and for how long, when to wipe, when to apply what, etc. etc. It can be a full time project just figuring out these variables. To make changes in the firmware, compile, load, reboot the device, etc. per test iteration can be very time consuming when defining the routines. For this reason, I designed a scripting system. The core firmware has default cleaning routine code which it defaults to, however, cleaning scripts can be downloaded to overwrite or add cleaning routines.



To be sure that the steps the script generates are as the engineer intended, the script editor is also able to "execute" the script. While executing, the engineer can see the state of all the different pumps, valves, etc. as the script executes. The script can be paused, restarted, etc.

To make this functionality possible, I defined a scripting language which can be interpreted by the printer's firmware (soft-controller inside the main FPGA) as well as the editor/emulator app. which is a Windows program.

There are plans in place by C-Horse Software to expand this scripting language. It could then be used to define the complete functionality of a device such as a printer by a non-programmer type engineer. This would open a whole new market for supplying OEM board-level functionality for printers and other devices.

The script can be executed and paused, stopped and restarted, etc.

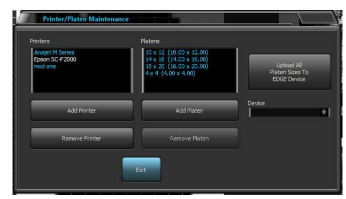
The Belquette Edge pretreater and supporting software and firmware

Another innovative product for which C-Horse did a significant amount of development is the Belquette Edge pretreater. This product features content-sensitive pretreat application. It will apply pretreat only where ink will be applied (with a user defined border).

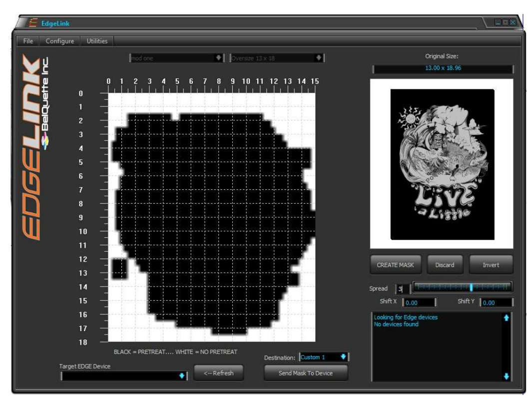
The product also includes PC software which collects the mask data from various types of files and ripped jobs. It can generate mask data from bitmaps, C-Breeze RIP files, PowerRIP files (all versions), Garment Creator, CADLink and AnaRIP.

C-Horse Software was responsible for all the on-board firmware (including motion control) as well as the PC software (EdgeLink).

EdgeLink is fully networkable and supports multiple RIPs and multiple pretreater units. It features a graphical easy to master GUI. Each instance of EdgeLink software can communicate with up to 100 Edge pretreaters, although realistically, each station would typically be limited to a handful of them.



The Edge control system consists of an all-in-one ARM motherboard running Windows CE and two additional sub-processors. C-Horse Software was responsible for the CE build as well as the actual control app. Out-of-the box, the SDK which came with the ARM board did not handle SPI or I2C communications correctly. In addition to making all the GPIO ports work correctly, I also fixed and modified the kernel-mode driver code in the CE build to properly handle SPI and I2C.



The EdgLink application includes a good bit of reverse-engineering which was performed to be able to extract the mask shape from other RIP's print data. It also includes the ability to define different size platens as users often procure specialty platens for their printers.

The code for the additional sub-processors include all the motion control for the platen and the spray nozzle. As with all other projects, it provides smooth acceleration/deceleration for both the servo and stepper axes.

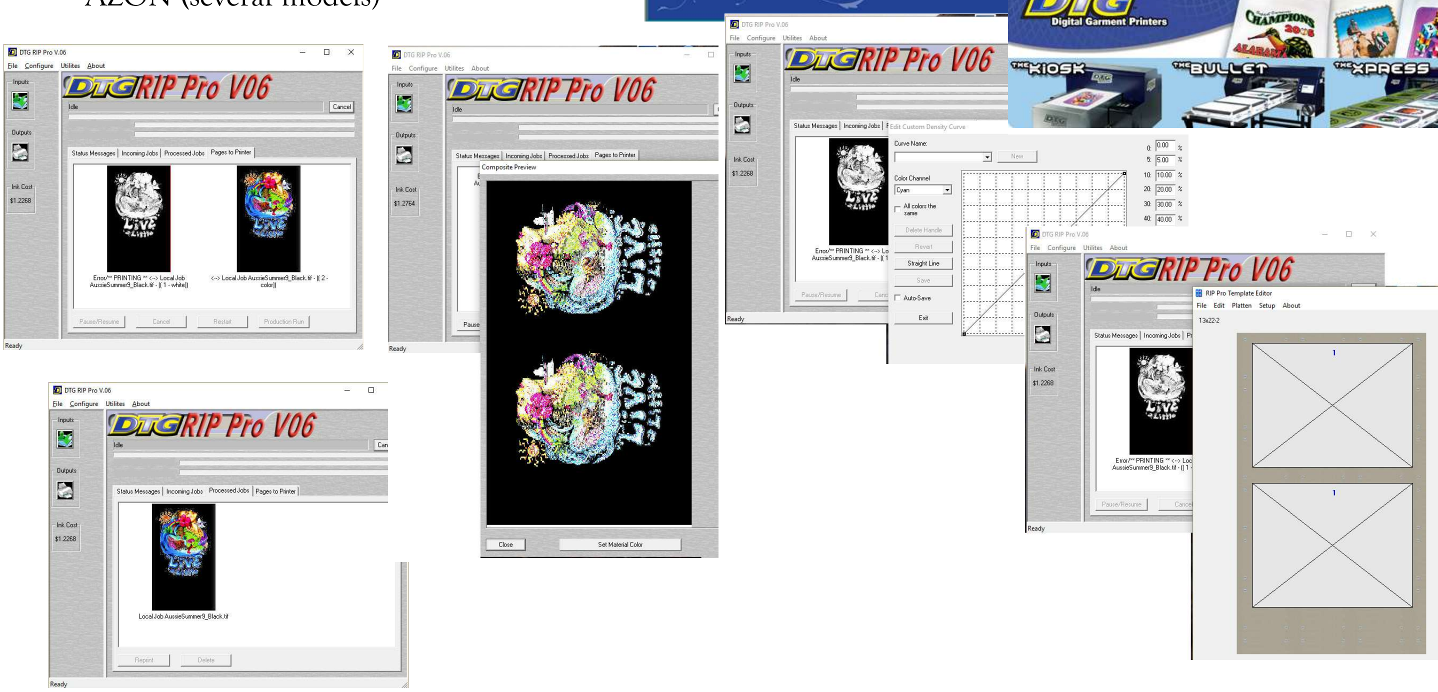
PowerRIP line of Direct-To-Garment RIP Software (sold by iProof Systems)

For several years, the iProof direct-to-garment RIP occupied the number one spot in the direct-to-garment market. Its quality, ease of use and functionality were unequaled and the product was sought by many direct-to-garment distributors and developers. The RIP reached an installed user base of over 5,000 users. This is a pretty significant number considering the nature of the early dtg market.

Due to business decisions beyond the control of C-Horse Software, that product, albeit still in existence, no longer holds the number one position. However, all the knowledge and expertise that came from developing that product, lives on in the C-Breeze line of RIPs. This product is briefly outlined in this portfolio. More details on the product can be seen at **www.c-breezerip.com**

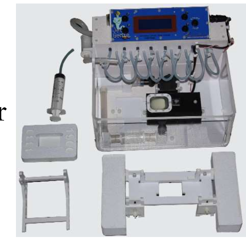
The software was available for the following printers:

Belquette Flexi-Jet
DTG Kiosk
DTG Bullet
DTG Express
DTG K3
DTG Summit
Belquette Mod One
Sawgrass Direct Advantage
SpringSun (several models)
AZON (several models)

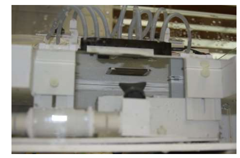


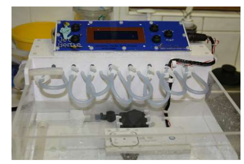
The innovations present in this software have since been emulated by other software companies, for example, the imposition / step & repeat mechanism which uses ID numbers to place graphics in particular locations.

Jet Genie and Jet Genie II Inkjet head cleaner



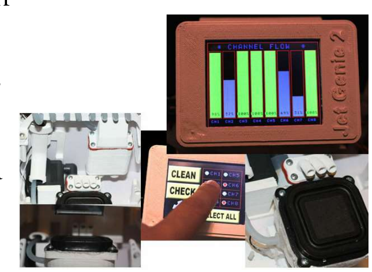
This product was designed and sold by C-Horse Software, Inc. It is designed to clean inkjet heads. It was specifically designed to clean the Epson DX5 head which was and still is, a very popular head for re-purposed direct-to-garment applications. It has been known to revive print heads that would have otherwise been deemed “unrecoverable”



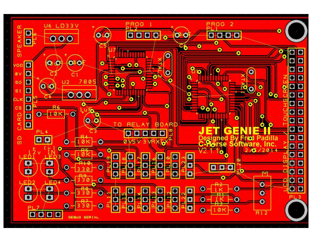


The product has since been discotinued due to lack of time. The Genesis project took the majority of the available development time. It was therefore decided to temporarily cease development and marketing for the Jet Genie line. Next year will see more than likely see the release of version 3 of the product. That version is expected to do very well as it will benefit from all the lessons learned from versions 1 and 2 over the last 7 years.

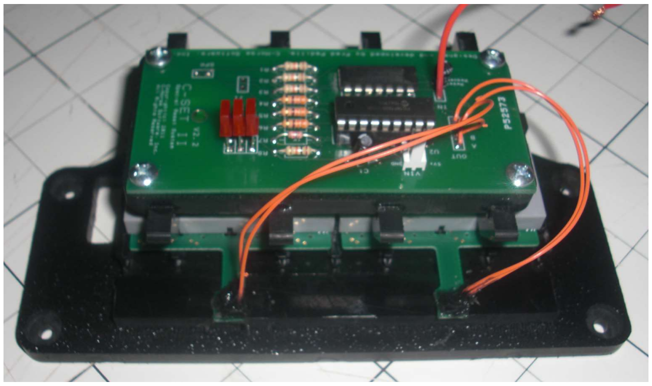
I discovered early on that while vibration is effective in clearing inkjet clogs, the source of the vibration and direction of acoustic waves is not optimal when a jewelry cleaner or other sonic device is used. Such cleaners employ (as they should) a more omni-directional source of acoustic waves. To clean clogs from an inkjet head, however, omni-directional waves do little good. What is needed are acoustic waves which strike the nozzle plate and travel up the nozzle opening. Furthermore, ultrasonic cleaners have a tendency to break down the adhesive that holds the nozzle plate to the head and thus delaminate the head, rendering it useless. The Jet Genie design uses a high-amplitude low frequency acoustic sounder placed directly below the nozzle place. This provides agressive acoustic waves which smash much harder into the clog. Because these waves are low fequency (60 to 100 Hz), however, they are much less likely to delaminate or otherwise harm the print head. Obviously, the solution used in the cleaner also makes a significant difference on whether or not the print head survives the cleaning.



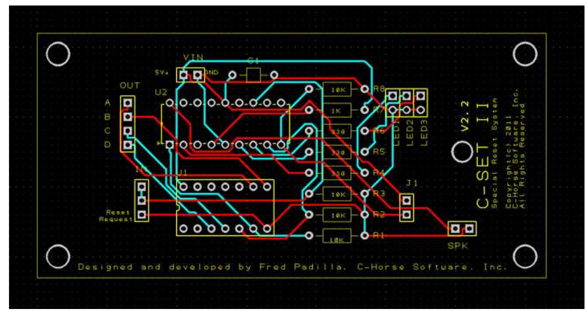
The second iteration of the product (Jet Genie II) improved the design in several ways. With version 1, the user would select the channels to clean by manually clamping tubes with built-in clamps. Version 2 included servo pinch valves that would do it automatically. Version 2 also features a touch-screen and a pressure transducer which measures the flow of the nozzles with a reasonably good degree of accuracy. This helps you discern when the head is actually cleaned. Jet Genie II is also smaller and quieter. Most all ABS components are 3D printed, thus making the production of replacement parts extremely easy and inexpensive. The

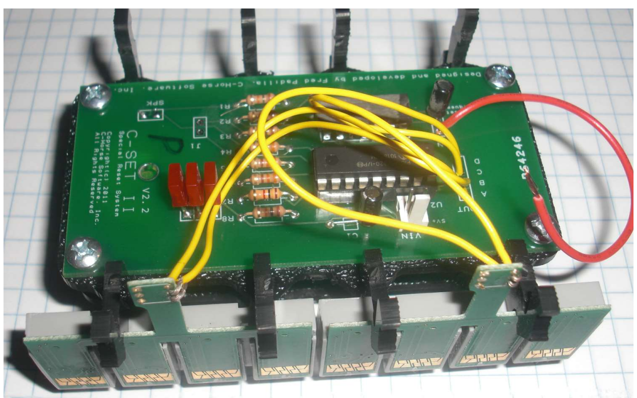


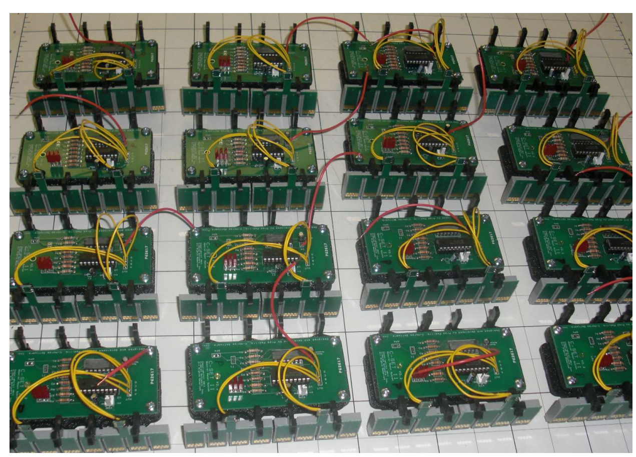
C-Setter ink chip reset module



This hardware module was designed for specialty printers based on the Epson Stylus Color R1900. These printers normally used bulk-ink supply tanks instead of cartridges. For this reason, modules are used which simulate a full cartridge. These modules are typically reset by hand with a built-in button whenever they read empty. The C-Set module was designed to make it possible for the modified printer's hardware to initiate the reset rather than a manual buton press. It consists of a microprocessor which polls a signal for a reset request. When it sense the request it simulates a button press by temporarily holding low two of the triggers on a 4066 quad bilateral gate CMOS chip. As an unexpected (and welcomed) side effect of the chosen components (perhaps mixing TTL and CMOS) the device caused the Epson R1900 to execute a reset without executing a clean. This saved around 3 minutes per reset. On a direct-to-garment printer running white ink, this would mean a 3 minute time saving every 2 to 3 t-shirts printed. This made the device very popular with several companies. The following copanies purchased these on a regular basis: SWF East, SWF Mesa, Impression Technologies, Yes Technologies, Direct Color Systems and Equipment Zone. A few hudred of these were shipped. Although finiky and fragile, mostly because of the poor quality of the Chinese resetters I had to use, these devices were a Godsend when they were working. A good majority of the hundreds that were sold are still in operation today.





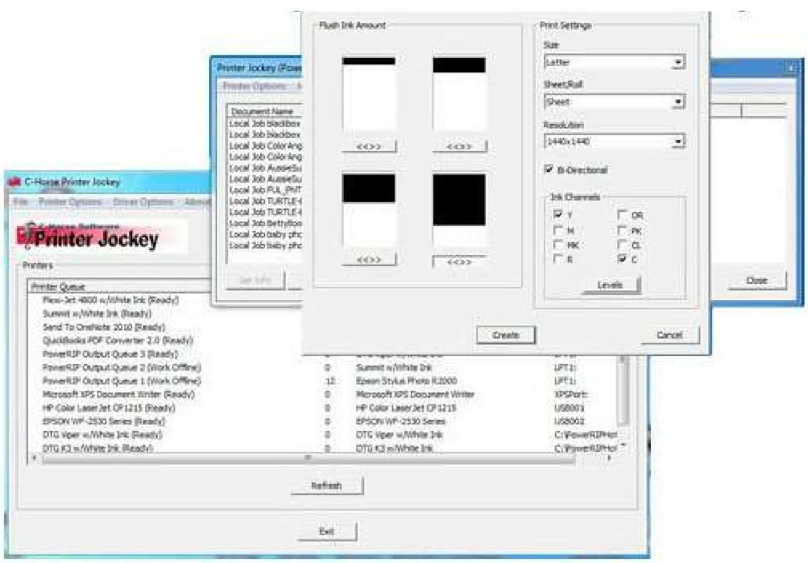


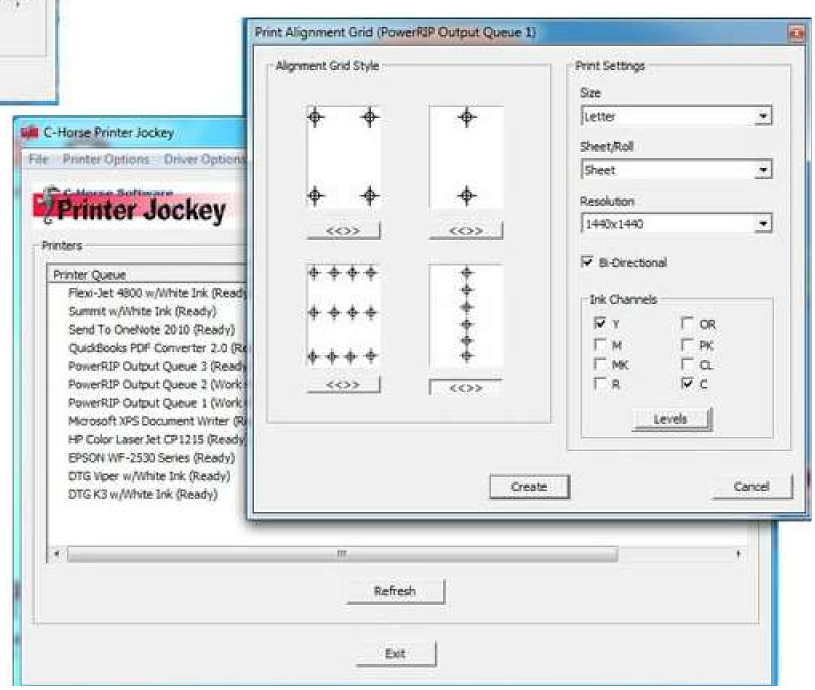
Printer Jockey Software

Printer Jockey is a printer utility. It facilitates several printer-related tasks but its main function, the main purpose for its inception, is to “flush out” single channels or combination of channels.

Often during direct-to-garment printing, it is necessary to exercise a single channel or maybe a couple of channels. For example, the Cyan ink is replaced and it develops some bubbles in the print head. Using the printer’s cleaning or power-cleaning routines would spend the other 3 (or 4) ink channels while attempting to clear the Cyan channel. Using Printer Jockey a user can exercise the Cyan channel until the bubble is gone, saving considerable amounts of the other colors.

This product has shown very good longevity. Surprisingly, it is still purchased and downloaded at least weekly, even though the supported printers list does not include a lot of the newer printers being re-purposed for direct-to-garment.



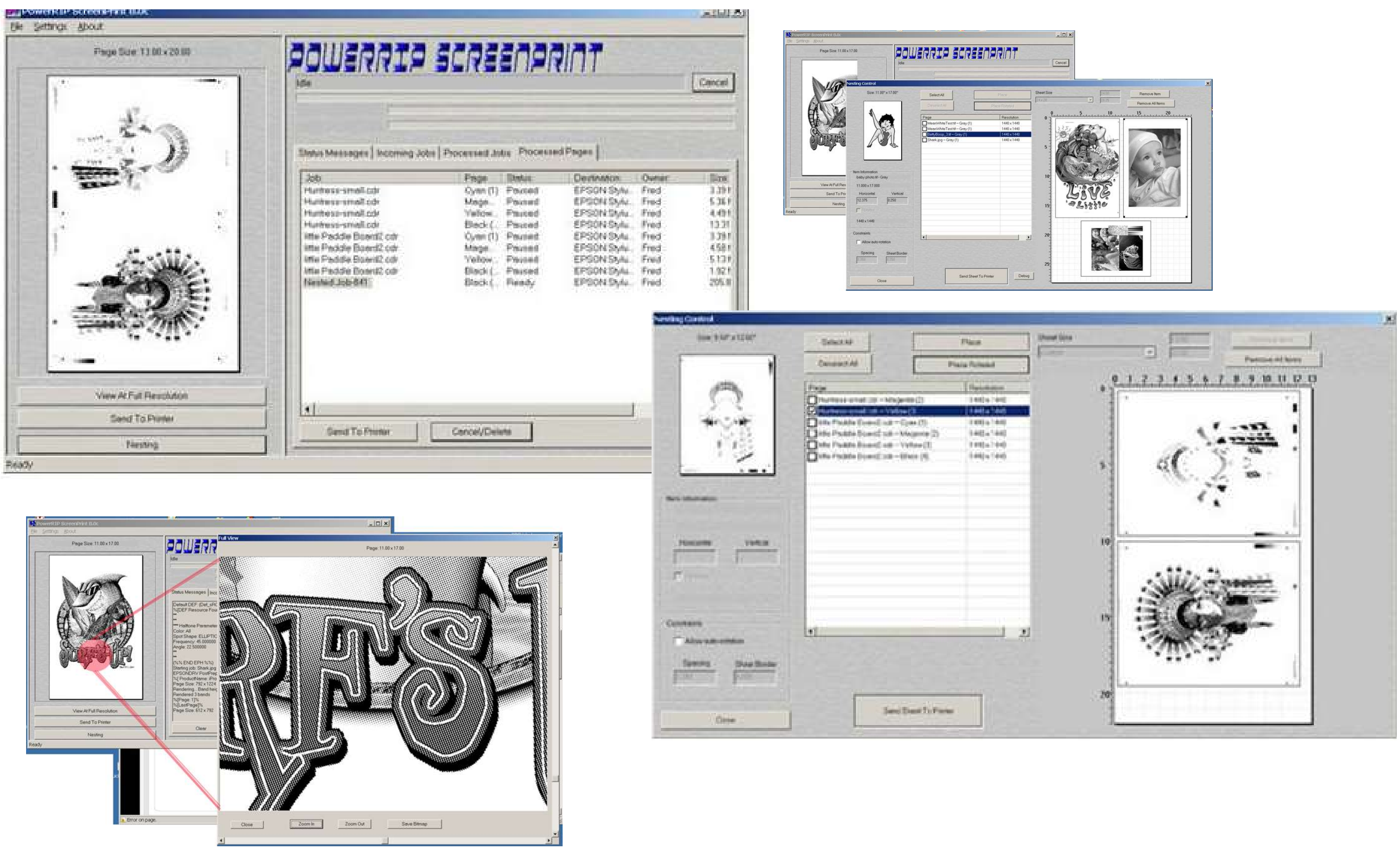


PowerRIP ScreenPrint and private-labeled variations (written for iProof

One of the more sucessfull packages I designed and wrote for iProof Systems is the ScreenPrint version of PowerRIP. This product is still being sold and competes relatively well in a somewhat saturated market. It includes several features that were not adopted by other similar RIPs for several years. The FM screening (stochastic) implementation is still, in my opinion, the best and most flexible on the market.

The product was private labeled for several companies and sold under various names. It includes a very easy to use but powerful imposition system (show below). Also, it displays a job thumbnail in real-time as it's being ripped. This avoids waste. In case of a printing mistake, it won't cost the user an expensive piece of film to find out.

There are several more variations and versions of this software that are not listed in this portfolio as they don't necessarily illustrate any addition to the skillset.



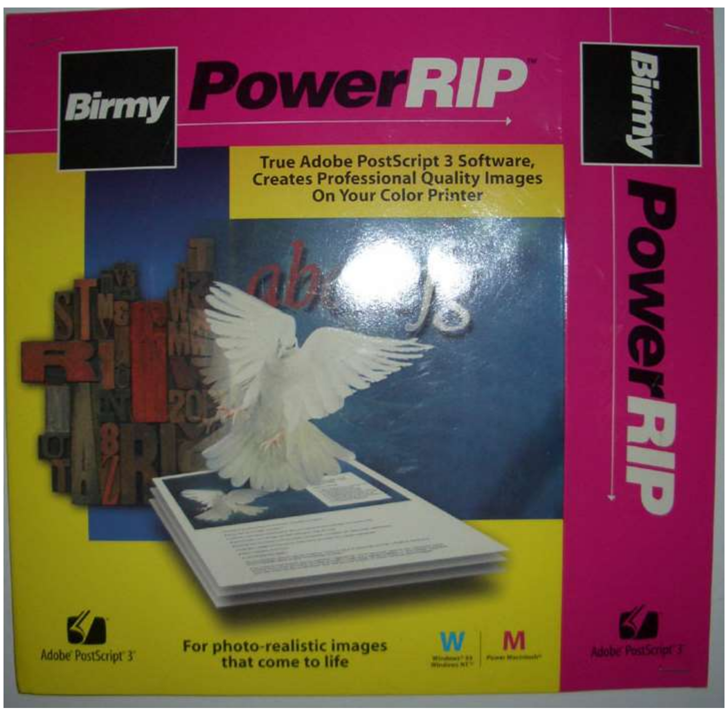
SimplyRIP (written for iProof Systems)

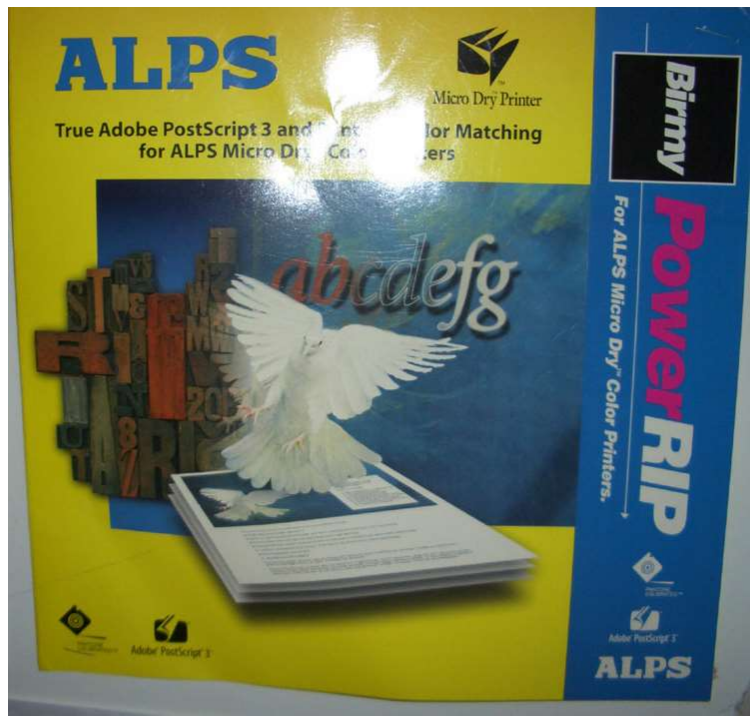
This is a variation of the ScreenPrint RIP version which was tailored to **print directly to printing plates**. It was commisioned by a large distributor of Mitsubishi plates and paid for by Mitsubishi themselves. This version includes advanced PDF handling.

Desktop Proofing RIP Solutions (1990's)

After the company (Birmy Graphics, now called iProof Systems) exited the laser-imagesetter RIP market, it became one of the leaders in desktop proofing RIP solutions. As a matter of fact, Birmy Graphics was contracted by Epson America to build the very first PC and Mac version of a PostScript RIP for the Epson Stylus COLOR. This was actually the very first inkjet printer in the world. I designed and wrote the PC version. Over the years, the code I wrote for the PC was all ported to the Mac. As a matter of fact, the majority of the Mac version code is simply my code ported to a different platform.

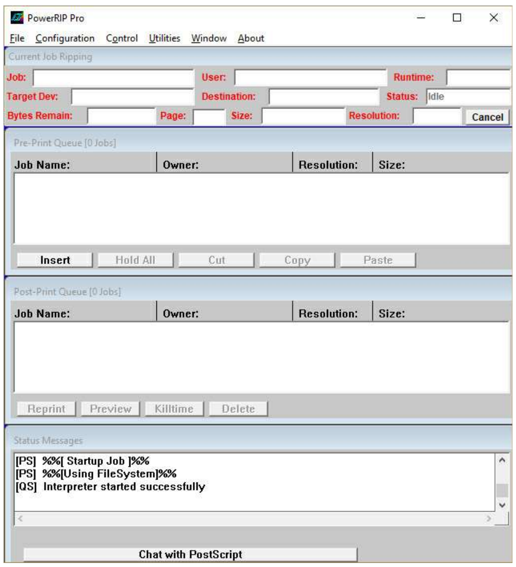
Later, Canon, HP and Alps approached Birmy to develop RIP solutions. These versions sold very well for several years. I estimate there are approximately 70,000 users of these versions worldwide.



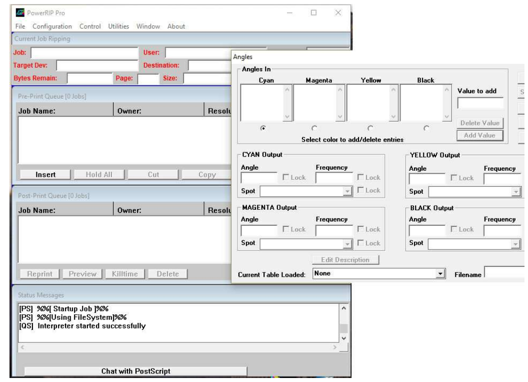


PowerRIP Pro ImageSetter RIP (1990's. Written for Birmy Graphics)

The last laser imagesetter version I wrote for Birmy Graphics. A few months after this version was launched (and it was actually doing well), Agfa decided to start giving away imagesetter & RIP packages in exchange for film contracts. This proved too much to compete with and thus the majority of 3rd party RIPs for the UltreSetter and other engines gracefully bowed out of the market. Birmy Graphic concentrated on color RIPs going forward.

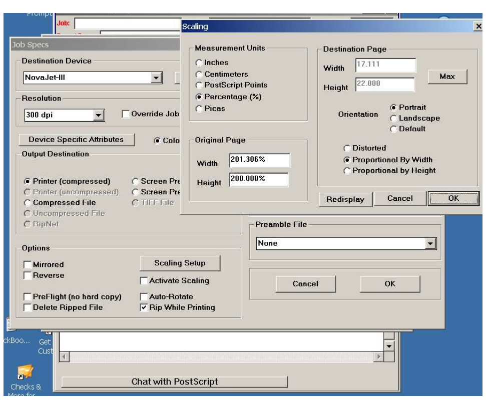


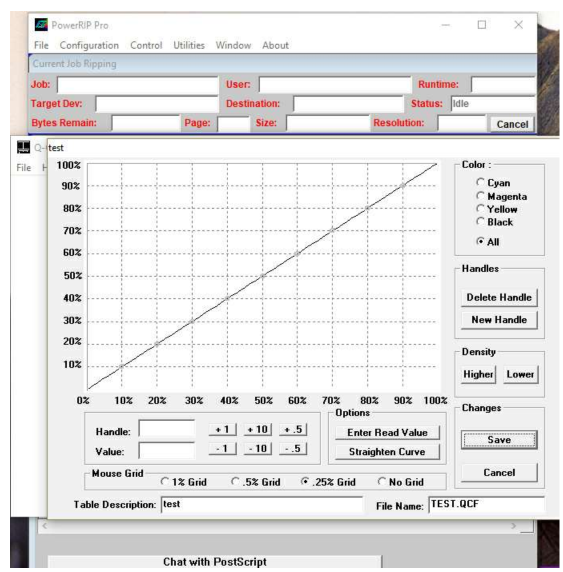
PowerRIP Pro drove all the major laser markers such as the Linotype series (all of them), the ECRM Pelbox series (all of them) and the Ultre and Extra imagesetters (all models). In addition, it supported the Canon Bubble Jet printer which was used for proofing before imaging on film. The Bubble-Jet printer could accomodate a full newspaper page. This made it a very attractive proofing solution.



This was also the begining of the "color era" for Birmy Graphics as this RIP also supported a popular large-format color inkjet called the NovaJet.

This was Birmy Graphics' first Adobe CPSI implementation. It ran under Windows NT. I was the first person to succesfully multi-thread the CPSI interpreter. According to Adobe Systems, the CPSI interpreter did not support multi-threading. This resulted in that ALL CPSI implementation on Windows NT were very "choppy" and unresponsive in their behavior, particularly the user interface, while ripping a job. However, by protecting certain areas and intellingently structuring my code, I was able to employ a worker thread to run the CPSI core, leaving the main application thread free to handle all the OS-specific code as well as the user interface. The result was a very smooth user experience with no lag and no mouse hesitation. Subsequently, Adobe published my method as the recommended way to multi-thread the CPSI interpreter.





Although now considered a very old program, PowerRIP Pro included many of the popular features now found in modern RIPs such as custom linearization and screening-parameter overrides.

RIP-Net (1990's. written for Birmy Graphics)

I designed this high-production imagesetter RIP system for newspapers. It was able to distribute the workload among a large number of RIPs and several ImageSetters of varying models. By now, PC speed had grown to the point that it was feasible to perform PostScript interpretation right on the PC without a co-processor board. The RIP and Q-Station code was intertwined. Q-Station ran in real mode while the RIP code (which was 32-bit) ran in protected mode using the Phar Lap DOS extender. The system woudl switch modes to share toe processor. I was able to throttle as needed to give protected or real mode more or less time. The first RIP-Net system to be installed served a newsroom with approximately 115 operators working on Macintosh computers. All the user had to do was click print and the system would automatically route the PostScript to an available RIP which created jobs for the first available imagesetter. The logic was pretty slick since not all the imagesetters were the same make and model and often not even the same resolution. In this installation there was a mixture of ECRM Pelbox imagesetters and Linotype imagesetters. The emagesetters were plugged, in pairs, into PCs running the "Print Unit Buffer" application which would in turn communicate with a "RIP farm" (pictured below) in order to properly schedule jobs.

The second RIP Net installation has the distinction of being installed into the first newspaper in history to come up 100% digital from day one. This newspaper (believe it or not) was Reforma in Mexico City. We also installed one in Monteray, Mexico at "El Norte" newspaper. Each installation was over $500,000

Q-Station (Written for Birmy Graphics, late 1980's)

My first commercial product as well as Birmy Graphic's. The BirmySetter was the first PostScript clone to ship in the U.S. It was also the first American-made PostScript imagesetter. The laser unit itself was manufactured by "Ultre" corporation. It was small, simple and inexpensive. It sparked the PostScript clone revolution in the late 80's. Birmy was the first to put a PostScript RIP on it. I lead the project and was considered one of the idustry's leading experts on RIP technology for many years.

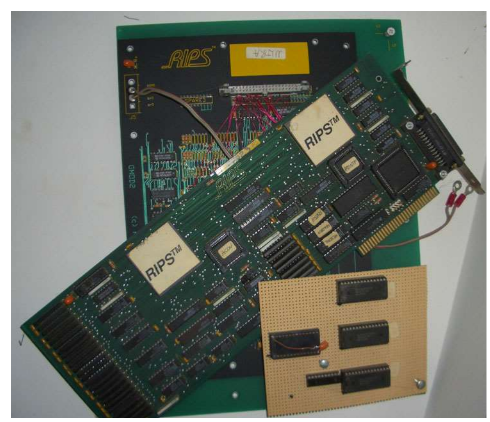
As PCs were still way too slow to rasterize PostScript, the BrimySetter used a co-processor board manufactured by RIPS (Raster Image Processing Systems, Inc.) in Boulder, Colorado. I worked very closely with R.I.P.S. Inc. for many months. As a matter of fact, I was part of the team that cracked the Adobe font encryption for type 3 fonts. If you know what the operators "eexec" and "cexec" are, you know what I'm talking about.

Q-Station was a PC program which ran in DOS (Windows wouldn't exist for a few years yet) and commuincated with the outside world and the RIPS co-processor board(s). Since I had an entire dedicated PC to handle I/O, I added features which were unique at the time but ended up being household words in the RIP business. I was the very first to add a lot of these which you may now know by different names.

Because the initial RIPS implementation did not include the eexec/cexec decryption, the Macintosh preamble file could not be executed "as is". So, implemented "Automatic Preamble Replacement" which is a fancy way of saying, "it would replace LaserPrep with my hacked-up version of LaserPrep on-the-fly.

For the same reason, the RIPS PostScript clone could not run "true" Adobe fonts (they contain eexec/cexec) so I implemented "Automatic Font Replacement". I had a table of font Adobe font names and a list of matching fonts with the Bitstream name. If would automatically perform the replacement, on the fly, as needed.

I was able to figure out how to publish multiple AppleTalk and EtherTalk on the network and thus one BirmySetter could accept jobs from up to 4 Macs simultaneously. Each name could be configured differently so by simply selecting the correct printer on the network (whic was just an alias for the same BirmySetter) you could request different resolutions and other parameters (mirror, negative, etc.)

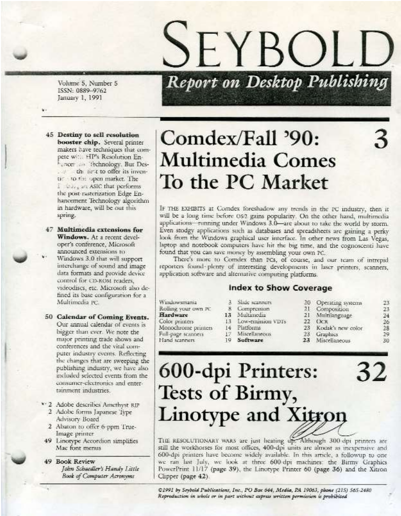


As all jobs would spool to disk, even if they were also being processed, it freed up the Mac user much faster and could be re-printed without having to re-send them. Because you could network the PC using off-the-shelf software (Novel Netware, for example) it was very simple to add the BirmySetter to a network.

For the same reason, I was able to implement what I called "Scan Folders" which you now know as "Hot Folders". Each scan folder would have a device, resolution and other attributes associated with it. Simply drop the PostScript job in the correct scan queue and that's it.

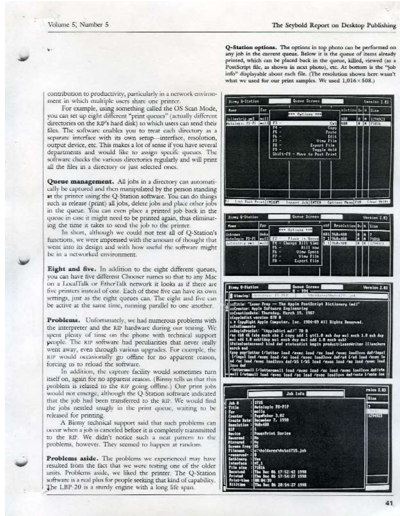
Q-Station and the BirmySetter were often written up in various publications. Most notable, The Seybold Report, which was once consiered "The Desktop Publishing Bible".

The original prototype RIPs coprocessor and interface board currently, shown above, hang on the office wall as a reminder of where and how it all started back in the 80's.


SEYBOLD
Report on Desktop Publishing
Volume 5, Number 5
ISSN: 0889-9762
January 1, 1991

Comdex/Fall '90: Multimedia Comes To the PC Market 3

600-dpi Printers: Tests of Birmy, Linotype and Xitron 32

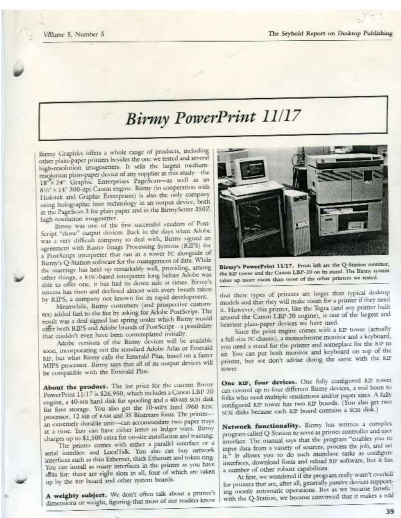

Volume 5, Number 5 — The Seybold Report on Desktop Publishing

41


September 11, 1989 — The Seybold Report on Publishing Systems

Birmy Graphics

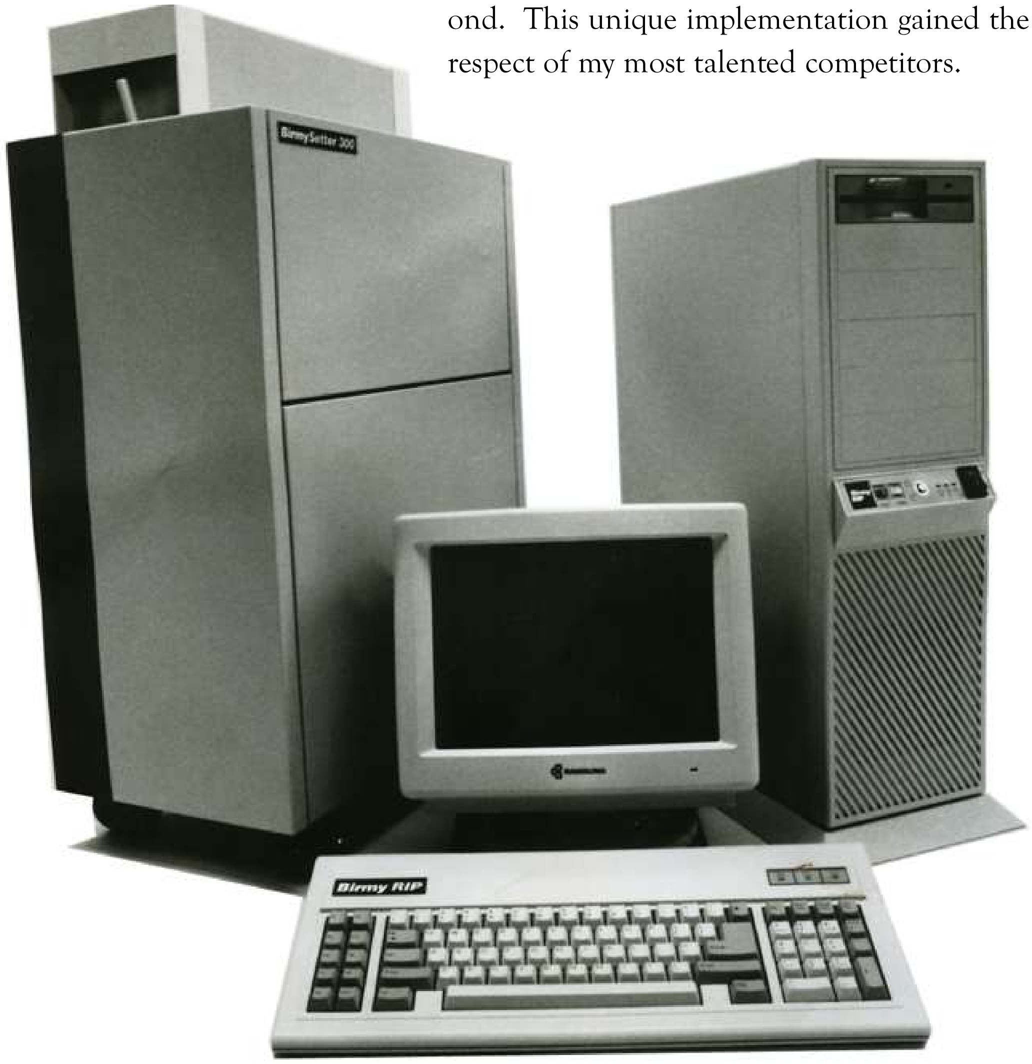
12


Volume 5, Number 5 — The Seybold Report on Desktop Publishing

Birmy PowerPrint 11/17

39

One of several articles published by The Seybold Report back in the late 80's and early 90's. There were only a handful of RIPs back then and the Birmy RIP (Q-Station / BirmySetter) was the only one written in America. With the advent of the Intel 80486, PCs finally became fast enough to rasterize PostScript and we did away with the CoProcessor board. Q-Station then became the first RIP to run in both real and protected mode on the Intel processor. The Q-Station code in real mode, the PostScript rasterizer in protected mode, switching back and forth 18 times per second. This unique implementation gained the respect of my most talented competitors.



Contact Information

C-Horse Software, Inc.
1510 Charles Blvd N.E.
Palm Bay, FL 32907

email: c-horse@cfl.rr.com

Phone: 321-917-6969

URL: www.c-horsesoftware.com
www.c-breezerip.com