CIRCUIT CELLAR®
THE MAGAZINE FOR COMPUTER APPLICATIONS

FEATURE ARTICLE

by Peter Csaszar

Nixie Tube Propeller Clock

Peter built his Nixie Tube Propeller Clock (NTPC) around a Microchip PIC16F84A microcontroller. The firmware was developed in assembly language with the MPLAB development suite.

In this article, I'll introduce a propeller clock that features a nixie tube, which I named, quite creatively, the Nixie Tube Propeller Clock (NTPC). The primary purposes of my project were to demonstrate the application of this simple yet capable device and to have a little fun in the process.

The heart of the design is a Microchip Technology PIC16F84A microcontroller. I developed the firmware in its native assembly language using the MPLAB development suite.

NIXIE TUBES

The nixie tube is one of the most ancient electronic display technologies, predating LEDs, LCDs, and other contemporary displays. (It is now an obsolete technology that's seen only in old Cold War video footage as the symbol of the era's state-of-the-art technology.) The tube is a cold cathode discharge tube with a shape resembling that of a vacuum tube, although its cathodes are not heated (hence the name). In principle, nixie tubes are the same as the neon bulbs typically found in the illuminated switches on surge protectors.

Photo 1 shows the Matsushita Electric Industrial Co. (Panasonic) CD72P nixie tube that I used for this project. The glass enclosure contains neon gas at a low pressure. When a sufficiently high voltage is connected between the cathode (negative) and the anode (positive), the neon gas ionizes in a rather complex process. As a result, neon ions and electrons fly to the cathode and anode, respectively. The anode quietly swallows the electrons; however, the recombination of neon ions and electrons covers the cathode in a beautiful, ethereal orange glow. Refer to G.F. Weston's *Cold Cathode Glow Discharge Tubes* for more information about cold cathode tubes.

The aforementioned phenomenon is used in the nixie tube to display characters (typically numeric). Multiple cathodes etched to the shape of each supported character are stacked on top of each other, standing behind the anode mesh that resembles miniature chicken wire. Energizing one of the cathodes and the anode produces a glow in the shape of the selected character.

This is a delicate and fragile technology that has been extinct from manufacturing for decades. Yet, the sparkling glass case and glowing neon has kept a group of nixie enthusiasts loyal ever since. The community got a great boost in 2002, when artistic inventor Raymond Weisling of Zetalink Technology founded the NEONIXIE-L Yahoo group. Today, the group has more than 1,500 members across the globe, with new members joining every week.

Finding an actual nixie can be tricky because vendors don't always carry them. But there is a plethora of nixie-related information on the Internet. Refer to the web sites listed in the Resources section of this article for more information.

PROPELLER DISPLAYS

The fundamental and truly ingenious idea behind propeller displays is based on the phenomenon of positive after-images, which is also known by the obsolete term "persistence of vision." It's the principle of producing a wide display perception from a narrower but physically moving display element. (The technology deserves to become a new meaning for the term "spatial multiplexing.")

Bob Blick created the first propeller clock, which spins a single column of LEDs along a horizontal circle to produce the display on a cylindrical surface. (A similar approach is used for "window wiper" clocks like the Fantazein Message Clock.) My project was a fun way to combine an exotic

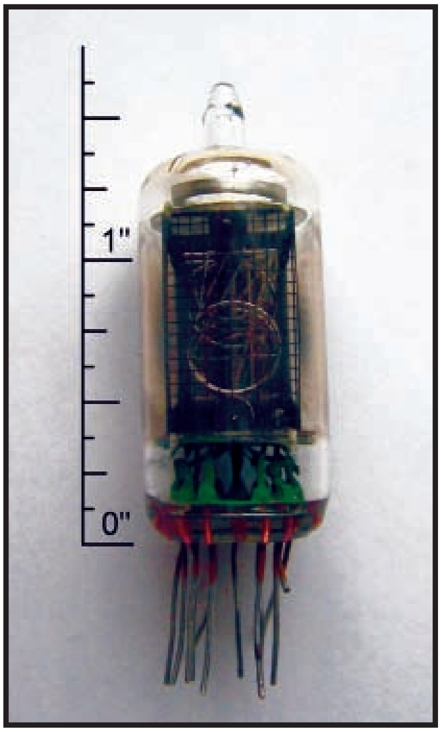

Photo 1—*The CD72P nixie tube is relatively small. Its size and wire pins make it optimal for the propeller display application.*

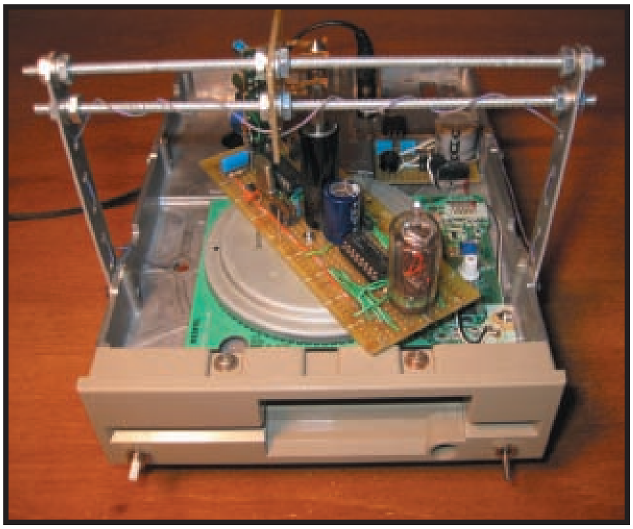

Photo 2—*The Nixie Tube Propeller Clock is built on an old 5.25″ floppy drive turned upside down. In addition to the speed-controlled motor, an infrared LED (inside the white plastic cylinder) and a photodiode can be used from the drive electronics.*

display method with an unusual display device. It replaces the column of LEDs with a nixie tube.

FLIP THE FLOPPY!

The very first thing that you need to address before building a propeller display is the type of arrangement that will provide the rotary motion. The rest of the mechanical design hinges upon this decision.

As an avid collector of computer junk (more precisely, all electronic junk), a little browsing among my musty boxes turned up the solution. The Holy Grail was a long-forgotten 5.25″ floppy drive, which I found sitting at the bottom of a box with a few other drives.

The plan immediately became clear. I realized that I could implement the propeller display by attaching a rectangular circuit board to the floppy motor and mounting a side-view nixie tube upright to one end of the board facing outward (see Photo 2). That unit proved to be an ideal choice.

First, the size of the drive is about right, even if every point of the rotated circuit board remains within the confines of the drive, which is recommended for a simpler and safer mechanical implementation. However, the real boon is the drive's speed-controlled motor, which spins away at 300 or 360 RPM, depending on the state of the SPD input signal received by the motor electronics. A single revolving nixie tube's readout is not completely flicker-free (even at the higher speed), but it's easily legible. (The very first experiment I performed was to verify if this central conjecture was true or false.)

Turning the drive upside down revealed the motor electronics and the magnet-filled metal plate acting as the rotor. After stripping all of the other components from the drive, I mounted the propeller clock's circuit board to this side, as opposed to the original top. The three tiny holes on the plate seemed perfect for holding the mounting screws. With a little effort, I pried the plate off of the rotating hub. After that, I put the mounting screws in the holes and tightened them. I then glued the plate (referred to as "carousel" henceforth) back to its original place.

The clock's electronics were assembled on a 5″ × 1.75″ prototyping board, but a PCB design could probably achieve a narrower footprint. In any case, it is important that the board is balanced (i.e., its center of gravity falls as close as possible to the point at which the rotation axis intersects the board). The device's Achilles heel—and the part that required the most precision—was the array of linkages for the power and signal lines between the stationary world and the rotating clock circuitry. That problem is typically solved using commercially available slip rings. However, an extensive search for a suitable replacement came back empty, so the NTPC received a homegrown "slip ring" assembly. It consists of a 0.25″ stereo headphone plug screwed vertically right in the

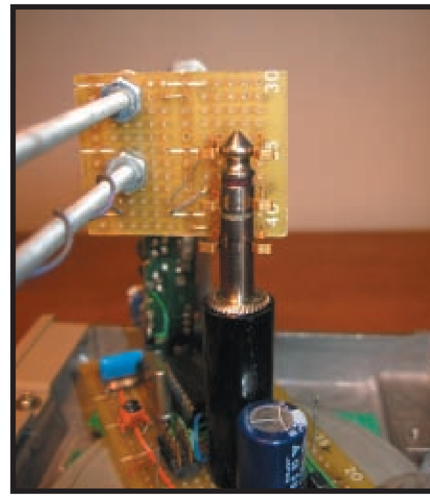

Photo 3—*Check out the homemade "slip ring" solution. The 0.25″ stereo headphone plug is another off-the-shelf component that turned out to be ideal for its intended purpose in the clock. It is fastened to the board by a screw after dropping a 0.375″ diameter #10 washer in its plastic case.*

middle of the circuit board (exactly above the rotation axis) and a set of stationary contacts (see Photos 2 and 3). I yanked the contacts out of an old edge connector and soldered on a piece of prototyping board, which was then mounted on the chassis using old Mecano pieces and threaded aluminum rods. (If you want to build a similar clock, you're free to come up with other creative ideas.) With an occasional spray of WD-40 for lubrication, this slip ring has been churning away without a glitch. The contacts can withstand the slight wiggle of the spinning headphone plug, provided that they aren't pressing against the plug too tightly.

The stereo headphone plug has three connections (two of which are needed for power), so there is only one signal line left for any widgets to control the clock. The NTPC implements one push button. Anything more would complicate matters significantly. Fans of the one-button Apple mouse will certainly appreciate the user interface, which required some extra finesse on the firmware side.

ELECTRONIC DESIGN

The NTPC's floppy motor electronics demand 5- and 12-V DC power supplies (see Figure 1). These supplies are placed on the stationary power board. The four-way rectified and filtered power of the 9-V AC adapter comes out at about 11 V RMS, with a 1-V_{pp} ripple that's a good enough 12 V to run the floppy's motor flawlessly. The 5 V are generated with the classic 7805 voltage regulator. The clock's electronics are powered from this 5-V source as well.

The power board also hosts jumper JP1 to select between the two speeds of the floppy motor (just for fun, 360 RPM is always the recommended speed). In addition, this is where the incoming wire from SW2 is connected. This switch is mounted on the face plastic of the floppy drive. It stops the motor for in-circuit firmware upgrade or test measurement purposes.

The nixie tube, as well as all of the clock electronics driving it, are placed on the spinning board, which is the circuit board mounted on the carousel. Capacitor C7 is responsible for removing slip ring noise from the power signals. The PIC16F84A microcontroller implements all of the clock's functions, including time keeping. The signals from the environment are received by inputs on port B. The firmware turns on the port's internal pull-up resistors. Port A provides the one-digit BCD signal, plus a bit for driving the nixie's decimal point. The oscillator is formed by the microcontroller's internal circuitry and the external 19.6608-MHz quartz crystal X1.

This device is a clock (albeit quite eccentric, no pun intended), so it's a good idea to replace the crystal by a temperature-compensated crystal oscillator (TCXO), whose frequency tolerance and stability are only a few PPM (10 to 100 times better than naked crystals, but for a heftier price). The TCXO's output is connected to the microcontroller's OSC1/CLKIN input.

The 74141 TTL chip is responsible for converting the BCD-encoded digit information to nixie-edible decoded signals. Functionally, this IC is an ordinary active low BCD-to-10 decoder. However, a closer look at the datasheet reveals that the outputs are open collector (i.e., the inactive outputs are high-impedance), and this is still not the entire story.

What's so special about the 74141 is that its output drivers are able to withstand high voltages appearing on the inactive outputs. That would make ordinary TTL outputs malfunction or burn out permanently. In fact, the 74141 was specifically designed to drive cold cathode tubes, which rendered the chips obsolete as soon as nixies themselves became obsolete. As a result, this part is no longer available through conventional channels. Fortunately, the special vendors who carry nixies often have inventories for this chip as well, and they also turn up regularly on Internet-based auction sites, especially the IC's Russian equivalent, which is more proof that the Cold War is really over.

The NTPC firmware also uses the nixie's decimal point, but the 74141 does not drive its cathode. But don't even think about connecting it directly to the PIC16F84A's RA4 pin! Coincidentally, this output is also an open collector (more exactly, an open drain); however, it does not possess the 74141's high voltage tolerance. Therefore, the decimal point must be controlled by a suitable transistor, such as the popular MPSA42, or the ZTX458 used in this circuit.

HIGH VOLTAGE SIDE

The price to pay for the warm neon glow that drives nixie enthusiasts crazy is the burden of generating the high voltage that drives the nixie itself. Let's focus on the requirements and possible solutions.

Like most nixies, the CD72P requires a 170-V ionization voltage to turn on the cathodes quickly and reliably. Once ionization has gained full momentum, a smaller voltage called the sustaining voltage is sufficient to keep the nixie lit. The recommended operating current for the CD72P is 2 mA. At that current, the voltage drop on the tube is about 130 V.

High DC voltage generation in a DC-powered circuit

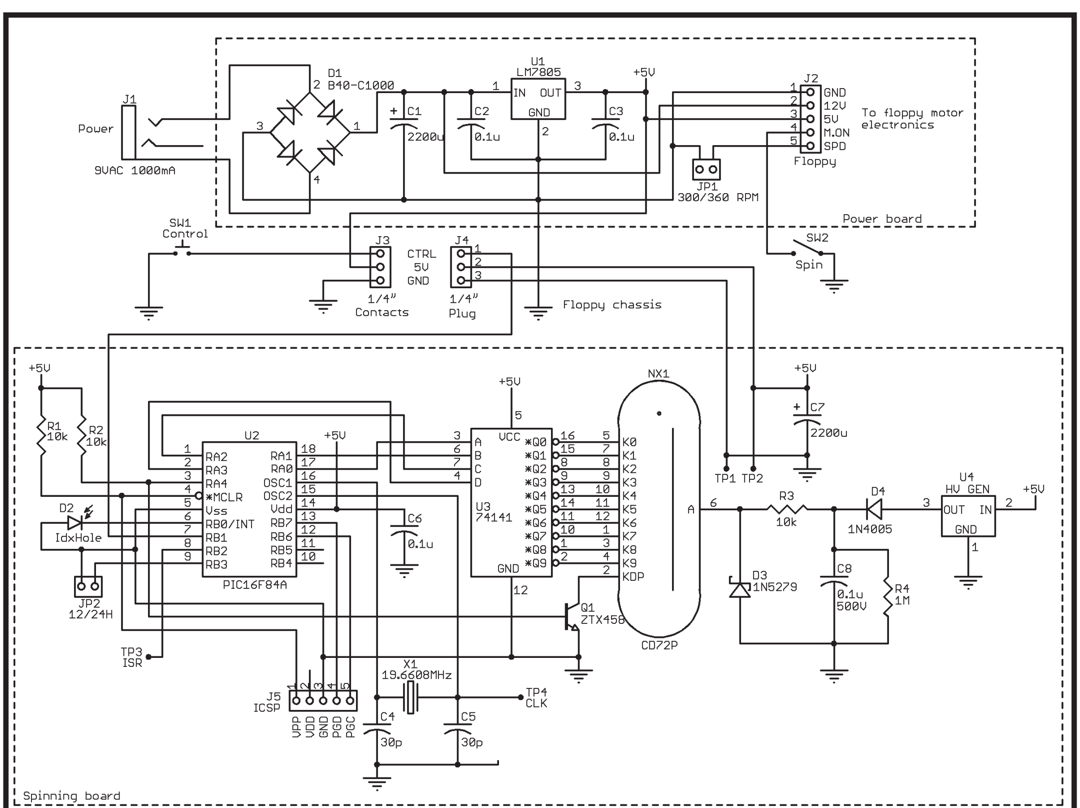

Figure 1—*The clock features two circuit boards. The components hosted by the boards are clearly identified.*

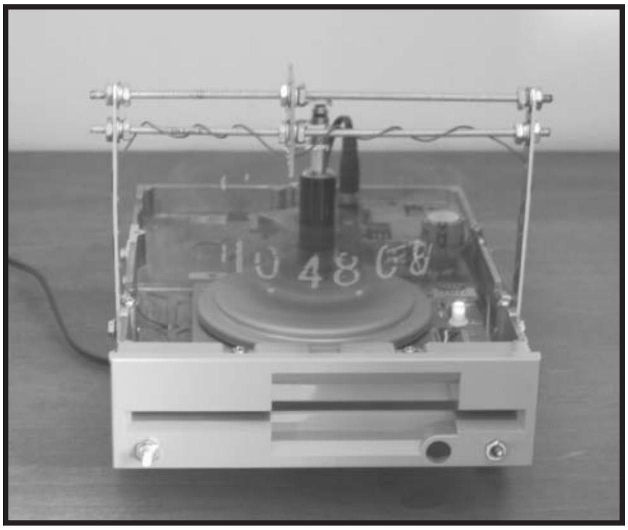

Photo 4—*Notice one of the best-kept secrets in the digital camera industry: the near-infrared radiation of the LED shows up as a white glow with a slight bluish hue.*

requires a DC/DC converter. A popular genre of circuits is a Switching mode power supply that uses a single coil as the precipitator of high voltages. These are typically based on specialized chips, such as Maxim's MAX771; however, I've read about a similar circuit that uses a plain 555 timer.[1] Regardless of the details, these circuits often have arcane traps, such as sensitivity to board layout. A search on the Internet for "nixie power supply" reveals kits and circuits that offer an easier alternative.

I followed yet another simple approach for the NTPC. Most small, battery-operated fluorescent lights available in hardware stores contain a simple circuit to produce the high voltage necessary for operation. Luckily, the circuit taken from one of these lamps also turned out to be suitable for squeezing out the juice for one nixie tube.

I didn't include the schematic for the reverse-engineered circuit in this article. Suffice it to say that it implements a relaxation oscillator connected to a transformer. When the relaxation oscillator "pops," a pulse of up to 400 V is induced on the transformer's secondary coil. Let's define this pulse as a positive voltage. In the remainder of the cycle, an almost flat curve occurs between –50 and –100 V. The pulses are caught with D4 and C8, and the desired high voltage is now available. R3 sets the nixie's current to 2 mA. (R4 is the so-called bleeder resistor that makes sure that after power-down, the high voltage is removed from the capacitor.) This DC/DC converter is not very efficient (about 40% at a 2-mA output), and it is only capable of producing 2.5 mA before the voltage on its loaded output drops below 130 V. However, it can easily drive multiplexed displays (including the "spatially multiplexed" propeller display).

The role of D3 (a 180-V Zener diode) may require some explanation. When none of the nixie digits are on, the converter's load-free output voltage jumps to the sky. This alone would not be a problem. However, due to the protective 70-V Zener diodes inside the 74141 between the outputs and ground, there is now a high enough voltage between the nixie's anode and all of the cathodes for ionization currents to start trickling, and the cathodes become immersed in an eerie, smudged orange haze. This phenomenon is referred to as ghosting, which is a consequence of the unregulated high-voltage supply. The 180-V Zener caps the nixie's anode voltage, busting the ghost problem once and for all.

Just a word of caution before carrying on: even though the high voltage converter can only output low currents, all safety precautions must be made to avoid any sort of accident!

FIRMWARE DESIGN

I developed the NTPC's firmware entirely in the microcontroller's assembly language. With the price of microcontrollers constantly dropping, you may wonder why I didn't choose a more powerful device and program it in a high-level language like C. However, you should realize that when mass-producing millions of copies of an item, even cutting the cost of components by a few cents can result in a great savings. This is an incentive to use parts whose features are the most suitable for a given problem. Assembly also provides an easier grip on execution timing, which is essential for this project. Therefore, the PIC16F84A was an ideal choice in all regards.

The NTPC displays the time and date along the nixie's entire circular path, with two selectable modes. Scrolling mode is used when the printout is scrolling clockwise a couple of times per minute. Stationary mode is used when the time and date are swapped between the front and back of the display every 2 s. The latter requires an "index hole" on the NTPC circuitry, signaling the position where printout needs to begin. (Therefore, this index hole is somewhat related to those found on floppy disks.) The necessary infrared LED and photodiode were reused from the drive. The LED with the white plastic enclosure (now turned up and glued to the top of the motor control IC) can be seen in Photo 2. (Ironically, however, this LED was part of the drive's Write Enable notch detection. The actual index hole LED is still shining away on the bottom of the clock.)

The real challenge in the firmware development arose from the need to properly handle four real-time events: the passing of 1 s, the press of the clock's Control button, the moments when the display needs to change (according to the position of the rotating carousel), and finally, the index hole with the photodiode passing over the LED. These events are mostly independent, but some of them are correlated.

The first two events are handled by periodic interrupt calls, a familiar method present in virtually all embedded systems. In the PIC16F84A, the wrap-around of the 8-bit counter called Timer0 (actually, the only timer module) triggers this interrupt. In the NTPC, the Timer0 module is configured so that its interrupt hits exactly 75 times per second. Therefore, the corresponding interrupt service routine (ISR) has an easy job, determining when 1 s has passed by maintaining a 1-byte count variable. This ISR is also suitable for handling the Control button. Its period is much longer than the button's sub-millisecond mechanical bounces, but it's much shorter than the minimum human reaction time of about 100 ms. A state machine in the ISR determines the length of button presses and differentiates between two button events (Short and Long) for the purpose of setting the clock.

The background code is responsible for generating the propeller display by tracking the turn of the carousel using carefully crafted code execution delays. This routine must run in the background because the display is being generated "around the clock," no pun intended, so this functionality hogging an ISR would bring everything else to a grinding halt. The implementation uses $\pi/1{,}000$ (called point) as the atomic unit of angle measurement. Its heart and soul is the `PtDelay` subroutine that provides delays equivalent to the desired points of carousel motion. Values directly related to physical quantities are the angle for which a nixie digit is turned on and the largest angle taken up by a digit. The former (`cDIGLTP`) is an empirical value that's set as the optimum between the digit's appearance being too faint versus too smudged. The latter (`cDIGWP`) was determined from the following formula:

$$\alpha = 2\arctan\frac{w_{DIG}}{2r_{MIN}} \times \frac{1{,}000}{\pi}$$

where w_{DIG} is the width of the digits. r_{MIN} is the distance of the nixie's deepest digit from the rotation axis (both measured in the same unit). All other angles specify the propeller display's layout.

The only fly in this ointment is that the background process is constantly being disturbed by the Timer0 interrupts, threatening to compromise all timing efforts. Fortunately, however, the execution time of the Timer0 ISR is very short. It is way within the tolerance of delay accuracy, above which humans would perceive the resulting display as distorted or herky-jerky.

The last outstanding event is the passing of the index hole. The first idea that comes to mind is to make the index hole's photodiode trigger an external interrupt (or at least a silent interrupt, where the interrupt itself is disabled, but its flag is set by the microcontroller's hardware for future polling). However, measurements showed that the LED illuminates the photodiode for about 5 ms, which is two orders of magnitude longer than the ISR's worst-case execution time. Furthermore, a new index hole event never occurs before the current revolution's printout has been completed. Therefore, the index hole can be detected from the background code, simply by sampling the port input directly. (Here I leveraged the correlation between carousel rotation and index hole transit and was rewarded with simple code.)

THE WORLD'S FIRST

Creating my nixie tube propeller clock was a tremendous amount of fun. The display is not quite bright enough to be easily visible in broad daylight; however, in dim areas and at night, it offers a mesmerizing view (see Photo 4). It is also a great conversational topic.

While the firmware's resource requirements are generally well within the PIC16F84A's capabilities, the microcontroller's eight-entry-deep hardware stack was definitely a bottleneck. The length of the deepest subroutine call chain in the background code and in the ISR combined must not exceed seven. (Remember that the ISR itself is also a subroutine call.) Due to this limitation, the ISR code's `DayRoll` subroutine needed to be "de-subrutinized." (Fortunately, it was possible because the routine was called only once and was separated only for clarity.) Stack overflows are terrible bugs because they are random in nature, which makes them very hard to diagnose, and can cause a total system crash. They often remain hidden during testing and wreak total havoc out in the field. For this reason, a careful "call count" before the product release is crucially important.

Implementing the motor itself, the slip ring, and the DC/DC converter with off-the-shelf components can improve the NTPC's design. Taking AC to the spinning board makes high-voltage generation easier. In addition, the high-accuracy 60-Hz timebase of the power grid can be used (although it may not be trivial, due to slip ring noise). Ultimately, the most elegant (but also most advanced) solution would be to power the spinning board through inductive coupling, thus eliminating all rubbing components from the design. As for the firmware, the time and date format selection could be implemented as EEPROM-stored software options. While the former could be added with little effort, the latter would require significant changes in the code. ■

Peter Csaszar (csaszar@ltu.edu) is an assistant professor of electrical engineering at Lawrence Technological University in Southfield, MI. Prior to his academic appointment, he worked for Motorola as a software engineer. His areas of interest include hardware and software design for embedded systems, which he pursues as a consultant in projects with Fortune 500 companies. In academia, Peter's primary goal is to make a new generation of students fall in love with the engineering profession. For more information about Peter's NTPC, visit www.nixiana.com.

PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2007/199.

REFERENCE

[1] L. Turner, "Build a Nixie Power Supply," www.ledsales.com.au/kits/nixie_supply.pdf.

RESOURCES

B. Blick, "Popeller Clock," http://bobblick.com/techref/projects/propclock/propclock.html.

NEONIXIE-L, http://groups.yahoo.com/group/NEONIXIE-L/.

Nixie tube information, Sphere Research, www.sphere.bc.ca/test/nixies.html.

D. Wächter, Nixie World, www.tube-tester.com/sites/nixie/nixie-tubes.htm.

G. F. Weston, *Cold Cathode Glow Discharge Tubes*, ILIFFE Books, 1968.