TAPR

PACKET STATUS REGISTER

PSR #85 AUTUMN 2002

president's corner

Open Source Hardware

BY JOHN ACKERMANN, N8UR

The Digital Communications Conference was a great success, and the presentations were more than usually thought-provoking. I came away with the strong sense that this year's DCC attendees got a very good glimpse into the future of ham radio.

I found Bruce Perens', K6BP, keynote presentation particularly challenging. Bruce talked about the history and philosophy of the open source software movement and he did a great job evangelizing the cause — TAPR has supported open source for quite a while now, and Bruce certainly strengthened our commitment to it.

But then Bruce took a surprising turn. He asked the question: why not open source *hardware*, too?

He described the problems that challenge hardware design: proprietary interfaces, high cost of low-volume parts, and short product life cycles. That last point is one that TAPR knows only too well; it's perhaps the primary factor that torpedoed the FHSS radio project.

It's easy to complain about how difficult hardware projects are, but Bruce actually had an example of folks who are doing something about it. He told us about OpenCores (**www.opencores.org**), which is the gathering place for a bunch of silicon hackers who are designing logic "cores" for field programmable gate arrays (FPGAs) and making them available to all comers under the GNU General Public License.

I need to step back a bit to explain why this is a really big deal. Of all the deep magic in the electronics world, designing ICs is about the deepest. Not only does it require lots of specialized knowledge, the software tools used to

Continued on page 3

in this issue...

Holiday Office Closure

The TAPR Office will be closed from December 20, 2002 through January 5, 2003 for vacation and inventory.

Packet Status Register

#85 Autumn 2002, ISSN: 1052-3626
Published by
Tucson Amateur Packet Radio Corp.
8987-309 East Tanque Verde Road #337
Tucson, AZ 95749-9399 USA
phone 972-671-TAPR (8277)
fax: 972-671-8716
URL **www.tapr.org**

TAPR Office Hours

Monday – Thursday, 9 AM – 5 PM Central Time
Office Closed: Dec. 20 – Jan. 5 (for vacation and inventory)

Packet Status Register

Opinions expressed are those of the authors and not necessarily those of TAPR, the TAPR Board of Directors, TAPR Officers, or the Editor. Acceptance of advertising does not constitute endorsement by TAPR, of the products advertised.

Postmaster: Send address changes to TAPR, P. O. Box 852754, Richardson, TX 75085-2754.

Packet Status Register is published quarterly by Tucson Amateur Packet Radio Corporation, 8987-309 East Tanque Verde Road #337, Tucson, Arizona 95749-9399 USA. Membership in Tucson Amateur Packet Radio Corporation, including a subscription to *Packet Status Register*, is $20.00 per year in the US and possessions, of which $12.00 is allocated to *Packet Status Register*. Membership is $20.00 in Canada and Mexico, and $25.00 elsewhere, payable in US funds. Membership and a subscription to *Packet Status Register* cannot be separated. Periodical postage paid at Richardson, Texas USA.

TAPR Officers:

TAPR Board of Directors:

Tucson Amateur Packet Radio is a not-for-profit scientific research and development corporation [Section 501(c)(3) of the US tax code]. Contributions are deductible to the extent allowed by US tax laws. Tucson Amateur Packet Radio is chartered in the State of Arizona for the purpose of designing and developing new systems for digital radio communication in the Amateur Radio Service, and for disseminating information required, during, and obtained from such research.

Submission Guidelines

TAPR is always interested in receiving information and articles for publication. If you have an idea for an article you would like to see, or you, or someone you know, is doing something that would interest digital communications, please contact the editor (**psr@tapr.org**) so that your work can be shared with the Amateur Radio community. If you feel uncomfortable or otherwise unable to write an article yourself, please contact the editor for assistance. Preferred format for articles is plain ASCII text (Microsoft Word is acceptable, please save the document as Rich Text Format - .RTF). Preferred graphic formats are PS/EPS/TIFF (diagrams, black and white photographs), or TIFF/JPEG/GIF (color photographs). Please submit graphics at a minimum of 300 DPI.

Production / Distribution:

Packet Status Register is exported as Adobe *Acrobat* version 5 and distributed electronically at **www.tapr.org**

PSR *Packet Status Register* Editor:
Stan Horzepa, WA1LOU
One Glen Avenue, Wolcott, CT 06716-1442 USA
phone 203-879-1348
e-mail **wa1lou@tapr.org**

Continued from page 1

design chips cost tens, if not hundreds of thousands of dollars. Then, of course, there's the small matter of putting sub-micron sized bits of silicon and metal onto a wafer and turning it into a finished chip. Homebrew ICs are something of a rarity.

Chips that are designed for specialized uses rather than generic building blocks are called Application Specific ICs, or "ASICs". In order to be efficient (and profitable), ASIC designers rely on libraries of standard building blocks, called "cores." A core might be as simple as a small block or RAM, or as complex as a complete CPU. Most ASICs consist of a bunch of cores plus some unique circuitry linking them together. Because this is a competitive business, ASIC houses build their own libraries of proprietary cores and guard them with their lives.

In short, if you want to develop an ASIC, you'd better be ready to pay one of the ASIC houses some really big bucks.

In the last few years, FPGAs have started to change the chip market. An FPGA is a generic chip that has lots of logic gates. Make that *lots and lots* of logic gates; FPGAs are available today with more than 1 million gates. FPGAs are programmed by the user; you load a program into the FPGA that tells it how to interconnect all its logic gates. The program defines the hardware.

FPGAs can take the place of moderately-sized ASICs. In fact, the FHSS project team handled the discontinuation of one ASIC by replacing it with an FPGA. However, designing an FPGA program from scratch has much of the complexity of designing an ASIC; it's not a project for an evening or two, and designing all the cores you need in order to do something useful is way beyond the capabilities of most individuals. By itself, the FPGA is a big step forward, but it doesn't make every ham a potential chip designer.

The OpenCores folks recognized that FPGAs allow for a whole new way to pursue hardware development. Instead of jealously hoarding cores so that everyone has to start from scratch, why not follow the lead of the free software community and develop a set of open source cores that are free for anyone to use?

They've done just that. If you visit **www.opencores.org/projects**, you'll find dozens of finished cores, ranging from Ethernet controllers and UARTs to complete microprocessor cores. I'm no chip designer, but it looks to me like the stuff that's available right now could be put to use in some interesting ham projects.

One downside of the open source IC idea is that at present, the FPGA manufacturers have a lock on the programming tools and they haven't released enough data to allow open source software implementations. Hopefully, this situation will improve with time. In the meantime, there seems to be enough folks with access to the tools to provide a very good start on the concept.

Any ham working on a digital communications project ought to take a look at OpenCores and we at TAPR will be watching this project closely, helping it where we can, and rooting for it to succeed.

New Faces

This year's TAPR elections brought two new faces to the Board of Directors. Stan Horzepa, WA1LOU, and Darryl Smith, VK2TDS, have been elected to fill the seats vacated by Bob Hansen, N2GDE, and Steve Dimse, K4HG, both of whom chose not to stand for re-election. Steve Bible, N7HPR, was re-elected to the Board and continues as TAPR's Vice President.

Both Bob and Steve D. have been valuable members of the Board and we will miss them; fortunately, both have indicated a willingness to stay involved in the organization. Bob was the tireless editor of *PSR* for many years and Steve was not only an interface to TAPR's APRS constituency, but also an involved Board member who really helped push TAPR along. I'm sure I join all TAPR's members in thanking them for their hard work and involvement.

Many of you know Stan Horzepa. He's an author and *QST* columnist; he presides as moderator over the

Continued on page 4

Continued from page 3

madhouse known as the aprssig mailing list, and he's been editor of *PSR* for a few issues now. I know Stan will make sure that the Board keeps in close touch with the APRS community. Since he's a glutton for punishment, Stan will continue to serve as *PSR* Editor.

Darryl Smith is our first intercontinental Board member. He's been very active in APRS in Australia, but he also has a wide interest in other facets of digital radio. He's nothing if not enthusiastic and I'm looking forward to having an international voice on the Board.

I'm also asking Darryl to take on the role of Membership Director. Frankly, we haven't done a very good job of selling ourselves to the ham public and lots of folks who should be TAPR members aren't, mainly because they haven't been asked. Darryl will be responsible for boosting TAPR's membership both internationally and in the US.

I'm sure that Stan (**wa1lou@tapr.org**) and Darryl (**vk2tds@tapr.org**) would like to hear your comments and suggestions about TAPR.

TAPR and *Linux*

Continuing with the open source theme of this column, I'd like to spend a minute on the role of free software and *Linux* in particular, within TAPR. As you may have guessed, we strongly support the use of open source software in ham radio projects and we think Linux is a great platform on which to build ham applications. In fact, we endorse *Linux* as the preferred environment for TAPR software projects.

There are already many ham applications available under *Linux* including several that are every bit as sophisticated as their *Windows* equivalents. *Xastir* is probably the most rapidly developing APRS application and it is truly amazing. I've played with it a bit and its capabilities, particularly the variety of maps it can use, are tremendous.

A second program that is pretty slick is *kpsk*, a PSK-31 application written to run under the KDE desktop environment. It has good functionality, a great interface, and it works reliably. As I write this, I've only been using *kpsk* for a few days, but I haven't found anything yet that makes me want to go back to any of the *Windows* PSK programs.

Of course, there are many more *Linux* ham applications than these; just check out **http://radio.linux.org.au** if you need proof.

There's been some discussion about which *Linux* distribution is best for ham use. That's a tough question, because the various distributions are tuned for different audiences and "ham" doesn't fit neatly into any of them.

All of the major distributions — Red Hat, SUSe, Mandrake, Debian, etc — work well and you won't go wrong picking any of them.

However, I have to admit a bias toward Debian for three reasons. First, Debian is a volunteer effort that is driven by the open source concept. Philosophically, I like that. Second, Debian has had a strong ham involvement since its inception. Bruce Perens, mentioned above, was one of the early Debian contributors, and Bdale Garbee, KB0G, is the current Debian Project Leader. Third, Debian supports more ham radio applications than any of the other distributions. In fact, it's the only distribution to have a separate "ham radio" category in its installer. Most of the software you'd want to use in your ham station has been packaged for Debian.

On the down side, Debian's installation program is a bit daunting for the newcomer and its dedication to stability means that you often won't find the latest program versions in the standard distribution. It also lacks the graphical configuration tools of some distributions. In short, it's not the easiest version of *Linux* to install or configure (but it's certainly not the mess that some people have made it out t be).

TAPR believes that *Linux*, in any of its flavors, is a "good thing" and we encourage developers to consider it as a platform for any ham radio project. For the reasons described above, we have a fondness for Debian and suggest that it's worth at least a look if you're choosing a *Linux* distribution.

If you're using *Linux* in your ham shack, why not write an article about it for PSR? Help spread the word!

73, John

2002 ARRL/TAPR Conference

More than 100 of the most active Amateur Radio digital enthusiasts from around the world turned out in Denver, Colorado, September 13-15 for the 2002 ARRL/TAPR Digital Communications Conference. This year's event marked the 21st conference. Agenda topics ranged from APRS (Automatic Position Reporting System) to high-speed digital networking and software-defined radio (SDR), among others.

Friday's forums were dominated by discussions of APRS. Topics included a discussion of single-wire APRS weather stations, high-altitude balloon tracking and recovery—presented by representatives from Edge of Space Sciences (**www.eoss.org/**)—APRS in the Sydney Olympics, and the versatile Findu.com (**www.findu.com/**) on-line APRS database.

Saturday's sessions included forums on the prospect of using consumer wireless devices (popularly known as 802.11b or "Wi-Fi" devices) to create high-speed Amateur Radio digital networks. A forum on HF digital voice also drew considerable interest.

One of Saturday's highlights was a demonstration of the new ICOM D-Star (**www.tapr.org/tapr/dv/DStar brochure.pdf**) digital radio system. At the heart of D-Star is the ID-1 transceiver, which ICOM had on display at the Dayton Hamvention last spring. The ID-1 operates on 1.2 GHz and can communicate using FM analog voice, digital voice, and data. The transceiver can be programmed with a desktop or laptop computer, or it can be operated in a more conventional manner via a remote front panel. ICOM's Ray Novak, KC7JPA, said D-Star will be available in the US in November.

Bruce Perens, K6BP, (**http://perens.com/**) was the featured speaker at the Saturday evening banquet. His entertaining presentation stressed the notion that individuals, not just corporations, still can innovate and invent. Perens called for grassroots development of Amateur Radio software and hardware according to the Open Source model. He also encouraged the audience to become educators, because, he explained, "the future strength of Amateur Radio is in our value as technology teachers."

Continued on page 6

OVER 100 PEOPLE ATTENDED THE MAIN TECHNICAL PRESENTATIONS.

BESIDES ALL THE TECHNICAL TALKS AT THE DCC, THERE WAS ALSO A DEMONSTRATION ROOM WHERE PEOPLE BROUGHT THEIR PROJECTS TO SHOW AND TELL. THERE WAS ALSO A FLIP-CHART TO ENCOURAGE IMPROMPTU DISCUSSIONS.

Continued from page 5

SDR was another hot topic at the conference and the Sunday seminar was devoted exclusively to that subject. Projects such as GNU Radio (**www.gnu.org/software/gnuradio/gnuradio.html**) promise a day when amateur transceivers will achieve extraordinary levels of flexibility. Under the SDR paradigm, software, rather than the hardware, literally will "define" the way in which a radio operates.

Proceedings of the 21st ARRL and TAPR Digital Communications Conference are now available from TAPR and ARRL. The contents of the proceedings follow:

"1-Wire APRS Weather Station" by William Beals, N0XGA

"PIC-EII" by Richard E. Carter, KE1EV

"APRSWXNET/CWOP—A Beneficial Partnership of NOAA, Amateur Radio and Other Good Citizens" by Russ Chadwick, KB0TVJ

"Automatic Packet Reporting System: Building a Large Scale Geospatial Database" by James Jefferson Jarvis, KB0THN

"Created Realities Technology in Amateur Radio" by Greg Jones, WD5IVD

"Repeater Data Transmission System" by Peter Mudie, VK2XZP

"On Amateur Radio Use of IEEE 802.11b Radio Local Area Networks" by Paul Rinaldo, W4RI, and John J. Champa, K8OCL

"Emergency Radio Email (ER-Email)" by Paul Schreiber, W2UH

"801.11 and Ham Radio" by Darryl Smith, VK2TDS

"A Software-Defined Radio for the Masses, Parts 1 and 2" by Gerald Youngblood, AC5OG

– adapted from *The ARRL Letter*

MR. BRUCE PERENS, K6BP, WAS THE BANQUET KEYNOTE SPEAKER.

MATT ETTUS N2JMI PRESENTING HIS SOFTWARE-DEFINED RADIO SEMINAR ON SUNDAY, WHICH RAN FOUR HOURS.

Check AX.25 Packets Quickly with a 0x9F0B

BY FRANK H PERKINS, JR. WB5IPM

The common way of checking for errors in a received AX.25 packet is to accumulate all the bits between the leading and trailing 0x7E framing characters (with any “padding” zero bits removed), and then run a frame check sequence (FCS) calculation on these bits, which include the transmitted FCS bits. If the packet is valid, the result of the FCS calculation is the well-know 0xF0B8 “magic number.”

There is often plenty of processor time available between the reception of each packet bit (especially at 300 and 1200 bit/s) to do a running bit-by-bit FCS calculation. However, you have the problem of not knowing you have run the bit-by-bit FCS calculation into the trailing framing character until you are six bits into it, where you realize there is no padding zero following the preceding five one bits.

However, if all the bits in the packet have been valid when you reach the first bit of the trailing framing character, the FCS calculation will be holding 0xF0B8. If you run the FCS calculation on through the 0x7E trailing framing character, 0xF0B8 morphs into 0x9F0B. Since it is easy to detect a trailing framing character in a simple one-byte correlator, you can run the FCS calculation bit-by-bit until the last bit in the trailing framing character sets the correlator test to true. If your FCS calculation is holding 0x9F0B at this point, the packet is valid.

In the example code fragment below, nxtbit is the next bit with padding zeros removed. Detection of the leading framing character is assumed to be done by upstream code:

```
const byte FC = 126;                // 0x7E framing character value
const unsigned short POLY = 33800;  // 0x8408 AX.25 FCS polynomial value
const unsigned short OK = 40715;    // 0x9F0B FCS magic number with frame
byte nxtbit = 0;                    // nxtbit holds the next bit (1 or 0)
byte fdc = 0;                       // framing character detection correlator
unsigned short fcs = 65536;         // FCS buffer seeded with 0xFFFF
.
.
.
fdc = fdc >> 1;                     // shift bits in correlator right
if(nxtbit == 1)                     // if next bit is a 1
{
  fdc = fdc | 128;                  // OR 1 into correlator MSB
}
if(((fcs & 1) ^
  (unsigned short)ntxbit) == 1)     // XOR LSB in FCS buffer with next bit
{
  fcs = fcs >> 1;                   // if 1 shift bits in FCS buffer right
  fcs = (fcs ^ POLY);               // and XOR FCS with AX.25 polynomial
}
else
{
  fcs = fcs >> 1;                   // else just shift bits right
}
if(fdc == FC)                       // if trailing framing character detected
{
  if (fcs == OK)                    // and FCS calculation is 0x9F0B
  {
    // packet FCS is valid, display packet and reseed fcs, etc.
  }
}
.
.
.
```

A 2.4-GHz Yagi Antenna

BY DON ROTOLO, N2IRZ

In the last issue of the *PSR*, I wrote about some findings from when I disassembled a Linksys 802.11b wireless LAN adapter. I noted in that article that I still needed an antenna. In a search of the Internet, I didn't find any useful instructions for building a reasonably high gain antenna for 2.4 GHz. Yes, there is the famous Pringle's can antenna, but I wasn't terribly impressed by it. I'd considered scaling down a 1.2 GHz loop Yagi design I saw in the *Handbook*, but hesitated because of the complexity. Down East Microwave sells a nice 2.4 GHz loop Yagi, reasonably priced too, but I never seemed to have the spare cash. So, the project sat in limbo for a few months.

Out of the blue, a friend of a friend contacted me, having bought some gear from me last year at a hamfest. I had some more stuff he'd find useful for a project and it turned out that he happened to have a nice 2.4 GHz Yagi he'd be willing to trade. After a few days, I was the proud owner of a used commercial 2.4 GHz Yagi designed just for 802.11b gear and rated at a healthy 13.5 dBi. You can see it in Figure 1.

Of course, I then did what everyone does with a new piece of equipment, before even testing it to see if it would work: I took it apart. Remembering my lack of luck in finding a decent antenna or plans for same, I decided that I would again share what I have learned.

The Aironet WLAN antenna, model AIR-ANT1949, is designed for use with Cisco equipment. Details on the antenna can be found at (**www.cisco.com/**); search on Aironet Antenna. It is a compact unit housed in a plastic radome. It comes equipped with about three feet of RG-58U terminated with a reverse-sex TNC connector.

Before I get started, you should understand very clearly that modifying this antenna in any way, such as changing the connector end, invalidates its FCC approval for use with any 802.11b equipment. That means that (in my opinion and probably under FCC regulations) it is illegal to modify this antenna or use it with any 802.11b equipment other than for which it is designed and sold. That's the reason for the reverse sex connector; using an uncommon connector helps ensure the antenna is not connected to equipment for which it was not intended. If you buy or build one and use it under Part 97, like I plan on doing, then there's no problem.

Opening up the radome revealed a very simple antenna, one which might be duplicated by other amateurs for their own use. Figure 2 shows the antenna removed from the radome. It consists of a 16-element Yagi stamped from a single piece of 1/16" thick aluminum about 16 in. long and 2-1/2 in. at its widest with a small piece of PC Board as the driven element.

Continued on page 10

TABLE: DIRECTOR DIMENSIONS AND POSITIONING. ALL DIMENSIONS IN INCHES. SEE TEXT FOR DRIVEN ELEMENT AND REFLECTOR DIMENSIONS.

Director	Width overall	Distance from the front of the DE to the rear of the element
1	2-1/8	0.75
2	2-1/16	1.75
3	1-31/32	2-13/16
4	1-15/16	3-15/16
5	1-7/8	5
6	1-7/8	6-1/16
7	1-27/32	7-1/8
8	1-27/32	8-3/16
9	1-25/32	9-1/4
10	1-13/16	10-5/16
11	1-25/32	11-3/8
12	1-25/32	12-7/16
13	1-15/16	13-1/2
14	1-25/32	14-9/16

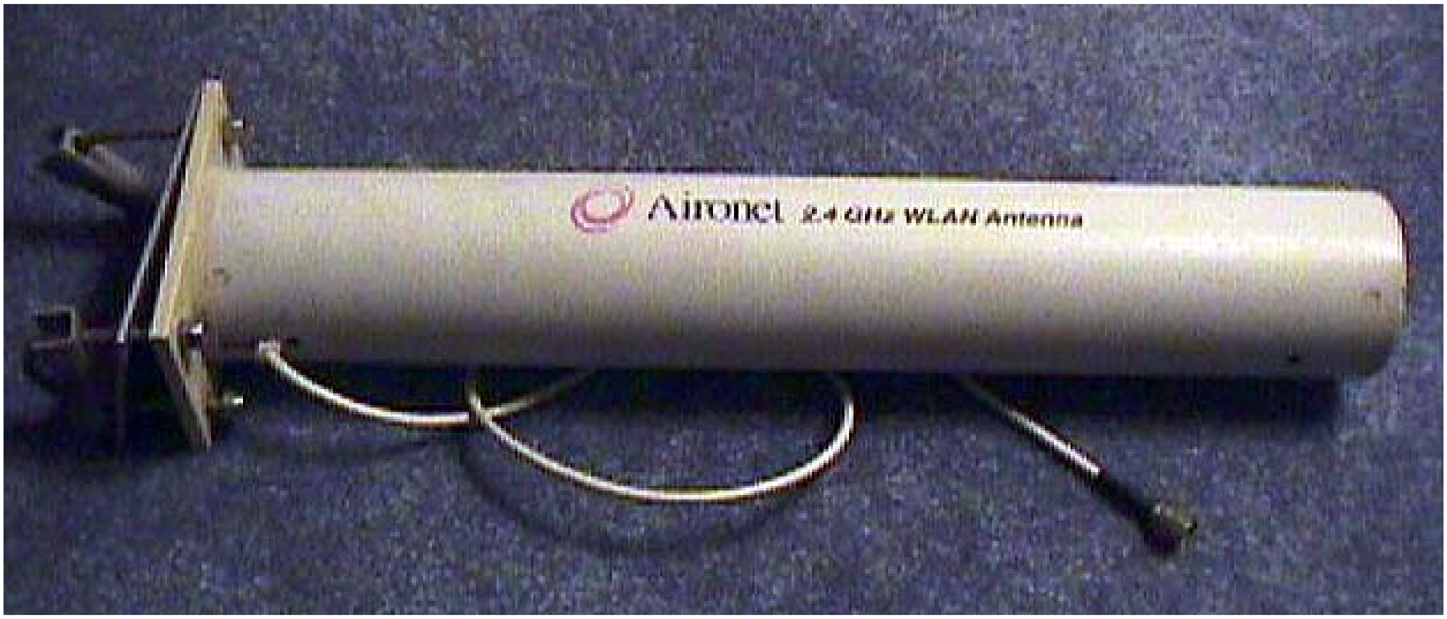


FIGURE 1: THE AIRONET 2.4-GHZ WLAN ANTENNA INSIDE ITS HOUSING.

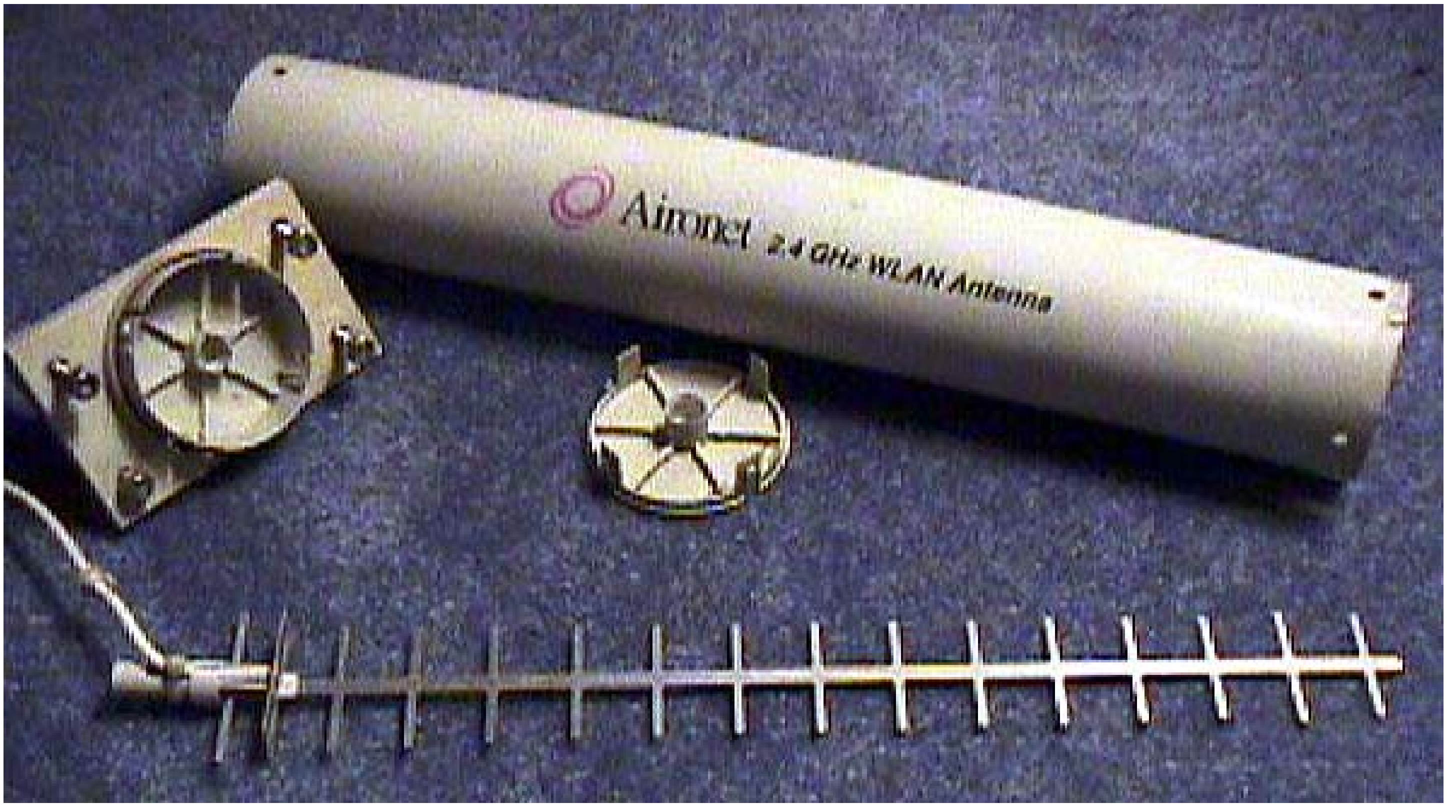


FIGURE 2: THE ANTENNA REMOVED FROM THE HOUSING. IT IS STAMPED FROM A SINGLE PIECE OF 1/16-INCH ALUMINUM SHEET.

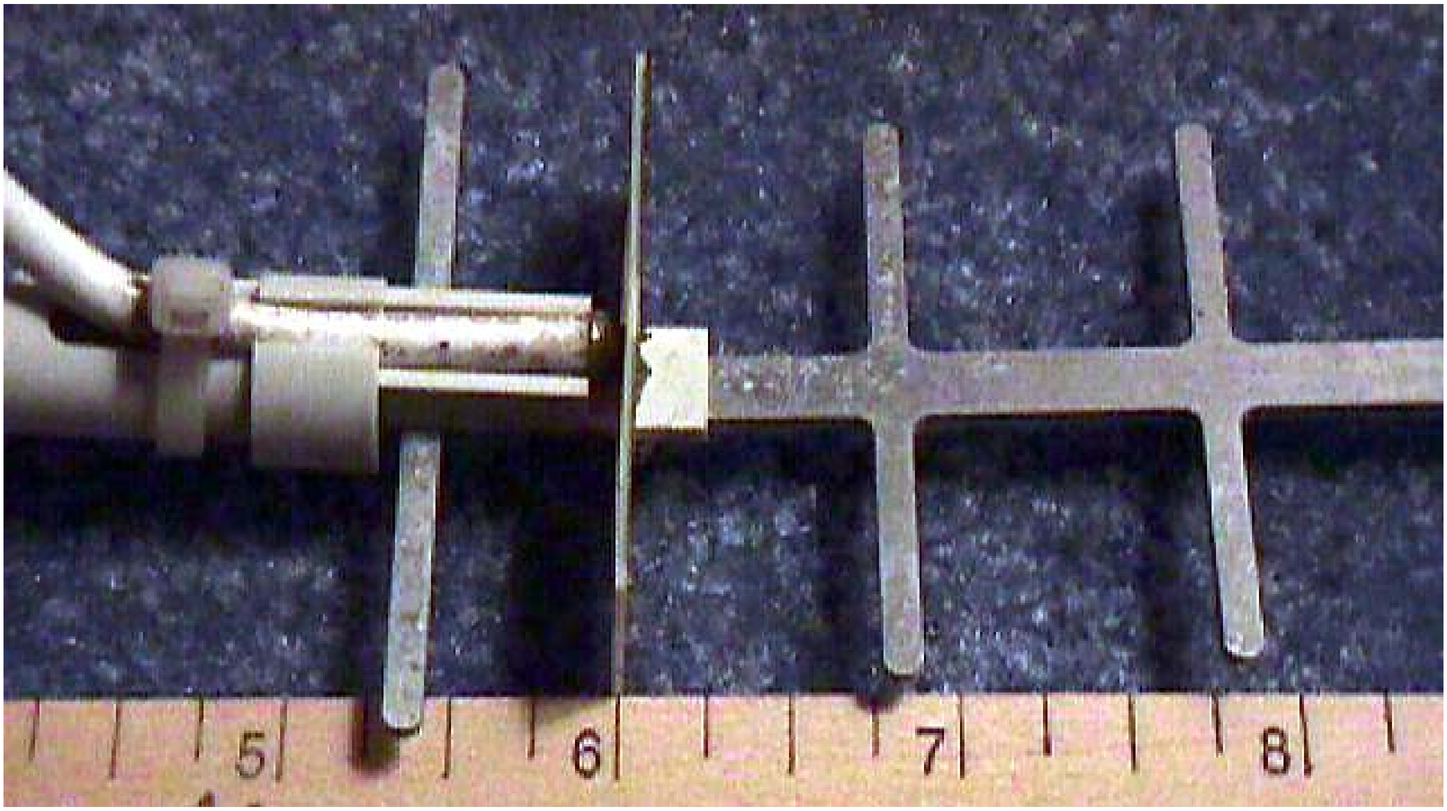


FIGURE 3: CLOSER VIEW OF THE DRIVEN ELEMENT END OF THE ANTENNA. THE DE IS FABRICATED FROM THIN PC BOARD MATERIAL

Figure 3 shows a close-up of the driven element end of the antenna. Each of the passive elements (14 directors and one reflector) is 1/8 in. across (front to rear) and the widths and distances from the driven element (DE) are shown in the table below. The reflector is 2-1/2 in. wide and the front edge is positioned 9/16 in. from the front of the driven element. Refer to Figure 4 for the naming of the elements. The center boom is 1/4 in. wide.

If you were manufacturing thousands of these antennas, then creating a tool to stamp them out would be worth it. For the rest of us, I don't see a really easy way of fabricating one or two. Since I have one already, I'm not that motivated to figure out a good method. Nonetheless, perhaps some kind of punch mounted to an arbor press, kind of like a giant nibbling tool, might work. Using copper instead of aluminum would allow an antenna to be soldered together out of individual strips. You could also use a band saw or similar to cut one out of sheet aluminum. If you made a full-size pattern on the computer and cut it out, I'd guess it would take an evening's work. Add another evening etching the PC Board driven element and assembling it into an antenna. Two evening's work seems like a reasonable investment for such an antenna.

Figure 5 shows the etching dimensions of the Driven Element (DE) as best I can measure with a micrometer and magnifying glass. The DE is made from a piece of cream-colored, ceramic-like PC Board material (perhaps Rogers Duroid) 1/32-in. thick. It is exactly 2 inches wide and 0.780 inches high. There is a 1/16-in. slot machined into the element used to allow the DE to be slid into position on the solid boom between the reflector and first director. Once in position, there is another slot, 1/4 in. x 1/16 in., into which the boom rests. A 1/16-in. thick piece of plastic helps position and lock in the DE relative to the reflector and also serves to support the coaxial feed line. It seems to me that one could use ordinary glass-epoxy material if you are willing to sacrifice some efficiency and gain.

Continued on page 12

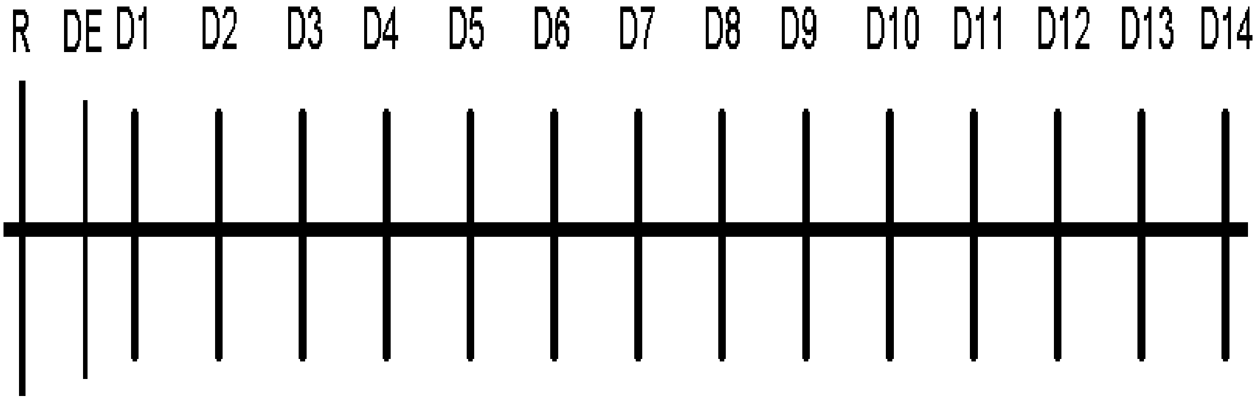


FIGURE 4: ELEMENT NAME DESIGNATIONS

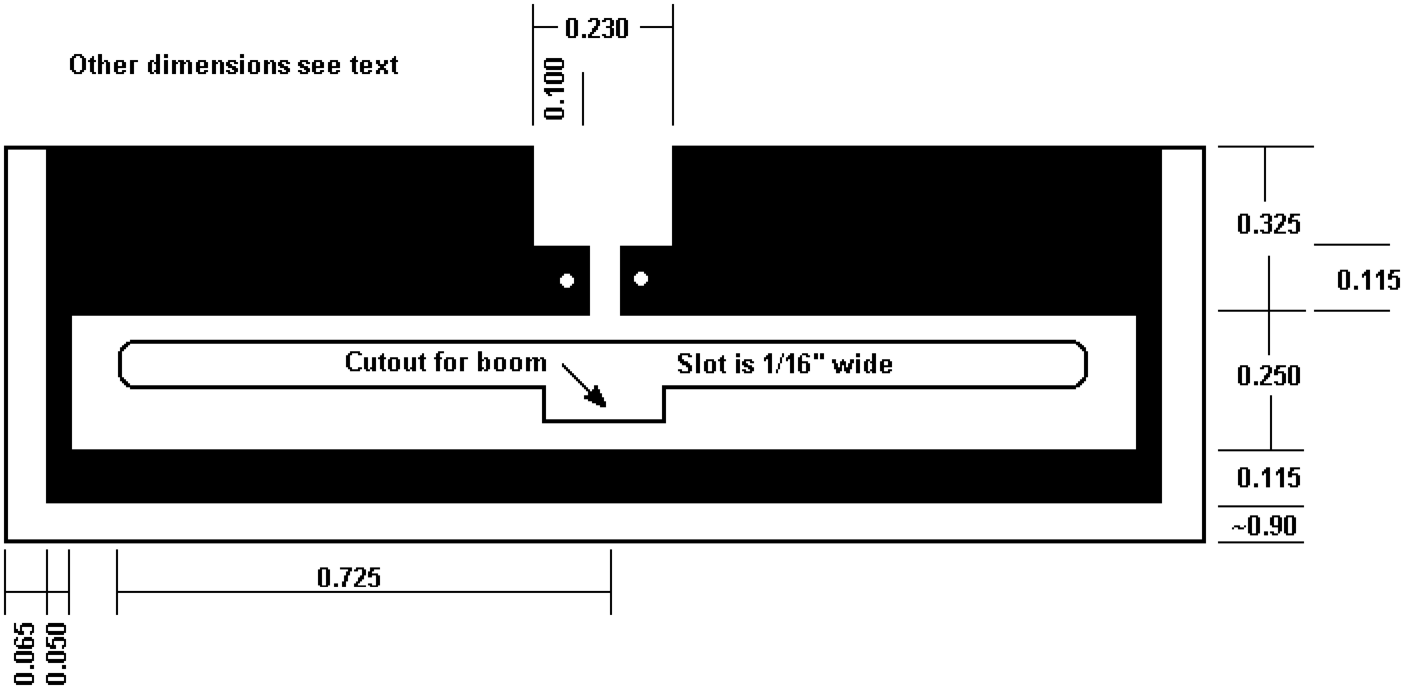


FIGURE 5: DRIVEN ELEMENT DIMENSIONS. SEE TEXT FOR DETAILS.

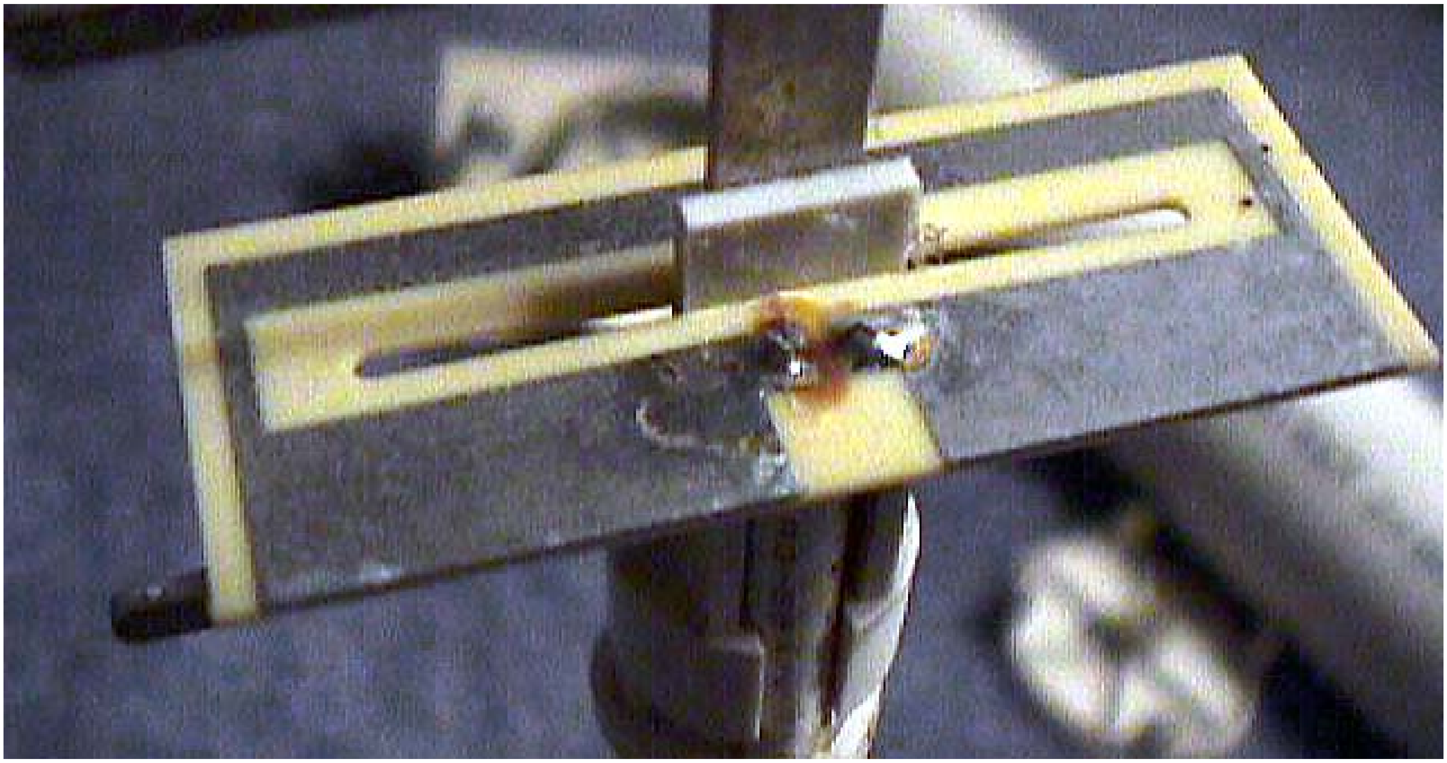


FIGURE 6: DETAIL OF THE FEED LINE CONNECTION TO THE DRIVEN ELEMENT.

Continued from page 10

The feed line is soldered into the driven element as shown in Figure 6. It should not matter which point is used for the center conductor and which is used for the shield. In the photo, you can also see the piece of plastic used for positioning and imagine how the metal boom fits into the slot behind the plastic piece.

That's all there is to it! For a mass-produced antenna, it is a very clever design which can be duplicated by the thousand for just a few dollars in materials. While a bit more tedious for homebrew, it is still within reason for both complexity and effort, especially when considering the gain. Although I haven't yet had a chance to put my new toy on the air, rest assured that it won't be much longer before that happens.

It is my sincere hope that this brief article gives you the information you need to build an antenna like this for yourself. I also hope that everyone reading these words takes a moment to consider what knowledge that you have and that you will take a moment and share that knowledge with the rest of us.

My Little Near Space

BY L. PAUL VERHAGE, KD4STH, **paul.verhage@boiseschools.org**

I received my Amateur Radio license (KD4STH) in August 1992. I was excited to make my first 2-meter contact through one of the local repeaters in Alexandria, VA, but not long after, the novelty wore out. Then packet radio was my primary interest, but there wasn't much for me to do with packet at the time. At the same time, GPS receivers were becoming available. They sounded like fun, so I purchased a Garmin GPS-45 as soon as the price dropped below $200. My other interest at the time was microcontrollers and their application in things like robotics. I discovered them with my other interest at the time: the Internet. It was a shame I couldn't combine all these interests with one of my life long interests, space (I hadn't even considered that possibility, at the time).

That was until I meet Pete Sias, WB0DRL. Pete was invited to present at the October 1994 meeting of my Amateur Radio club, the Manhattan Area Amateur Radio Society (MAARS). With him, he brought the most amazing project I had every seen. Pete was building functioning models of satellites and launching them to extremely high altitudes. From there, the capsules returned some of the most fantastic photographs. Until then, I had only seen photographs like those when astronauts took them. Pete's project involved packet radio, GPS, microcontrollers, and space. It was the ultimate combination of my four greatest interests. It was the program for all wannabe astronauts, including me.

It took me two years to get my program together and launch my first capsule. On 2 November 1996, with the help of MAARS and Pete, I launched my first near space capsule. On this inaugural flight, I sent the following experiments into near space.

Geiger counter

Two 35-mm cameras

Two toy gliders

Petri dish sampler

Two types of bubble wrap samples

Air temperature sensor

To construct the near space capsule, I built an airframe, avionics, and recovery parachute. The sole avionics consisted of a flight computer based on the Basic Stamp 2-IC. I designed the PCB of the flight computer on a sheet of graph paper with a permanent ink marker and then copied the artwork to a copper clad circuit board using the iron-on transfer method.

Now that I look back on it, I'm amazed the copper traces of the PCB were good enough to make the flight computer function. The transfer was so poor that I was required to make several repairs with wire bridges.

A Motorola OnCore VP GPS receiver onboard the capsule sent positional data to the Basic Stamp and the KPC 3+ v.6 TNC. Back in early 1996, it was difficult to get a GPS receiver capable of sending positional data above 60,000 feet. Synergy, the Motorola distributor I purchased the receiver from, required that I sign a contract stating I wouldn't sell the receiver to the US government. GPS companies were a bit paranoid, I guess, about the DoD and State Department limitations on GPS receivers.

During a mission, the Basic Stamp was pro-

Continued on page 14

Continued from page 13

grammed to send commands to the GPS receiver telling it when to send specific GPS sentences. Data from the GPS receiver and Basic Stamp were mixed into a single wire by a Basic Stamp controlled AND gate and then sent to a MAX232 TTL to RS232 converter. The result was sent to the KPC 3+ for transmission to ground stations.

The KPC3+ was put into transparent mode before launch so it would transmit any text it received. The flight computer didn't ensure the KPC3+ stayed in transparent mode, it just assumed it would stay in transparent mode. Before its maiden launch, the capsule and its avionics were tested on the ground for several hours. The capsule was even tested inside a walk-in freezer!

The first launch attempt failed. He winds were way too strong and snapped the balloon off of the load line, leaving the capsule in my hands as I watched my $150 balloon rocket away.

The next week, Saturday, 2 November 1996, was a cold but windless morning. The bright sun shone in a crisp blue autumn sky. A new balloon was filled inside a large building and carried outside for launch. Then the MAARS crew carefully raised the balloon on its load line and then the recovery parachute. All during this time I was holding my near space capsule. I would be the person to let the balloon, parachute, and capsule become independent. I'll never forget the sensation. When I released my grip, my capsule lifted gently out of my hands. It was incredible. I watched the stack (balloon, parachute, and capsule) rise skyward, all the time growing smaller. I've launched 35 more capsules since them, but probably have never watched one from the ground for as long. After the launch, it was chase time.

At an altitude of about 19,000 feet, the capsule stopped sending telemetry to the chase crews. My capsule, the Isaac Asimov, was now lost in near space. Fortunately, one of the chase crews experienced in packet determined he could connect to the Isaac Asimov and get about 15 seconds of data from it before the TNC timed-out. For the rest of the flight, he pressed the ENTER key every ten second ensuring we received science and positional data from the capsule. I have since forgotten his name now, but he saved the mission.

The Isaac Asimov reached an altitude of 90,200 feet before the balloon burst. While the winds at ground level and 90,000 feet were gentle, the winds in the jet stream were traveling at 120 mph. The capsule passed through these winds twice, once on ascent and once again during descent. These high-speed winds pushed the capsule way ahead of the chase crews.

Compounding our difficulties was the fact that the winds were carrying the Isaac Asimov towards the southeast, while our roads were primarily aligned either east-west or north-south. At about 30,000 feet altitude on the way down, the capsule was so far ahead of us that we could no longer maintain contact with it. So the TNC timed-out and stopped sending data. Again, we had lost the near space capsule.

We spent hours searching for the capsule. Our only hope was to get close enough to the capsule to make a packet contact with it. With a single

Continued on page 15

Continued from page 14

GPS position, we would be able to find the capsule. With the help of a pilot and hams in Pittsburg, KS, we finally made that contact and were able to locate the capsule. It had flown 166 miles from the launch site and landed some five hours before we found it. But it wasn't over yet.

The capsule landed in one of the few trees in the area and the tree was too high to climb. Eventually the landowner showed up and cut the tree down for us. I didn't get back home until 10 PM that night. It had been a long day in which I almost lost my first near space capsule (valued at about $700) twice. But by golly, I was addicted!

Since then, I have redesigned my flight computers (thanks to Steve Kelly) and capsule airframes (thanks to Jon Held). I have flown 35 more flights as of the end of September 2002. My capsules have reached altitudes between 48,000 feet and 114,600 feet. Over one dozen individuals have been instrumental in helping me launch and recovery near space capsules. Thanks, guys and gals.

Currently I have several near space projects on the burners. First is to finish writing a book on amateur near space programs to explain how to create and manage a program of your own. I spent a lot of time banging my head on the wall, trying to figure how to make things function as they should. This book will keep others from having to solve the same problems I did. I plan to title the book, *The Near Space Files*. Look for it from Parallax Inc. in the near future.

My second project is experimenting with the MIM as a near space TNC. More work has yet to be done, but I have discovered the following facts about interfacing it to a microcontroller. The MIM has a limit of 65 characters. The 66th and later characters sent to it are ignored. Any text can be sent to the MIM as long as you program the character pattern string for the data the MIM is to look for. The MIM and its programmer do not verify the character pattern string is a valid GPS sentence. Any sentence going to the MIM must begin with a $ and end with a *. I also discovered that the BS2p, a version of the Basic Stamp 2 with a large scratch pad RAM, can GET bytes from its scratch pad RAM and send them one at a time as serial data to the MIM. But I've not had the success I want receiving all the packets from the MIM. So more work remains.

Before long I will begin my third project, designing a flight computer that uses one of Maxim-IC's newer voltage regulators. The LM2940T-5 that I currently use is a low dropout voltage regulator. The low dropout voltage allows flight computers to operate from a six-volt flight battery, keeping battery weight low. However, it doesn't allow the capsule to use all the capacity of the flight battery.

There are many Web sites dedicated to amateur near space exploration. Please check them out. We're all happy to answer questions you may have about starting your own program of amateur near space exploration. Join us; the air is rare in near space. Some amateur near space groups I've been involved with are:

Arizona Near Space Research (ANSR) **www.ansr.org**

Edge of Space Sciences (EOSS) **www.eoss.org**

High Altitude, Basic Investigation And Testing (HABITAT) **http://habitat.netlab.org/index.shtml**

Kansas Near Space Project (KNSP) **www.ksu.edu/humec/knsp/**

Nebraska Stratospheric Amateur Radio (NSTAR **http://members.cox.net/mconner1/nstar.html**

Project Traveler **www.rckara.org/project-traveler/**

Treasure Valley Near Space Program (TVNSP) **www.tvnsp.org**

Emergency Radio E-mail (ER-Email)

BY PAUL SCHREIBER, W2UH

This description of ER-Email is submitted to the *Packet Status Register* to solicit from this pool of experts (1) a realistic assessment of its potential as an important emergency communications mode, (2) the effort required to develop it, and (3) what group or individuals are willing to develop it.

I feel very strongly that Emergency Radio-Email has the potential of becoming an important ARES/RACES emergency communications mode with a very high degree of acceptance among served agencies. The name was chosen to quickly convey to served agencies what ER-Email will do for them.

Acceptance

Widespread use of Internet messaging plus a review of the acceptance of packet by our served agencies in drills since 1988 convinces me that there will be less "keyboard fright' than mike fright among the personal we serve. They are no longer surprised by errorless text messages, *they will expect them*. Why else does AOL feature instant messages in its TV ads?

Advantages

Served agencies will highly value these features of ER-Email:

- A high degree of security, especially Red Cross Chapters.
- Errorless text messages.
- Hard copies of these messages.
- Date and time stamps, especially Offices of Emergency Management.
- All messages saved to disk

ARES/RACES emergency communicators will benefit from these features of ER-Email

- Elimination of constant monitoring as with voice nets.
- Elimination of a net control operator.
- Much less on-the-air time for each message.
- Immunity to local audible noises.
- Almost no operator training (if you do Internet email you can do ER-Email)

Bas-ic Outline of ER-Email

To minimize operator training and increase the comfort of personnel at served agencies, the most important element of ER-Email is a PC window very closely mimicking popular Internet browsers. The software includes a mailbox for incoming messages. For universal compatibility, AX.25 packet output comes from either an internal sound card or a small external TNC.

ER-Email Window

In overall appearance, the window must look and feel like e-mail windows of popular browsers. E-mail buttons and drop-down menus include **You have mail**, **Create mail, Send mail, Read mail,** and **Print mail**. Drop-down menus for **off-line mail** includes **Mail you've read, Mail you've sent,** and **Mail waiting to send**. Buttons specific to packet are **My call,** and **My alias**, and a **Settings** drop-down menu that has all the more exotic packet parameters for experienced packeteers to worry about and keep hidden for non-packeteer operators.

An **Address Book** lists other packet stations in the emergency by a full tactical description plus the shorter tactical alias used in the ER-Email address field. This allows changes in My Call with new operators. For instance, the Middle School Shelter listing has an alias MIDSCH.

After clicking on **Send email** pop-up messages indicate; **Connecting to [*], Connected to [*], Sending email to [*], Email [subject] received by [*],** or **Connection lost .** A **[*]** is full name of addressee, for instance, Middle School Shelter.

All received and sent mail is date and time stamped, and automatically saved and retrievable

Continued on page 17

Continued from page 16

using the buttons mentioned above.

Mailbox

ER-Email packets typically go to target station's mailbox, and **You have mail** button brightens. This operational mode more closely mimics Internet e-mail and eliminates need for constant operator monitoring. A **Chat** function allows keyboard-to-keyboard conversations, if desired.

AX.25 Packets

For minimum extra hardware, AX.25 packets come from an internal sound card or small external, battery-powered TNC. One of my ARES/RACES operators, Andy Stillinger, WA2DKJ, once crammed a TNC into a connector shell!

Digipeaters, not PBBSs or Network Servers

To eliminate complicating packet infrastructure, digipeaters connect out-of-range stations. Checking a PBBS for messages is eliminated. Some incidents may need a dedicated digipeater at a high location. Otherwise, stations in the net also digipeat. The Address Book includes the suitable digipeater for each target station.

Date Rate

Data rates of 1200 baud are completely acceptable. Only text messages of a less than a page are expected. One-page official damage assessment reports have been transmitted by packet at speed comparable to faxes.

Voice link

As envisioned, each packet station also has a voice link for quick tactical exchanges. The voice operator is also the control operator for the packet station. If desired, this allows an unlicensed person, preferably personnel from the served agency, to send and receive ER e-mail.

Printers

Printers, now available for $100 or less, are important for hard copies, especially at control sites such as EOCs, Red Cross chapters, and shelters. This may be a problem when power is lost.

My credentials

I've served as the ARES/RACES director of Chatham Borough and Chatham Township (New Jersey) for over 20 years. Packet was installed at both Emergency Operating Centers after a packet demonstration to officials of both communities in 1988. Since then, we used packet in at least ten drills. Acceptance is especially high at the Southeast Morris Chapter of the Red Cross, which serves the Chathams.

ER-Email was conceived after a county-wide packet drill on November 5, 2001. The post-drill critique revealed the need for a much more user-friendly packet program. Even our formerly active packeteers had become rusty using their favorite programs (we used five different programs) because we all now use the Internet. What would be easier for both hams and our served agencies, than a packet program that mimics Internet e-mail?

O. Paul Schreiber, W2UH

33 Ellers Drive

Chatham, NJ 07928-2218

973-635-1290

otmarpaul@cs.com or **s**

Growth of PC *FlexNet* in the Northeast US

...a networking history lesson

BY JOHN C PAPSON WB2CIK

NET/ROM

In the beginning, like all networking groups, we started off with *NET/ROM*. As various flavors of *TheNet* evolved, we continuously upgraded EPROMs in the TNCs.

The network grew, a 220 band backbone was added with second TNC linked to original 2-meter TNC via a simple null mode type of cable. With time, the single frequency 220 backbone approach evolved into a dedicated point-to-point links between major node sites. Backbone link speeds increased to 2400, 9600, and 19200 baud!

NET/ROM with Diode Matrices

Additional radios/TNCs were added via the use of a diode matrix to link TNCs at a node site. The first diode matrix board was an eight-port design. We discovered quickly it would not work with over six TNCs connected due to RS-232 port loading and sometimes less with a mix of RS-232 driver designs (ie, PacComm's MAX-231 vs. MFJ's Op-Amp). Maximum RS-232 speeds were needed to prevent congestion from occurring at the diode matrix.

To expand a node site beyond five or six radios, two diode matrixes were sometimes linked via a pair of back-to-back TNCs called a "dogbone."

NET/ROM with a G8BPQ Hub

Node sites became limited by number of ports on a diode matrix. At about this time, G8BPQ added driver software allowing connection between the RS-232 port of a *NET/ROM* TNC and the *BPQ* networking program. This enabled *BPQ* running in a PC to function as a hub or router when connected via RS-232 to node TNCs. We tried this approach first at the WB2CIK, Huntington, Long Island, NY site, using a 20-MHz 286 motherboard and just a floppy drive for booting. Inexpensive two-port serial cards were cut and jumped, to generate ninedifferent I/O addresses and nine different interrupt addresses.

We soon learned the value of 16550 UARTs with internal buffering when trying to service nine serial ports with a slow computer! Overall, this approach worked very well with AX.25 radio link overhead being handled by 4.9-MHz Z-80 processors within each TNC and hub-routing by a 20-MHz 286. As the traffic load grew, we upgraded the 286 to a 386 and then a 486! The same setup was adopted for use at NNJ:WA2SNA-2 site atop Ramapo Mountain in Oakland, NJ.

The only reliability problem found was occasional crashes of the *BPQ* software when a neighboring node or BBS would fail backing up traffic. A Touch-Tone decoder connected to one radio allowed for remote reset of PC. This setup was in operation for several years.

NET/ROM with a NOS Hub

Thereafter, we attempted to use *NOS* as the hub in place of *BPQ*. This was a simple swap of software in the PC as all the cabling and EPROMs in the TNCs remained the same as *BPQ* hub arrangement.

Barry, K2MF, built a special version of his *MFNOS* with the hedge that *NOS* was not designed to would work in our application. He was right, but for the wrong reasons.

When it worked, it worked very well. But after operating for a few days (hours at the busiest of node stacks), it would start backing-off AX.25 users in favor of TCP/IP users. And thereafter,

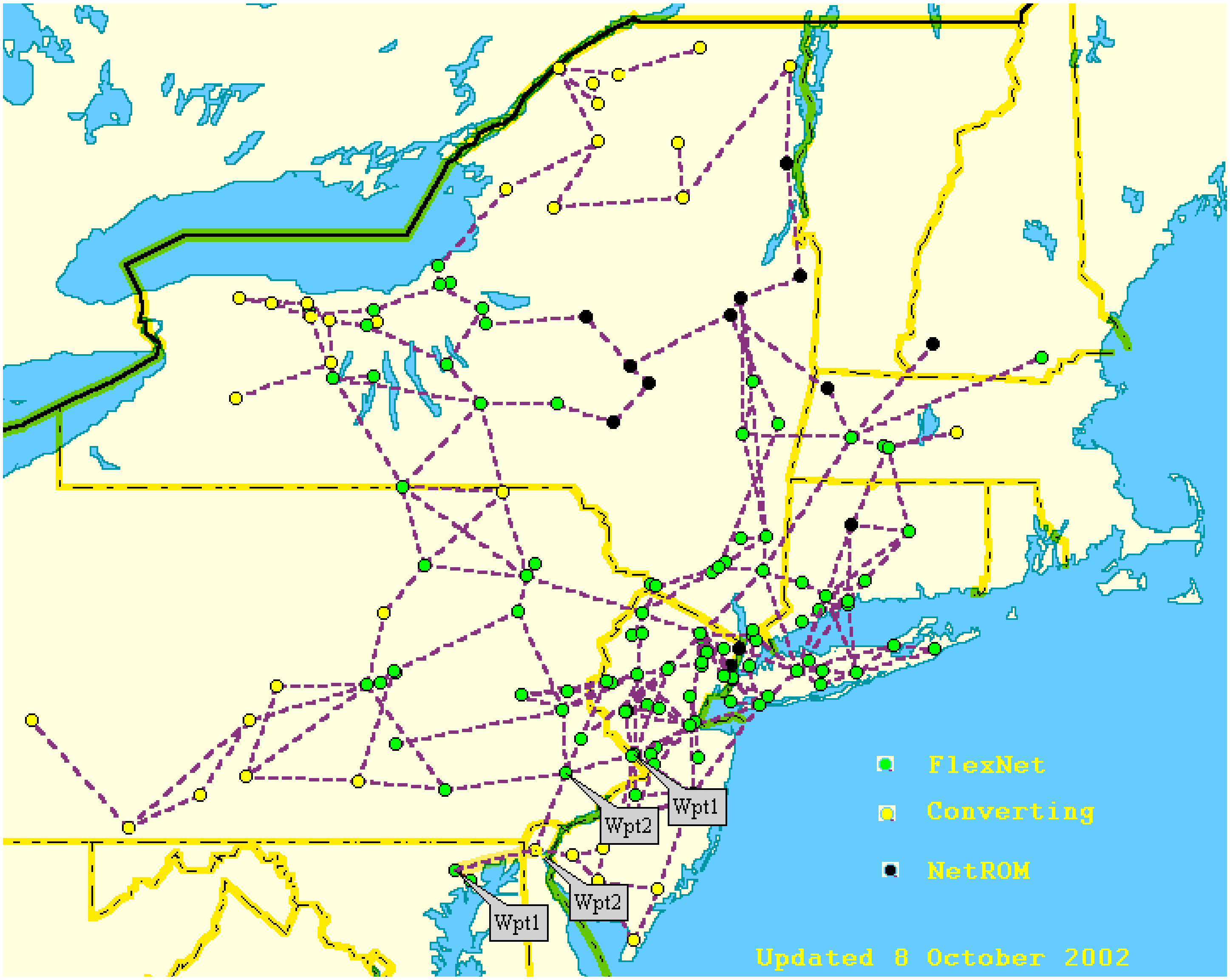
Continued on page 19

Continued from page 18

TCP/IP users would also start getting backed-off. This is a feature of *NOS* designed to protect itself, which unfortunately aggravated AX.25 users.

Software engineers (working at an unspecified company working on an unspecified program) discovered a "memory leak" within the basic *NOS* program. Attempts to obtain a copy of the fix were unsuccessful.

Continued on page 20



Continued from page 19

Never Satisfied

One of the long-standing gripes with all node software was the need to set parameters. Different regional groups would use differing sets of fixed parameters, thereby ensuring network incompatibility between regions. Trying to reach a common set of parameters was like arguing religion, there just was no agreeable solution to everyone!

The second gripe was the need to constantly test the various paths, as correctly stressed by Mark, N2MH, who managed our EastNet backbone network. Some of us had discussions about automating this process.

PCFlexNet with 6PACK

Then, Bob Stevenson, K1UOL, who had traveled in Europe, pointed a few of us towards *FlexNet*.

FlexNet tests each link every four minutes, automatically adjusting parameters and path qualities based solely upon these measurements. No parameters to set, no testing, no egos; it appeared to be just what we were looking for.

Bob's HF BBS forwarding partner, Gerd, DL3MFH, supplied us with *FlexNet* details, such as "D"estination lists of his local node located in central Germany.

At first, we could not understand what we were looking at. It was totally incomprehensible seeing node lists with 600 entries covering most of Europe!

In August 1998, we converted the first three Long Island, NY node sites to *FlexNet*, after months of trying to understand this very different looking software. Having *BPQ* routers surrounded by TAPR-2 TNCs in place at several sites, we simply loaded *PCFlexNet* software into the computer and changed TNC EPROMs from *TheNet* X.1J4 to *6PACK*.

The plan was we could instantly restore previous *BPQ/TheNet* setup with a software and EPROM swap.

What happened is that we were so impressed with the speed and throughput improvement that we left *FlexNet* in place.

Reliability Concerns

There were major concerns with *FlexNet* in a PC at remote sites: reliability of PCs, hard drives, and the software itself. All have proven to be groundless. *FlexNet* nodes are like the Eveready bunny, they just keep running and running.

With almost 100 node sites now converted, only two computer-related hardware failures have occurred over four years.

1) At WA2FNQ, a very old SCSI drive ground itself to dust while the SysOp was on vacation, but the *FlexNet* node just kept on running!

2) WA2WNI's power supply decided not to start one day. No smoke, no fire, no blown fuse, and the fan even still works!

Twice, W1TOM-10, located atop a remote 100-foot fire watchtower, has gone down with a lot of growling and grumbling about unreliable computers.

Both times, the problem turned out to be the AC power cord vibrating out of the outlet, due to wind shaking the tower! After the first time, we added a cable clamp screwed into the wall adjacent to the outlet to keep the plug in. Only to find that the second time, a local two-way radio shop had cut off the same clamp to unplug the node, attempting to isolate the source of a site interference problem. Grrrrrrr !

Further Info

For detailed information on PC *FlexNet* in the Northeastern USA, please go to **www.northeastflexnet.org**

Kits	Price	Qty	Total	Kit Code	Information
PIC-E(ncoder)	$65.00			16	
CompactFlash Card Adapter	$59.00			16	A FlashCard is not included in the kit
Motorola EVM56002 Interface	$150.00			16	http://www.tapr.org/tapr/html/dsp56002evm.html
APRS T-238 Weather Station	$134.00			16	
TAC-2 (Totally Accurate Clock)	$139.00			16	Requires a GPS to operate
Differential GPS (Member Price)	$199.00			16	Requires a GPS to operate
DAS (DTMF Accessory Squelch)	$68.00			8	As seen in Dec 95 QST
TAPR 9600 bps Modem	$80.00			8	
Bit Regenerator	$10.00			1	used for regenerative repeater operation
Clock Option	$5.00			1	used for regenerative repeater operation
PK-232 Modem Disconnect	$20.00			2	simplifies connection of external modems
PK232MBX Installation Kit	$20.00			2	for installation of 9600 modem in PK-232MBX
XR2211 DCD Mod.	$20.00			2	
State Machine DCD Mod.	$20.00			2	
State Machine DCD w/Int Clock	$25.00			2	For KPC2 or other TNC w/o 16X or 32X int clock
Firmware					
32K RAM w/ TNC2 update docs	$20.00			2	
TNC-2 1.1.9 w/KISS EPROM	$15.00			4	includes 1.1.9 Commands booklet (below)
1.1.9 Commands Booklet (only)	$8.00			2	full TNC-2 command set for 1.1.9
TNC-2 WA8DED EPROM	$12.00			2	8 connect version for ARES/Data standard
TNC-1 WA8DED EPROM	$12.00			2	
TNC-2 KISS EPROM	$12.00			2	
TNC-1 KISS EPROM	$12.00			2	
PK-87 WA8DED EPROM	$12.00			2	
TrackBox	$15.00			2	
Publications					
ARRL/TAPR DCC 17-19/98-2000 CD	**$50.00**			**4**	**CD w/ 98 and 2000 Proceedings**
ARRL/TAPR DCC 17-19/98-2000 CD	**$33.00**			**4**	**CD only**
ARRL/TAPR DCC 11-16/92-97 CD	**$33.00**			**4**	**CD with available Proceedings**
ARRL/TAPR DCC 1-10/81-91 CD	**$33.00**			**4**	**CD with available Proceedings**
TAPR Spread Spectrum Update	$18.00			16	200+ pages on ARS Spread Spectrum Topics
TAPR Software Library	$20.00			4	ISO 9660, 650 Megs of info! w/ html nav pages
Wireless Digital Communications	$39.99			28	300+ pages w/disk by: Tom McDermott, N5EG
Packet Radio: What? Why? How?	$12.00			8	130 pages. TAPR's Packet Radio book.
BBS Sysop Guide	$9.00			4	60 pages. by: Barry Buelow, WA0RJT
TAPR's 94 Annual Proceedings	$7.00			4	Papers from the Annual Meeting (Tucson)
TAPR's 95 Annual Proceedings	$7.00			4	Papers from the Annual Meeting (St Louis)
PSR Set Vol 1 (#1 - #17 '82 - '85)	$20.00			16	
PSR Set Vol 2 (#18 - #36 '86 - '89)	$20.00			16	
PSR Set Vol 3 (#37 - #52 '90 - '93)	$20.00			16	
PSR Set Vol 4 (#53 - #68 '94 - '97)	$35.00			16	
ARRL/TAPR 2001 DCC	**$10.00**			**8**	**2001 DCC**
ARRL/TAPR 2000 DCC	$15.00			8	2000 DCC Orlando, FL
ARRL/TAPR 1999 DCC	$15.00			8	1999 DCC Proceedings Pheonix, AZ
Other					
TAPR 11oz Coffee Mug logo	$11.00			4	Logo in black and microwavable gold
TAPR Badge	$10.00			0	include Name and Call for badge
TAPR Shirt - 4 styles					http://www.tapr.org/tapr/html/shirts.html
GPS					
TAC-32 Software Registration	$55.00			0	
Garmin GPS-25	$150.00			28	(data cable included)
Garmin GPS-20/25 Interface/Power Kit	$40.00			8	
Garmin GPS-20/GPS-25 Data Cable	$15.00			2	No Discount.
Garmin GA-27 GPS Antenna	$75.00			8	mcx connector, mag and sunction mounts
Oncore UT+ GPS	$169.00			28	
Oncore VP Interface/Power Kit	$40.00			8	
Oncore GT+ GPS	$149.00			28	
Motorola Antenna 97	$65.00			8	bnc connector, mag mount
MCX Rt-Ang Connector w/coax pigtail	$15.00			2	No Discount.

Subtotal: ☐ ☐ **Added Total Kit Codes**

All prices subject to change without notice and are payable in U.S. funds. Members receive 10% off on Kits and Publications. Please allow six to eight weeks for your order to be shipped. For specific information on kits, see Product Description flyer or check the TAPR web site - www.tapr.org.

Tucson Amateur Packet Radio
8987-309 E. Tanque Verde Rd #337
Tucson, Arizona • 85749-9399
Office: (972) 671-8277 • Fax: (972) 671-8716
Internet: TAPR@TAPR.ORG • www.tapr.org
Non-Profit Research and Development Corporation

Winter 2002

http://www.tapr.org • ftp.tapr.org • tapr@tapr.org

Membership	Price	Number of Years	Total
United States	$20.00		
Canada/Mexico	$20.00		
International	$25.00		

○ **New Member** ○ **Renewal**

☐ **SubTotal**

– ______ **Membership 10% Discount**
Except were noted Member #: ______ (Place new if joining)

______ **Total Sales (Subtotal minus discount)**

______ **Texas Residents (8.25% tax)**

______ **Membership (New or Renewal)**

______ **Shipping and Handling**

For Total Kit Codes Between

1 - 7	8 - 15	16 - 27	28 - 55
Add $6	Add $7	Add $8	Add $9

Kit Codes above 55 or International orders must contact TAPR for amount.

☐ **TOTAL Order Amount**

☐ Charge my credit card (check one):

☐ VISA ☐ MasterCard

Acct. # ______

Expiration Date: ______

Signature on card: ______

Name / Call: ______

Street Address: ______

City / State / Zip: ______

Country: ______ Phone Number: ______

Internet E-mail: ______