PACKET STATUS REGISTER

Tucson Amateur Packet Radio Corporation
A Non-Profit Research and Development Corporation

Winter 1999
Issue # 73

Published by:
Tucson Amateur Packet Radio
8987-309 E. Tanque Verde Rd.#337
Tucson, AZ 85749-9399
Phone: 940-383-0000
FAX: 940-566-2544
tapr@tapr.org
www.tapr.org

Office Hours:
Tuesday - Friday
9:00am-12:00, 3:00-5:00pm Central

In This Issue...

President's Corner

A few things to look for in 1999 include the ARRL and TAPR Digital Communications Conference to be held in Phoenix, AZ, September 24-26th. The Phoenix conference looks to be as strong as this last year in Chicago, so I hope everyone that couldn't attend the 1998 conference can make it this year. Don't forget that the Dayton HamVention will be on May 14, 15, and 16. We expect another strong speaking schedule on Friday and we should have some concept of the banquet speaker for our dinner by the next *PSR*.

We had several new items arrive at the office since the last *PSR*. The DSP56002EVM Radio Interface kit, CompactFlash Card Adapter Kit, and Spread Spectrum Update Publication are now available. The PIC-E evaluation is beginning with a fifty unit participation. Work continues on several other projects as well. More in the *PSR* on these various projects.

The big news this quarter is the revision of the TAPR.ORG site. Four major areas for updating had been on our list for sometime. When visiting www.tapr.org or ftp.tapr.org here are some of the changes you will encounter. First, the ftp directory structure has been flattened a level or two to make getting to those files you are looking for easier. Second, the web pages have had a total overhaul. The new design allows visitors once in a

Look for TAPR at these Upcoming Events

May 14-16, 1999	Dayton HamVention
Sept. 24-26, 1999	ARRL & TAPR Digital Communication Conference Phoenix, Arizona

Ballot Deadline APRIL 15th

President's Corner, continued...

section to access all the major areas of that section without backing out of a page all the time. As part of the web page revision, a new on-line order system was developed and coded. We are planning on implementing a secure web server sometime later this year to help further with our e-commerce ability. Third, we have implemented an entirely new mail archive index and search engine. The previous utility we had used was falling far short of our requirements. The new system provides very quick access to the information databases we have on TAPR.ORG. Fourth, we replaced listproc with Lyris as our listserver. The new list package reduces the amount of daily maintenance the previous system required to zero. The old system required daily watching and was klunky. Lyris is fast, seems to be very reliable, and is easy to manage. We hope that everyone enjoys these changes to the TAPR.ORG system.

To finish this quarter, let me point out that this is election time. We have four people running for the three slots available on the TAPR Board of Directors. Each member has strong credentials, so please take a moment and look over the election information and send in your vote. The TAPR Board of Directors is responsible for setting directions and goals of the organization, so this is your opportunity to have a say. We will again be accepting ballots printed in this *PSR* as well as doing an on-line balloting system. Please vote!

Until next quarter and lots more fun!

1999 ARRL and TAPR Digital Communications Update

Phoenix will be the site of the 1999 ARRL and TAPR DCC. Contracts are underway with the hotel and the conference should be held on the weekend of September 24th-26th. No local host has been found yet. PRUG (Packet Radio Users Group of Japan) will again be hosting a Friday evening social and technical presentation. More information will be disseminated in the coming weeks as the hotel, schedule, workshops, and registration prices are fixed by the conference committee. Keep an eye on the web page http://www.tapr.org/dcc.

There will again be a National APRS Symposium held on Friday and Steve Dimse, K4HG, will again coordinate the event. If you have suggestions for the Sunday seminar, please let the office know. Student papers will again be accepted and the deadline will be in June. We hope that this year's DCC will continue the positive trend witnessed over the last several years.

Opinions expressed are those of the authors and not necessarily those of TAPR, the Board of Directors, Officers, or the Editor. Acceptance of advertising does not constitute endorsement, by TAPR, of the products advertised. APRS is registered to Bob Bruninga, WB4APR
MIC-E is a trademark of Bob Bruninga, WB4APR

Postmaster: Send address changes to TAPR, P.O. Box 51114, Denton, TX. 76206-0114. *Packet Status Register* (ISSN 1052-3626, USPS 005-419) is published quarterly by the Tucson Amateur Packet Radio Corporation, 1418 Ridgecrest, Denton, TX 76205. Membership in Tucson Amateur Packet Radio, including a subscription to *Packet Status Register*, is $20.00 per year in the U.S. 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 U.S. funds. Membership and *Packet Status Register* cannot be separated. Periodical postage paid at Denton, TX.

***PSR* Editor**:
Bob Hansen, N2GDE
P.O. Box 1902, Elmira, N.Y. 14902-1902
Internet: psr@tapr.org

Advertising:
Contact the TAPR office.

The Tucson Amateur Packet Radio Corporation is a non-profit scientific research and development corporation [Section 501(c)(3) of the U.S. tax code]. Contributions are deductible to the extent allowed by U.S. tax laws. TAPR 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.

Article submission deadlines for upcoming issues:

Spring 1999	March 15, 1999
Summer 1999	June 15, 1999
Fall 1999	September 15, 1999
Winter 2000	December 15, 1999

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 communicators, please contact the editor so that your work can be shared with the Amateur community.

The preferred format for articles is plain ASCII text; the preferred graphic formats are HPGL or PCX. However, we can accept many popular word processor and graphic formats. All submissions on diskette should be formatted for MS-DOS.

1999 Dayton Hamvention Digital Communication Events!

We are currently setting the schedule for the TAPR Forum at Dayton HamVention. It will again be Friday morning like last year. It should again be followed by the APRS Forum. Keep an eye on the TAPR web page for full details as Dayton approaches (http://www.tapr.org/tapr/html/conff.html)

Make your plans now for the 1999 "Packet BASH" sponsored by TAPR and the Miami Valley FM Association to be held on Friday of the Dayton Hamvention. The BASH will again be held at the NCR location south of Dayton. It is a great location.

A buffet dinner, a raffle for some neat prizes, a great program, and lots of fun will cost approximately $20 per person. More on the final price once we have a contract on the dinner. Keep an eye on TAPR-BB and the Web pages.

We hope that this will provide an opportunity for packet and digital radio enthusiasts to have a great night out while at Hamvention.

The schedule of events is still tentative, but will look something like this:

1900	Dinner
1945	Welcome
2000	Keynote Address
2030	Raffle
2045	TAPR SIG Meetings

For more information, send email to "packbash@n8ur.ampr.org" or stop by the TAPR booth at Dayton for schedule and map.

PSK31 for Linux and SoundBlaster

Eduardo Jacob, EA2BAJ
jtpjatae@bicc00.bi.ehu.es
http://det.bi.ehu.es/~jtpjatae/ham.html

Hansi Reiser DL9RDZ has prepared a PSK31 version for Linux and the SoundBlaster. It's available at

ftp://det.bi.ehu.es/pub/ham/psk31/psk31_lx.tgz

This is provided linux style, only sources, no binary. As usual, you can take a look at what PSK31 is, at it's homepage:

http://aintel.bi.ehu.es/psk31.html

Now if somebody want to write a DOS or Windows version with the Sound Blaster, he has part of the work done.

1999 AMSAT-NA Annual Meeting and Space Symposium

Recent and future developments in amateur radio satellites will be presented in San Diego, California on October 8-11 at the 17th Space Symposium and AMSAT-NA annual meeting. The location for this year's event is the beautiful Hanalei Hotel in the heart of San Diego's Mission Valley, a short distance from many of San Diego's attractions. For entertainment you can visit Sea World or the world famous San Diego Zoo. History buffs can "discover" San Diego at Cabrillo National Monument or experience periods of San Diego's history by visiting Mission San Diego de Alcala, Presidio Park, Old Town San Diego State Park and Balboa Park. Golfers will enjoy playing on any of San Diego's many fine golf courses. Of course, there is no shortage of shopping with Mission Valley Center, Fashion Valley Mall, Seaport Village, and Horton Plaza a short ride away. If relaxation is more your style visit scenic La Jolla, Mission Bay Park, Coronado Island or any of San Diego's beautiful beaches.

If you're interested in an amateur radio vacation, visit the ARRL Southwestern Division Convention (HAMCON) on the Queen Mary in Long Beach, California October 1-3, then see some of Southern California's sights before attending the 17th Space Symposium and AMSAT-NA annual meeting in San Diego October 8-11. A tentative agenda includes the Space Symposium all day Friday Oct. 8 and Saturday Oct. 9. There will be a satellite session for beginners Friday evening and the annual banquet will be Saturday evening. An IARU Satellite meeting will be held Sunday morning Oct. 10 followed by the AMSAT-NA board meeting Sunday afternoon and all day Monday Oct. 11. Further details will be available later this year.

Hotel reservations for the Hanalei Hotel in San Diego can be made by calling the hotel at (800) 882-0858. Be sure to mention AMSAT to receive the $85 per night discounted group rate. This rate is available for rooms reserved between Oct. 4 and Oct. 12. The local contact for the AMSAT event is Duane Naugle, KO6BT and can be reached at ko6bt@amsat.org.

Hanalei Hotel	www.hanaleihotel.com
San Diego City	www.sannet.gov
San Diego Visitors Bureau	www.sandiego.org
San Diego Zoo	www.sandiegozoo.org
San Diego Gaslamp Quarter	www.gaslamp.com
San Diego Horton Plaza	www.hortonplaza.com
San Diego landmarks	www.sddt.com/arts/landmarks
Balboa Park	www.sddt.com/features/balboapark
Cabrillo National Monument	www.nps.gov/cabr/home.html
ARRL SW Division Convention	www.qsl.net/arrlsw/hamcon
Queen Mary Hotel	www.queenmary.com
Queen Mary Enthusiasts	www.queenmary.org
Disneyland	disney.go.com/Disneyland/index.html
Knotts Berry Farm	www.knotts.com

TAPR 900MHz FHSS Radio Design Update

Report: December 20th, 1998

The TAPR Spread Spectrum Web page has been updated as well as the FHSS project status page (http://www.tapr.org/ss). The new project page should be easier for the occasional reader to browse updates.

The development group met for a social/review meeting on November 28th, 1998. The goal of the meeting was to have some food and discuss the project.

As of the meeting, the RF board is transmitting a QPSK modulated signal on a single frequency of 913MHz. Basic functionality has been demonstrated; however, there's a lot more to test on the transmit side of the design. There has been really good progress made in a short period of time. This was a major milestone reached. Work will start to check out the receiver when the transmitter is a little further along. This testing will be more difficult, but since a lot of the circuitry is common, there is less to check out. A second unit will have to be completed, including known modifications to the PCB, for this testing.

The digital board CPU, network interface, and the Qualcomm forward error correcting (FEC) chip Q1900 is working. We are also able to communicate with the QPSK decoder registers. Initial testing indicated that a more elaborate PLD solution would be required to support the high speed phase adjustments and to support the odd clocking required for FEC. A design is completed and ready for test and debug. A small PCB will be built to test this before we make another pass at the digital PCB. The group has discovered that high-speed phase adjustment is required for the Harris QPSK demodulator. The PLD has some 350-500 gates of logic.

The group has been very happy with the performance of the Qualcomm part. Thanks to Frank Antonio, N6NKF, for his assistance in getting us the samples we require for the project.

The Software/Firmware is in place to handle the key networking interface and to continue radio testing. Current plan is to use an HTTP interface to control and test the radio functions. A PC-based tool to interface with the radio using UDP packets also works. The http software on the radio is working and the radio's basic parameters can be accessed via a web page and reconfigured.

Also, the group has decided to use a Dallas Semiconductor Button part in the project. The "button" is a two wire interface which provides a registered MAU ethernet address for the radio.

The function of the hardware and software is so complex that each small section of the radio must be proven before we can move on to the next issue.

The project moves forward, but as each small step is taken we can see just how much more work will be required before we have a fully functioning high-speed radio available.

Status update for the FHSS radio project.

1. The SPI driver is written and working. The protocol module for the interface to the PLD is written, but cannot be tested until the PLD is ready, and that's a ways down the road.

2. HTTP/1.0 daemon is written and code to send/receive HTML files is working. This allows using a web-browser to read the configuration from the kernel and updating some of the parameters. The final parameter save portion is just stubbed off right now until we settle on how they are stored and configured in memory. But we can demo the operation from a browser and it looks nice.

3. Further testing of the Ethernet performance is occurring. We are starting to write a layer-2 demultiplexer in the TCP/IP stack. The current XINU stack only performs layer-3 routing and has no layer-2 functionality. This will then feed some new software that does segmentation of the ethernet frame to fit into the radio's 10 millisecond transmit periods and then encapsulate it with the radio link protocol. Layer-2 demultiplexing will probably require some frame cloning on ethernet packets containing broadcast addresses (sending the copy to layer-3).

Update: January 7, 1999.

During the TPRS Fall Digital Symposium, held in Austin on December 12th there was a lot of discussion about connecting radios to Internet Service Providers, based on actual experiences reported first hand. Joe Borovetz, WA5VMS's, experiences in working with his ISP and past discussion at conferences makes it seem that the first level of networking of the radio should be basic LAN bridging. This would be easier to setup and configure than routing in the early stages of deployment, where a city will have a handfull of radios to begin with.

As a result, we have modified the XINU stack on the radio to incorporate a 6-port software Ethernet switch between the TCP/IP stack and the Ethernet interface. This switch interfaces at the Ethernet-frame level (layer 2) between the radio stack and the 10-base-T port on the TAPR radio. It is an Ethernet learning-bridge, and appears to be compliant with IEEE 802.1D specification for learning bridges.

This required modification of some of the modules in XINU, and the inclusion of several new modules dealing with layer 2 switching. This has been completed and tested (as much as we can without the radio link). The stack and the 10-base-T ports have been verified to operate properly through the switch.

The learning bridge provides some intelligence so that LAN bridges will not forward unnecessary LAN traffic across the radio link, while still allowing transparency in the LAN bridging for ARP and other broadcast protocols. Additionally, the switch allows the stack on one radio to talk to the stack on another radio. This allows configuration and status control of a stack from both the near and far ends of a radio link.

The solution that has been implemented results in a radio with one IP address (the on-board stack) and one Ethernet address (the 10-base-T port). The radio link shares the Ethernet address from the 10-base-T port so that the bridging tables on the two radios will properly synchronize automatically. The layer-2 switching function assures optimum usage of the bandwidth on all three ports: radio, stack, and 10-base-T. Discussion with some of the hams at the meeting who work at Internet Service Providers (ISPs) and Internet Network Access Providers (NAPs) indicates that this solution should work well with minimized setup difficulties.

The team has found and fixed about four bugs in the radio transmitter section and are now getting about 35 milliwatts output power at 915 MHz (should be 100 mW, but this is only 5 dB. away). Bob Stricklin, N5BRG, needs to get the -5v supply working before we can test the T/R switch and see how much of this gets to the antenna connector.

This development will allow signal tracing back down the receiver path on the radio. 35 mW on the transmit board provides a strong signal on the receiver for troubleshooting.

Tom McDermott is experimenting with two different acquisition algorithms and derived the analytic performance of one of them. He will continue to run simulation determining which acquisition algorithm we will want to try to use in the first radio.

TAPR 900Mhz FHSS Project Fund Raiser

The TAPR FHSS Project continues to need money to support development. If you can help, please call the office and make a donation.

You can call the office at (940) 383-0000 or Fax (940) 566-2544 to make your donation by MC/Visa.

Galois C++ Class Library and RS Codec

Phil Karn, KA9Q
karn@homer.ka9q.ampr.org

I have written a C++ class library that does arithmetic and algebra over finite fields, more specifically GF(256).

Also included is a Reed-Solomon codec functionally equivalent to the one in C I've had on my web page for some time.

The C++ version is considerably slower than the C version so I recommend the C++ version only for study and educational use. It's much easier to follow the C++ code as the expressions more closely resemble those in the coding theory textbooks.

http://people.qualcomm.com/karn/code/

TPRS Fall Digital Symposium

The Texas Packet Radio Society held its Fall Digital Symposium this past December in Austin, Texas. There was a lot of discussion regarding the TAPR FHSS radio project and other topics. The following presentation were made:

Linux systems
Stu Green
Setting up Linux as a gateway router.
How to setup routing tables, default routes, etc.
Static vs. dynamic addresses, class-c addresses, etc.
Tunnels - what they do, how to set one up.
Proxies / IP address masquerading
Slip and PPP on serial interfaces
Ethernet interfaces

TAPR FlashCard OS Project
Greg Jones, WD5IVD in for John Koster, W9DDD

TAPR FHSS Radio Design Status Update
Tom McDermott, N5EG

ISP interconnection project
Joe Borovetz, WA5VMS

TexNet - AX.25 tunneling
Bob Morgan, WB5AOH

TexNet - APRS capability
Bob Morgan, WB5AOH

Texnet Network and Status Update
Harry Ridenour, N0CCW

APRS Update
Mike Heskett, WB5QLD

All the audio and overheads can be found on the TAPR virtual meetings page:
www.tapr.org/tapr/html/tprs98winter.html

TAPR Board of Directors Elections

The following four members have agreed to run for the three available positions on the Board of Directors. You may vote for these individuals and/or any write-in candidates using the ballot printed on the next to last page of the *PSR* or using the on-line election web page (details in the next section).

Deadline for balloting is March 31st, 1999. Board members elected will serve a three year term.

The following TAPR members have been nominated for election this year to the TAPR Board of Directors.

Steve Dimse, K4HG
Internet: k4hg@tapr.org

I am a 39 year old Emergency Physician for the University of Miami. My undergraduate degree is in Computer Science, and I've been using microcomputers for 20 years. I've been a ham for 9 years, and in that time have sampled most of the facets of amateur radio, but have settled on the digital modes (and APRS in particular) as my primary activity. I've spent most of my free time in the last three years working on the APRS internet system, authoring javAPRS (a java version of APRS) and APRServe (the central server for APRS on the internet). I've also been involved in a number of TAPR projects, including co-chair of the APRS QSY committee, an organizer of the Mic Encoder Beta test, and a member of the Pic Encoder development team. I've attended the last four Digital Communication Conferences, presented papers at the last three, and organized the APRS seminar at the last two. I believe that the success of R&D organizations like TAPR are essential to the future of amateur radio, and I intend to do all I can to see that TAPR fulfills its promise.

Bob Hansen, N2GDE
Internet: n2gde@tapr.org

My primary interest in Amateur radio is digital communications. Since becoming a TAPR member in 1989, I have served as the *Packet Status Register* editor and as a Board member. My educational background is electrical engineering, computer science, and business administration. Professionally, I am project leader and system designer for embedded-computer control systems.

TAPR and its members have contributed a great deal to the packet radio community, but there is still much more that can be done. To remain effective, TAPR must ensure that it has adequate resources to fund, and staff, its projects. I would like to continue to be involved in these efforts.

Steve Bible, N7HPR
Internet: n7hpr@tapr.org

In 1985 I developed a love affair with packet radio. The timing couldn't be better. At the time I was studying computer engineering at the University of Utah. My senior project was a weather station accessible via packet. To me amateur radio has always been a vehicle to learn new things. Serving on the board of TAPR for the past three years has been an extension of this learning.

The first project I took on was W3IWI's Totally Accurate Clock. Serving as project manager taught me all about putting kits together, from design to out-the-door. Since then over 300 TAC-2 kits have graced my kitchen table. Other projects I have worked on are the EVM Radio Interface, PIC Encoder, The Spread Spectrum Update Book, and many web pages on DSP, Spread Spectrum, etc.

I see TAPR in the role of creating "enabling technologies" that aid the experimenter in learning. I find my role in TAPR to scout for these technologies and bring them to the TAPR experimenter community as new projects. It's pretty exciting to watch others do great things with these projects.

TAPR represents a technologically savvy group that is defining the future of amateur radio. When new technologies are introduced we find that the FCC rules and regulations are not flexible. TAPR has an important role in making changes to rules and regulations.

I am running for another three-year term as TAPR board member. I would very much like to continue the role of technology scout on your behalf. Your vote for me says that you want new and exciting projects and flexible rules and regulations to use them with. Thanks es 73 de N7HPR

Steve Stroh N8GNJ
Internet: n8gnj@tapr.org

My name is Steve Stroh N8GNJ, and I'm running for a three year term on the TAPR Board of Directors and would appreciate your vote. If elected, I hope to bring a bit more geographic and Amateur Digital Networking "diversity" to the TAPR Board of Directors.

The Pacific Northwest, the area I hope to represent, is home to some of the most influential technology, computing, telecom, and Internet organizations in the world, and the people who make them what they are. Just a few of them are headquarters of the Boeing Company, Microsoft Corporation, AT&T Wireless, RealNetworks, Teledesic LLC, Walker, Richer, & Quinn, Linux Journal, Meteor Communications Corp., and the University of Washington.

The Pacific Northwest also has many Amateur Radio Networking accomplishments, including the Vancouver Amateur Digital Communications Group (VADCG) and their ground breaking work on Packet Radio, the Vancouver network of multiple 56K repeaters, and the Puget Sound Amateur Radio TCP/IP Network, a network of 2m, 222 MHz, and UHF 9600 baud repeaters all running TCP/IP as the primary networking protocol. Notably, the 1996 DCC was held in Seattle.

Professionally, I'm a System Administrator dealing primarily with Windows NT and UNIX systems for a large company. My original training was to be an Electronic Technician.

I've been a member of TAPR for more than ten years. In the last several years, I've become more directly involved in TAPR's activities. Since early 1997, I've held the appointed position of TAPR Secretary. For the last several years, I have been the Coordinator of TAPR's Networking Special Interest Group (NetSIG). I'm a co-coordinator of the ARRL and TAPR Digital Communications Conference (DCC).

In the last three years, I have attended two of the three Board of Directors meetings held in conjunction with the Dayton Hamvention, and all three of the Board of Directors meetings held at the DCC.

A related activity is that since mid-1997, I write a column in Boardwatch Magazine called Wireless Data Developments. Boardwatch's target audience is Internet Service Providers (including "hardcore" Internet techies and "internal ISP's such as those at corporations and colleges). With that target audience in mind, Wireless Data Developments is written to try to cover developments in Wireless Data Communications that are relevant to ISP's. Not surprisingly, there's significant crossover between Amateur Radio Networking and Wireless Internet Networking. In the sidebar of the column, I mention TAPR prominently and periodically take minor liberties with the column to mention Amateur Radio Networking topics and to plug TAPR's activities. I've been encouraged to "broaden the crossover" and begin writing a column in the TAPR *Packet Status Register*, and I hope to begin that in 1999.

It seems to me that no one ever seeks office without having at least something of an agenda, and I'm no exception to that. Here's an overview of some of the things I hope to work towards in the next few years.

For several years, it's been a goal of mine to get TAPR and local / regional Amateur Radio Digital Groups a bit more involved with each other to leverage each other's strengths. If I and TAPR are able to implement such a program, I have volunteered to be the TAPR coordinator for this activity.

It's my belief that Amateur Radio can attract many more new members if there is something in Amateur Radio that they consider relevant and interesting. 1200 baud and 9600 baud Packet Radio, Packet Bulletin Board Systems, and Net/ROM-type networking are simply not relevant to this target audience. TAPR's Spread Spectrum Radio currently under development, several proposed Amateur Radio / Internet projects, and increasing use of TCP/IP Networking and UNIX ARE relevant and interesting to this target audience. To capture the interest of this target audience, TAPR enjoys a unique position in Amateur Radio, and with carefully tailored advertisements, Web pages, articles, etc., it's my belief that we REALLY CAN increase the Amateur Radio population. Best of all, by definition, this increase would knowledgeable about, and supportive of the issues TAPR is most concerned about in Amateur Radio.

Lastly, being involved in TAPR and commercial wireless data communications more intimately than most, I've discovered that there is an INCREDIBLE amount of activity going on that is interesting and relevant to TAPR's mission and membership. But, much of this information doesn't find its way into the general Amateur Radio "press". I think that should change, and if elected, I'll be able to work to change that. One thing that I think has been needed for some time is that I feel that back issues (but not necessarily the most current issues) of the Packet Status Register newsletter should be available on www.tapr.org. I'd also like to see TAPR pursue "strategic relationships" with International Amateur Radio Digital groups and non-commercial organizations that make extensive use of wireless digital communications. Much of what TAPR and Amateur Radio has to offer- low-cost, reliable, wireless digital communications is EXACTLY what is needed in situations such as lesser-developed countries.

I'll pursue these goals even if I'm not elected. I previously ran for the TAPR BoD in 1997 and "kept going" even though I wasn't elected. However, I feel I would be able to pursue these goals much more effectively if I was a full, voting member of the TAPR Board of Directors. If you have any questions or comments about my goals for TAPR, I'd appreciate hearing from you via e-mail - n8gnj@tapr.org. If you agree with my goals for TAPR, I would sincerely appreciate your vote. Thanks!

Voting for the TAPR BoD

You can vote either by the ballot printed in this *PSR* or by electronic means using the World Wide Web.

Please find the mail-in ballot on page 23. You will need to include your membership and check numbers on it so that we can ensure no duplicate ballots. (Both numbers are printed on your mailing label for this issue.)

To vote electronically, go to www.tapr.org/elec. You will be asked to give your membership number and check number which can be found to the right of your membership number on the mailing label of this *PSR*.

TAPR List server upgraded to Lyris

TAPR has been looking at implementing a new list package for sometime. After some research, Lyris seemed to be the best package for our organization. In January we built up a second system for the TAPR.ORG site to host the list processor and eventually to be the secure server for handling order transactions. We installed the Lyris package on the new system and began to move over the development and project lists. After about a week of testing, we took the big leap and moved everyone over. The one thing everyone should notice is that it is fast. You post something on APRSSIG with its 1800 members and all the mail is sent out in a matter of a few minutes. Compared to the older list software, this is a drastic improvement.

First of all, almost everything that you might need to do on the server is most easily and effectively performed via Lyris's Web interface at:
http://www.tapr.org/cgi-bin/lyris.pl.

In case you do not have Web access, here are some key commands; in all cases, the default server address is: lyris@lists.tapr.org. Note that commands may be placed in either the subject line or in the body of the message. Just replace 'listname' with the name of the actual list.

Subscribing:
subscribe listname your_name
join listname your_name
alternate address: join-listname@lists.tapr.org

Unsubscribing:
unsubscribe listname
leave listname
alternate address: leave-listname@lists.tapr.org

Setting Membership To Digest Mode:
set listname digest

Setting Membership To Index Mode:
set listname index

Setting Membership To Discussion Mode:
set listname mail

Temporarily Suspending Mail From The List:
set listname nomail

Resuming Mail From The List:
(command depends on your preferred mode)
set listname mail
set listname digest
set listname index

Acknowledgment:
set listname ack
(sends you a confirmation message when your posts are sent)
set listname noack
(no confirmation message when posts are distributed)

Receiving Copies Of Your Own Posts:
set listname repro
(receive copies of your own posts)
set listname norepro
(do not receive copies of your own posts)

Setting Your Password:
set listname pw=mypassword
(where "mypassword" is the password you wish to set)

Determining Your Membership Settings:
query listname

Retrieving The Current Version Of This Document:
get listname hello

More Information On Lyris's Commands:
help

Posting To The List:
You must be a subscriber to post messages. Sending mail to this address will distribute it to all the members of the mailing list: listname@lists.tapr.org

Changing Your Settings/address Via The Web Interface

To do this, go to www.tapr.org/cgi-bin/lyris.pl and click the link in the "Change Your Settings" section. You'll need to enter your e-mail address and password (if you chose one) to continue.

At the following page, you can read messages, post a message, adjust your settings, or unsubscribe. From the settings page, you can change your status (MAIL, DIGEST, INDEX, or NOMAIL), choose whether or not to see your own messages, and choose whether or not you want to receive a separate acknowledgment via e-mail when one of your messages is posted to the list, You can also change your e-mail address if you need to. When you change your settings, make sure you click the "Save" button at the bottom of the page.

Archives

The listname SIG archives are available on the Web, at the following URL: http://www.tapr.org/sigf.html

Bounced Mail

One of the key features of Lyris is its ability to handle mail bounces transparently to the list owner; bounced mail will be handled primarily by the Lyris list server.

Posts/Replies From Digest Users

You **must** change the Subject line when replying to a digest; all posts with "listname DIGEST" in the Subject line will be rejected by the server (you will receive a "rejection letter" when this happens).

APRS Network Congestion

Bob Bruninga, WB4APR
bruninga@nadn.navy.mil

[From the APRS SIG]

On Thu, 21 Jan 1999, someone wrote:

> I sometimes wonder what use APRS provides... seems to lack purpose... My first thought was that the fixed stations, ...(i.e., WIDEs), should significantly reduce their reporting rate. I think the default for MacAPRS is every 60 minutes.

It is important to note that APRS was written ONLY to be a real-time tactical communications tool to help the flow of information at things like special events, emergencies, Skywarn, the EOC and just plain in-the-field use under stress. But like the real world, 99% of the time it is just used 24 hours a day 7 days a week waiting for the unlikely serious application.

I still fight hard to make sure that nothing we do to enhance 24/7 operations ever undermines our ability to operate in local areas under stress. Here are the details of that philosophy:

1) The objective is to have a net-cycle time of 10 minutes under local use meaning that within 10 miuntes of your arrival on the scene, you will have captured the entire Tactical picture without transmitting a thing (and adding QRM). This fundamental "net cycle time" assures users of a "no-worse-than" time latency in his information that he can trust.

2) ALL-Queries. These are not the solution. We do not want everyone doing an all-net-query each time he starts up just to find out. For each and every QUERY, it is like DOUBLING the rate, since EVERY station on the air then responds. This is bad QRM to everyone.

3) All stations, even fixed stations should BEACON their posit at the "net-cycle-time" rate. In a stress situation, stations are coming and going all the time. The POSIT not only shows you where he is without asking, but also that he "is still on the air." This is important. We do it in voice nets, we do it in APRS. We must know who is on the air and available. I do not want hour-old ICONS on the map masquerading as usable assets when they long since shut down.

4) We cannot assume that all APRS users responding to a stress event understand all the ramifications of APRS and the statistics of the channel. We can not rely on user-settings to avoid killing a stressed net. Thus, APRSdos automatically adjusts its net-cycle time to try to anticipate when the software is under stress usage. It uses the following assumptions:

- The rate is 10 minutes for DIRECT operations (probably an event)
- The rate is 10 minutes for ONE digipeater (Probably an event)
- The rate is 20 minutes for TWO digipeaters
- The rate is 30 minutes for THREE or MORE HOPS..

5) Since almost all home stations set their paths to 3 or more digipeaters for routine operations, then we have a default rate of 30 minutes. This again I hope is a universal standard that everyone can BANK on. If you turn on your radio and APRS and do nothing else, then in 30 minutes under routine operations you can **assume** that you pretty much have the total picture of all APRS stations within range.

6) Digipeaters are our most valuable and important assets. Travelers who may be within range of a given digi for only a short while may completely miss a Digi beaconing less often than 10 minutes. More distant users have time to wait. With recent TNC firmware, DIGIS can set multiple paths and rates for their packets. I request the following settings for their self identifying posit beacons:

DIRECT	Once every 10 minutes
VIA WIDE	once every 20 minutes
VIA WIDE,WIDE	once every 30 minutes
VIA WIDE,WIDE,WIDE	once every hour
VIA WIDE4-4	once every 2 hours
VIA WIDE5-5	once every 4 hours

Notice that it takes you more than 4 hours to cover the distance of 5 WIDEs, so on any trip anywhere you will see all the digis you will need. You will see the local ones first, but will also see the ones in front of you before you need them too.

7) We can argue all day about what are the optimum rates, but I feel we must have a common expectation for all users. My best guess was 10 and 30 minutes for the two extremes (but common) usages and it is built into APRS that way. I do not like the 1 hour rate. It is too long to wait, and encourages people to QUERY. And for every Query, the "effective" rate for every station in the net just HALVED.

8) Finally, by having these two assumed rates, it gives all the network designers the fundamental assumptions of channel loading necessary for good engineering design.

[Also see the article "APRS Emergency Planning" on page 30 for related information.]

palmAPRS Beta on FTP

Mike Musick, N0QBF
71301.3106@compuserve.com

With great pride (and a healthy dose of trepidation), I would like to announce that the first public beta release of "APRS for the Palm OS Computing Device" - palmAPRS to you and me - is now in the APRS area of the TAPR (our host) FTP site. Download it from:

ftp://ftp.tapr.org/tapr/SIG/aprssig/files/palmstuff/palmaprs/palmAPRS.zip

The compressed file contains the Palm application, the main "read me" file, and a short note about maps.

Speaking of maps, there are a dozen or so maps available at:

ftp://ftp.tapr.org/tapr/SIG/aprssig/files/palmstuff/palmmaps/

Check the "Maps_Read_Me.txt" file for information and an index of what's what. Map generation is a continuing project, so if what you want isn't there yet, drop me a line and we'll see about getting it there. One important note about maps - "large area" maps such as whole states, regions, countries or continents don't have a way to be created yet. This problem has a solution, but probably one not available for a few weeks.

palmAPRS only runs on the 3Com Palm III, or PalmPilot Personal or Professional models upgraded with the 3Com "2MB upgrade", which updates the device to Palm OS 3.0. Grayscale display is an important feature of palmAPRS, and it plus a couple of other minor features will not work on the earlier Palm OS.

I must ask everyone's indulgence to keep in mind that palmAPRS is A BRAND NEW, BUILT-FROM-THE-GROUND-UP program. Things that Bob, Mark & Keith, Steve and Brent have worked out months if not years ago are now new again. So bugs in things long ago "not a problem" in the other APRS programs are going to be subject to glitches as we get the wrinkles ironed out.

Also, the program as released is "early" beta. The basic function is there, but there are many feature and operational holes, representing a work still in progress. Check the readme.txt for a list of things we know aren't done yet before saying "but it doesn't...".

One caveat not mentioned in the docs - you might keep a paper clip handy (for hardware reset) just in case the program crashes hard and locks-up your Palm. Hard crashes have been few and far between during recent development, but I'd keep one handy just in case.

Looking forward to user reports!

DGPS - Differential GPS

Tom Clark, W3IWI
clark@tomcat.gsfc.nasa.gov

[From the APRS SIG]

There have been a number of DGPS comments/queries the past few days. Let me try to give some answers.

It started when Skip Allison (K9SA) suggested that a reference station transmit its apparent position and then users apply offsets to their positions -- this is sometimes called map shifting.

Good thought, but it doesn't work very well. Mark Sproul gave part of the answer in his reply, but let me amplify.

Your GPS receiver measures the timing of the 1.023 Mb/s CA code (293 meters per timing bit) with respect to its internal (and crude) xtal clock. In common practice, this is expressed as an apparent distance to the satellite, called a pseudo-range (PR) with the "pseudo" indicating that the clock is really unknown. By combining 4 PRs the receiver can compute a 3-D position and the error in the clock. Similarly the receiver measures the apparent frequency offset of each of the 4 satellites and expresses this as a pseudo-doppler speed, called a pseudo-range rate (PRR). The four PRRs yield a determination of your 3-D velocity and the frequency error of the receiver's local oscillators.

Nowadays, every receiver you use has multiple channels and uses more than the minimum of 4 satellites to improve accuracy. It also makes use of the integrated PRR data to help smooth the PR data (called carrier-phase smoothing). This is done in a multi-state mathematical black box (BB), usually a Kalman filter. If you are observing N satellites (N4) and averaging the PR & PRR data over T seconds, you have something like (2 for PR+PRR data)*(N satellites)*(T seconds) of data to produce the same 8 numbers -- i.e. the solution is over-determined. The solution depends on precisely which/how many satellites are in use, and on the recent history of the measurements, and on the precise details of the algorithms in the BB, and on the effects of noise, multipath, etc. on the raw data. No two receivers will have a perfect match of all these parameters, so the results of two receivers will not match!

If you doubt this, I suggest you try the experiment many of us have done. Put two receivers side-by-side (or if you have an external antenna and a microwave power splitter, let them share the same antenna). Turn them on and log the position for a couple of hours. Compare point-by-point the results and you will see that the receivers produce different "ground tracks," with the differences at levels of tens of meters. In this test, if the receivers are a different breed, or if they are not activated at precisely the same time, or if the signals going into one receiver are interrupted for a few seconds you will find that the differences may be even bigger, approaching the size of the "wandering" you see in the position measured by either receiver.

The "real DGPS" systems which use the RTCM SC-104 "buzzword" acronym solve this problem differently. The DGPS "master" station is told where it is located. It measures the same PR and PRR for each satellite, but only computes the "zero-D" clock/frequency errors from all the satellites in view. It then transmits the PR and PRR errors it has determined to all its "slaves" in a standard format (called RTCM SC-104).

The slaves then use the PR and PRR differences to correct their raw measurements before feeding them into BB filter algorithm.

How often are these corrections needed? The answer is dependent on the power spectrum of the errors that the DGPS is correcting. The biggest (and most annoying ;{) of the errors comes from the fact that our DoD has chosen to degrade the perfectly good GPS satellite signals in a process known as SA (Selective Availability). They dither the code and carrier phase of the GPS satellites with a pseudo-random sequence known only to them. This causes errors in the PR and PRR measured by your receiver. DGPS works because the "master" is measuring the DoD's dither and telling your receiver how big it is via a radio link. It can only determine the PR and PRR errors based on past history, so the quality of the corrections die with time. Although we don't have the "Magic Cap'n Midnite Decoder Ring" to know the details of the dither, we have been able to measure its power spectrum. The spectrum shows that their modulation extends from ~1 second to ~30 minutes. The short-period part of this creates noise in your velocity determination (this is the reason that your GPS shows 2 MPH speeds when you are stopped) and the cumulative effects over longer times determine your positional accuracy. Most user GPS receivers require a DGPS refresh a few times each minute, and hence the 15 or 20 second beacon intervals we have used.

Answering a few other questions that cropped up -- Arte, N2ZRC asked:

> I take it, BTW, that the IWI-13 experiment has long been concluded, and that busy APRS networks such as we have here in the northeast corridor will not be overly saturated with the minimum necessary 20-second DGPS correction signals. Is this true?

My W3IWI-13 DGPS beacon has made use of a $25,000 Trimble 4000SSE receiver I have at work. It has been on-the-air only when it is not needed elsewhere. We have summertime programs (especially in Alaska) which have prime call on it. Hence it has been off the air for the past few months. I have no idea whether it will be re-activated.

Bob, WB4APR said:

> TAPR is producing a kit that will convert any GPS with precise time output to a differential corrector suitable for just this application.

Not quite correct. The TAPR DGPS base works only with the Motorola ONCORE VP receiver, which happens to also be the best timing receiver. Motorola has had a proprietary DGPS capability in the ONCORE (and the earlier PVT-6) OEM receiver. The TAPR DGPS board, designed by Steve (N7HPR), has a PIC microprocessor that converts from the proprietary format into RTCM SC-104.

Boyd, W2HOB chimed in with:

> Sure... just fork over $499 to Garmin for their DGPS receiver and connect it to the Data In of your mobile GPS receiver, assuming it was "Differential Ready". It receives the Coast Guard corrections on 200~300 KHz.

Yep -- but the reason I did IWI-13 in the first place, and the reason that Steve did the TAPR DGPS project, was that hams are a cheap bunch of bastards! Rather than having everybody shell out $500 for a VLF DGPS box, lets see if we can't do the same thing with amateur reasources on amateur frequencies with the same equipment we are using for other purposes.

Ralph Fowler added:

> However, I just do not see the need. We are doing vehicle location here, for free in the Amateur Service. Why do we need 10 foot accuracy? Even though I have had all the gear sitting here, I can't see a useful purpose in APRS for it. Certainly not any purpose to justify keying my digi every 20 seconds and spewing out this data.
>
> The people putting the pressure are individuals who don't even run APRS. All they want is a "free" source of Diff correction.
>
> Usually my common sense doesn't lead me astray, so can someone give me any good reasons WHY we even NEED DGPS on APRS frequencies?

Personally, I prefer metric units! My reason for bringing up IWI-13, and Steve's reason for developing the TAPR project was to advance the technical state of the art. GPS can do a lot of things for a lot of people in our community. Some of the users really want/need/appreciate the higher accuracy. Have you ever navigated a sailboat back home in the dark in a pea-soup fog?

Personally, I find the repeated keying up of a digi by someone at home who isn't even moving, reporting that he isn't moving, to be even more obnoxious! I could ask the related question "Why do we need APRS at all?"

And the last time I read Part 97, nobody really "owns" any frequencies. It is up to us to use the resources for the good of all facets of the hobby.

Wes Johnson (KD4RDB) brought up "inverted GPS". This is fine if you want to know where your company's truck/bus/train is now. "Big Momma" at home base computes the true location of "rover." But most of us have GPS to find out where we are located autonomously, and inverted DGPS just doesn't hack it.

73, Tom

Vice President Gore Announces New GPS Modernization Initiative

Vice President Gore announced a $400 million new initiative in the President's balanced budget that will modernize the Global Positioning System (GPS) and will add two new civil signals to future GPS satellites, significantly enhancing the service provided to civil, commercial; and scientific users worldwide.

"The United States is proud to be a leader in the development of the Global Positioning System -- a wonderful example of how technology is benefiting our citizens and people around the world," Vice President Gore said. "This initiative represents a major milestone in the evolution of GPS as a global information utility, and will help us realize the full benefits of this technology in the next millennium."

This initiative is only the most recent step in an ongoing public-private effort to make GPS more responsive to the needs of civilian users worldwide. National and regional GPS-based networks are now being created by governments and industry around the world to help guide everything from planes, trains, ships, and cars to tractors, snowplows, earthmovers, and mining equipment.

As announced by Vice President Gore last March, the second civil signal will be located at 1227.60 MHZ along with the current military signal, and will be available for general use in non-safety-critical applications. The President's Budget supports implementing this new signal on the satellites scheduled for launch beginning in 2003.

Key to the overall modernization initiative was a recent White House decision on the frequency for a third civil signal that can meet the needs of critical safety-of-life applications such as civil aviation. The third civil signal will be located at 1176.45 MHZ, within a portion of the spectrum that is allocated internationally for aeronautical radio navigation services, and will be implemented beginning with a satellite scheduled for launch in 2005. This initiative will cost $400 million over six years. The date that new services will be available to users will depend on the actual launch dates, orbiting sufficient numbers of satellites to provide useful services, and maintaining operational capabilities.

When combined with the current civil signal at 1575.42 MHZ, the new signals will significantly improve the robustness and reliability of GPS for civil users, and will enable unprecedented real-time determination of highly accurate position location anywhere on Earth. This new capability will spur new applications for GPS, further expanding the rapidly growing market for GPS equipment and services worldwide.

New Version of HamView

Alberto di Bene, I2PHD
dibene@usa.net

Version 2.10 of Hamview has been uploaded to: http://www.freeyellow.com/members/padan

HAMVIEW is a freeware program that:

- gets an audio signal either from a Sound Blaster(TM) compatible card or from a .WAV file
- performs a windowed spectral analysis on the incoming signal
- displays the signal spectrum on the video screen using a carefully matched color scheme
- performs some enhancement on the signal using signal processing methods
- plays the filtered signal at the output of the audio card
- and last but not least, all these goodies are done in real time! (well, to be honest with about one second of delay from the input to the output, due to the data windows used for signal analysis and processing).

The program was especially developed for radio amateurs interested in all those communication techniques where signals very close to, or below, the receiver noise floor are used, like EME, VLF, and so on.

All the hardware needed is:

- a Pentium computer with MS-DOS or PC-DOS 6.xx or greater (a W95 DOS session is OK). Windows NT is not supported, as the program needs to talk directly to the hardware, bypassing any driver, and this is not allowed under NT.
- a Creative SB16 sound card or better. 100% compatible cards should work, provided that the card is full-duplex compatible.

What's new in version 2.10 ?

- New INI file management (many more settings are now remembered)
- Passthrough mode (useful when tuning, as the audio delay is suppressed)
- Now both output channels are driven (this was a bug in the previous versions)
- The output volume is adjustable with cursor Up and Down keys
- The local time is diplayed at the full minute
- New graphics library, which means more compatibility with various video cards
- The East/West indicator has been moved to the bottom of the display
- Improved .WAV files management
- Medium speed and Fast speed modes added, dynamically changeable from the keyboard
- A few bugs have been exterminated

Enjoy, and, please, do bugs reporting ! Thanks.

Technology Demo Day

TAPR is proud to announce the first annual Technology Demo Day sponsored by the TAPR APRS SIG.

The purpose of this event is to demonstrate technology independence at the location of Field Day. Technologies like Packet Radio, SSTV, ATV, Spread Spectrum, IGates and other operating modes accessed via RF means only will be allowed, since this reflects current emergency situations at disaster scenes in our present day world. The 1999 Technology Demo Day is to coincide with the ARRL Field Day. Full details are available at http://www.tapr.org/tdfd

1999 Technology Demo Day

- Commercial Power Restriction:
  No commercial power is allowed for stations submitting a score.
- Band of operation:
  Any frequencies that are currently allowed for the mode in use. HF, VHF, etc.
- Use of networks and digipeaters:
  Digipeaters, Networks, can be used for contacts. All APRS paths allowed. Sites must use RF (No direct Inet).
- Participants:
  Any licensed amateur radio operator may participate. Participants that are at a direct connection to the net can participate but are not permitted to submit scores.
- Operator Designations:
  Single Operator designated by "SO". Multi Operator by "MO". No Operator by "NO".
- Minimum exchange consists of:
  Call Sign, ARRL section, Operator Designation (SO, MO, NO).
- Submission of scores:
  Do not send your Logs. Send only your score. Send your score to kb2scs@worldnet.att.net or use http://www.tapr.org/tdfd. Scores must be submitted within two weeks of the end of the event.
- Points:

» 25 points for receiving the ARRL message in digital form
» 2 points for every LIVE packet QSO (packet network, conference nodes, digis, etc)
» 1 point for every APRS station "received"
» 2 points for every APRS station exchanging messages
» 25 points for receiving ATV signals from a range over 5 miles
» 25 points for transmitting ATV to another station over 5 miles
» 5 points for every received SSTV picture received
» 5 points for every transmitted SSTV contact
» 25 points for receiving any Satellite digital MESSAGE
» 25 points for transmitting any Satellite digital Message.
» 25 points for making a complete exchange using Spread Spectrum
» 5 points for each Spread Spectrum contact thereafter

Suggested Operating Instructions

- HOOK UP YOUR SPEAKER. If you aren't listening to the packets, then you may not know what is going on the channel. TNC's do not detect collisions. DCD's do not either, so just because the little green light is OFF, does not mean the channel is not all QRM
- Operate normally.
  USE SHORT packets and MINIMUM PATH. Use WIDEN-N where possible. Keep hops practical for your area (any more than 3 is probably impossible). Work everyone you see on mail/traffic pages.
- ACKS are not required!
  If you get a QSL message from the other station, don't wait for an ACK. Erase the outgoing message to reduce QRM!
- Internet Gateways
  The key here is that IGates only pass through MESSAGEs to LOCAL stations in their area. This means you can QSO with anyone if you already know their callsign! You will probably NOT see their posits since only messages go via IGates (normally). This is why you should always include your GRIDSQ in all of your text messages. Here are some ways we have come up with to get around these quirks:

» Assume an SSID of -6 for stations and guess at some APRS calls.
» Watch your traffic list. As soon as anyone QSOs anyone in your area, you will see them for a return QSO.
» Send a CQ FD MESSAGE with your GRIDSQ and report to a DIGI in a known area with an IGate. This will go through all IGates.
» Do a MAPS-OVERLAY-DIGIS for an area to see a good target call.
» Do NOT send CQFD or TARGET messages until you have exhausted all other contacts, else we may totally BLOCK the RF channels around IGates.
» Once you are working via IGATES, PLEASE minimize your path to ONLY the hops needed to hit your local IGate.
» Home Stations: If you are not going to operate, please set up your UN-ATTENDED home station for AUTOREPLY with something like 'QSL. [gridsq].(NO), ARRL section, etc. For APRSdos this simply means send ONE message to the callsign of REPLY and make sure this message is the first in your SEND list.

APRS is registered to Bob Bruninga, WB4APR.

PIC-et Radio: How to Send AX.25 UI Frames Using Inexpensive PIC Microprocessors

John Hansen, W2FS
hansen@fredonia.edu

Abstract

This paper provides step by step documentation of how to implement AX.25 UI frames using inexpensive PIC microcontrollers. It is designed primarily for those who wish to implement packet radio UI beacons for point to multipoint communications. The article assumes some knowledge of programming concepts and PIC microprocessors. It also discusses the limitations that must be overcome in order to build a completely PIC based terminal node controller.

Introduction

Over the past several years there has been a growing number of applications that utilize packet radio communication via UI (unconnected) frames. These applications include the PACSAT broadcast protocol, APRS transmissions, and my own HamWeb file transfer protocol. All of these share the advantage that they provide communication of data from one source to many recipients simultaneously.

It is my view that we have only begun to exploit the possibilities of broadcast data in amateur radio applications. Furthermore, the efficiency of sending data from one source to many recipients is such that it can often overcome the limitations imposed by the relatively low data rates common in amateur radio applications. In short, broadcast protocols can breathe new life into otherwise "slow" data streams of 1200 baud or less. I believe the day will soon arrive that at 1200 baud, UI frames will be the most common means of transmitting amateur packet data.

Currently the principal means of communicating UI frames is to use a standard packet radio terminal node controller. While the price of these had fallen to the $100 range, by using small inexpensive microprocessors for these applications, the cost can be reduced even further.

This paper provides the basic information that is needed to construct AX.25 UI frames using inexpensive, easy to program Peripheral Interface Controller (PIC) chips produced by Microchip Inc. This paper assumes that the reader has some familiarity with PIC chips, how they work and how to program them. For more information on these subjects, see my recent article in QST.[1] PIC microcontrollers are available in single quantities for around $6. In larger quantities they are even less expensive. Using PIC microcontrollers, it is possible to build a device that can take serial data as an input and send it out as AX.25 UI frames using a PIC microcontroller for less than $20.

This paper is intended to be generic in nature. While a large variety of PIC microcontrollers are available, this article does not refer to any one particularly, instead the concepts described here apply to at least the entire series of Microchip midrange (series 16) controllers. Further, I did not want to limit this discussion to any one particular programming language. All of the examples presented in this article are in the C language. I have used this language because I think that the code is much easier to read and understand than is assembly language (the most common means of programming PIC chips). But the basic constructions seen here can be applied in any language (as long as there is an appropriate PIC compiler available).

A Sample Data Stream

For the purpose of this discussion, assume that you want to broadcast the following simple packet:

```
W2FS-4 CQ, RELAY: Test
```

This is a simple frame where the source is W2FS-4, the destination is "CQ", there is one digipeter called "RELAY" and the text to be transmitted is "Test". I selected this frame because it is a fairly simple packet, but includes most of the features that you would want to incorporate in a more complex UI frame. In a real world situation, the source of this data could be almost anything... a GPS data stream, an automated weather station, a DX Cluster, a highway traffic information server or perhaps even a generic file server like HamWeb.

What you see above is the packet as it would appear on your TNC. There is actually somewhat more to it than this. The AX.25 protocol divides the packet into seven sections as follows:

1. Flag (s)
2. Address
3. Control
4. Protocol Identifier (PID)
5. Information
6. Frame Check Sequence
7. Flag (s)

Most of the information below describing these fields is a summary of what appears in the AX.25 protocol description.[2] I have simplified this material to only describe what you need to know to send UI frames.

Flags

The flags are simply the hex value 7E (01111110 in binary) sent over and over again when no information is being transmitted. For example, when you set the TXdelay on your TNC to some value, it sends flags (7E's) over and over again for that period. These flags provide the receiver with a clear indication of when one packet has ended and the next is beginning. Thus, there must be at least one flag between two adjacent packets.

Address

The address field contains the destination (CQ, in this case) the source (W2FS-4 in this case), and up to eight digipeaters (only RELAY, in this case). Each of "callsigns" in the address field must contain exactly 7 characters, six for the callsign and 1 for the SSID. If a callsign is less than 6 characters long, it must be padded with blanks. In addition, the receiving station must have some way to determine when the address field has ended (since it could have anywhere from 0 to 8 digipeaters in it). This is handled by shifting each of the bits one position to the left so that a 0 appears as the least significant bit. This bit is then set to a one for the SSID (seventh byte) of the last callsign in the address field. For example, the destination callsign would be encoded as follows:

Character	HEX Value from ASCII Table		Shifted Hex Value	
C	43	(01000011)	86	(10000110)
Q	51	(01010001)	A2	(10100010)
Space	20	(00100000)	40	(01000000)
Space	20	(00100000)	40	(01000000)
Space	20	(00100000)	40	(01000000)
Space	20	(00100000)	40	(01000000)

The seventh byte, the SSID, is a bit more tricky. Use the following bit pattern: 011SSSSx where SSSS is the SSID (in binary) and x is a 0 if this is not the last callsign in the address field and a 1 if it is the last callsign in the address field. Since the destination address in this case has no SSID (that is, the SSID =0) and it is not the last callsign, the value should be: 01100000 or 60 in hex.

The next callsign is the source callsign, which in this case is W2FS-4. Using the rules established above we get the following string of bytes for this callsign:

W	2	F	S	Space	Space	SSID = 4
AE	64	8C	A6	40	40	68 (01101000)

There is only one digipeter so it is necessary to set the least significant bit of the SSID of that callsign to 1:

R	E	L	A	Y	Space	SSID = 0
A4	8A	98	82	B2	40	61 (01100001)

Since UI frames are generally broadcast and not directed at any station in particular, the destination space is often wasted with something fairly meaningless (like CQ or APRS). The APRS MIC-E protocol capitalized on this by actually encoding useful position information within the destination address.

Control and PID

Since we are only doing UI frames, the control and PID bytes are pretty simple. Always use the hex value 03 for the control byte and the value F0 for the PID byte.

Information

In this case the information being conveyed is simply the word "Test". This consists of 4 hex bytes as follows:

T	e	s	t
54	65	73	74

The only thing to be careful about here is that these values are NOT shifted to the left as the address bytes are.

Frame Check Sequence

I was rather disappointed with the description of the Frame Check Sequence that is contained in the otherwise excellent AX.25 Protocol definition published by the ARRL. Here is what it says:

> The frame-check sequence (FCS) is a sixteen bit number calculated by both the sender and receiver of a frame. It is used to insure that the frame was not corrupted by the medium used to get the frame from the sender to the receiver. It shall be calculated in accordance with ISO 3309 (HDLC) Recommendations.[3]

Fortunately there is considerable information on the Internet concerning the calculation of this value, which is more often referred to as a "CRC" than an FCS. In addition there is an excellent article in the September, 1986 issue of Byte Magazine on this subject.[4] Rather than review the theory of doing CRC calculations, I will provide a relatively simple mechanism for calculating this value, implemented in assembler and C. This code is included in Figure 4.

Putting It All Together

Aside from the flags and the FCS (which will be calculated as we go along), our test packet can now be implemented as an array of 27 hex bytes as follows:

C	Q	Sp	Sp	Sp	Sp	SSID=0
86	A2	40	40	40	40	60
W	2	F	S	Sp	Sp	SSID=4
AE	64	8C	A6	40	40	68
R	E	L	A	Y	Sp	SSID=0
A4	8A	98	82	B2	40	61
Control	PID	T	e	s	t	
03	F0	54	65	73	74	

Or, as an initialized C array:

```
byte SendData[27] = {0x86,  0xA2,  0x40,  0x40,  0x40,
0x40,  0x60,  0xAE,  0x64,  0x8C,  0xA6, 0x40,   0x40,
0x68, 0xA4, 0x8A,  0x98,  0x82,   0xB2,  0x40,   0x61,
0x03, 0xF0,   0x54,   0x65,  0x73, 0x74}
```

Doing It with a PIC Microcontroller

So aside from putting a few flags on the ends and adding the FCS, one simply needs to transmit this set of bytes in order to transmit the UI frame. There are a number of ways that this can be done. First, it is possible to get the PIC itself to actually send the data by simulating a sine wave with something called a resistor ladder. The theory is as follows. You take four (or more) output pins from the PIC and connect four (or more) different value resistors to them. You connect the other ends of these resistors together. Then you step through the output pins to get a rising and falling voltage that simulates a sine wave. This method is documented in a Microchip application note.[5]

A second method is to feed a data stream out of the PIC to a modem chip and have the modem chip generate the tones. There are a number of tone producing chips that will work for this application. Traditionally, packet TNCs have used the Texas Instruments TCM3105. Because this chip is no longer being produced and is getting harder to find and more expensive, I didn't want to start down this road. Instead, I've been experimenting with the MX-COM MX614 chip, which cost about five dollars. It works quite well for this application. It appears that the Philips PCD3311/ PCD3312 chips will also work in this application and cost around $1.50, but I've not had occasion to use them yet.

Even using a modem chip to generate the tones, you can't simply send serial data out a PIC pin and expect it generate packets that can be decoded by other TNCs. There are three reasons for this:

1. Serial data contains start and stop bits. These are not used in packet radio transmissions.

2. The AX.25 protocol specifies that if five consecutive 1's are received in a row, except in a flag byte, a zero should be added after the string of five ones. This is referred to as "bit-stuffing". If you are constructing the AX.25 frames yourself, you are responsible for bit stuffing.

3. Packet radio uses a modulation scheme called NRZI (Non-Return to Zero, Inverted). This means that the ones and zeros are not represented by high and low states (or tones). Rather, a zero is represented by a change in tone (if it was high, it goes low, if it was low, it goes high) while a one is represented by no change in tone. Together with bit-stuffing, this ensures that there will be a tone change at least every five bits, if not more often (except for flags).[6] This helps the transmit and receive timing stay in sync.

In order to implement this in a PIC microcontroller, therefore, you must take the incoming datastream, calculate the FCS, add the flags and route the stream of data to a subroutine that handles the transmission of the data taking into account the proper timing of the bits, bit stuffing, and NRZI. It's a tall order, but it can be easily handled by a $6 PIC chip!

On to the Code

Here is some stripped down C code that will send our sample array as an AX.25 UI frame. This code is written specifically for use with the PIC C compiler made by CCS, Inc. It is the least expensive C compiler currently available ($99). As noted above, the same logic flow could be applied to sending UI frames using assembler. An excellent assembler (MPASM) is available from Microchip, Inc. free of charge. Starting with an overview, the SendPacket() function shown in Figure 1 will send the packet that is contained in the array SendData.

At the heart of the SendPacket() function is the SendByte() function which is called to send each of the 27 bytes in the SendData array. The SendByte() function is shown in Figure 2.

Note that for each byte, the data is transmitted least significant bit first (that is from right to left, rather than from left to right). The function delay_us is a routine shipped with the CCS C compiler. It is supposed to create a delay of 850 microseconds. You might think that this is too long a period a time since 1200 baud would normally require that each bit last exactly 833 milliseconds (1 sec/1200). The CCS timing routine is not exactly accurate, however. Experimentation revealed that a value for the CCS function of 850 resulting in timing that is correct for 1200 baud.

```
void SendPacket(void){
   fcslo=fcshi=0xFF;    //The 2 FCS Bytes are initialized to FF
   stuff = 0;           //The variable stuff counts the number of 1's in a row. When it gets to 5
                        // it is time to stuff a 0.
   output_high(PTT);    //Turns on the microcontroller Pin that controls the PTT line.
   flag = TRUE;         //The variable 'flag' is true if you are transmitting flags (7E's)
   fcsflag = FALSE;     //The variable 'fcsflag' is true if you are transmitting FCS bytes
   for (i=0;i<20;i++) (SendByte(0x7E));    //Sends flag bytes.  Adjust length for txdelay
                                            //each flag takes approx 6.7 ms
   flag = FALSE;                            //done sending flags
   for(i=0;i<27;i++) (SendByte(SendData[i]));      //send the packet  bytes
   fcsflag = TRUE;                      //about to send the FCS bytes
   fcslo =fcslo^0xff;                   //must XOR them with FF before sending
   fcshi = fcshi^0xff;
   SendByte(fcslo);             //send the low byte of fcs
   SendByte(fcshi);             //send the high byte of fcs
   fcsflag = FALSE              //done sending FCS
   flag = TRUE;                         //about to send flags
   SendByte(0x7e);              // Send a flag to end packet
   output_low(PTT);                     //unkey PTT
  }
```

Figure 1

```
void SendByte (byte inbyte){
   int k, bt;
   for (k=0;k<8;k++){              //do the following for each of the 8 bits in the byte
     bt = inbyte & 0x01;           //strip off the rightmost bit of the byte to be sent (inbyte)
     if ((fcsflag == FALSE) & (flag == false)) (fcsbit(bt));   //do FCS calc, but only if this
                                     //is not a flag or fcs byte
     if (bt == 0) (flipout());       // if this bit is a zero, flip the output state
        else {                       //otherwise if it is a 1, do the following:
           stuff++;                  //increment the count of consequtive 1's
            if ((flag == FALSE) & (stuff == 5)){          //stuff an extra 0, if 5 1's in a row
                   delay_us(850);    //introduces a delay that creates 1200 baud
                   flipout();        //flip the output state to stuff a 0
            }//end of if
        }//end of else
     inbyte = inbyte<<1;            //go to the next bit in the byte
     delay_us(850);                 //introduces a delay that creates 1200 baud
  }//end of for
}/end of SendByte
```

Figure 2

The flipout() function changes the state on the output pin when a zero is being sent. It is shown in Figure 3.

Finally, we need the routine that actually calculates the FCS. The FCS consists of two bytes, which I have called fcslo and fcshi. These are both initially set to FF. In this example the FCS will be calculated on a bit by bit basis. Algorithms exist that can calculate the FCS either a bit at a time or a byte at a time. Figure 4 shows the calculation routine.

```
void flipout(){     //flips the state of output pin a_1
  stuff = 0;        //since this is a 0, reset the stuff counter
  if (!bit_test(port_a,1)) (output_high(pin_a1));   //if the state of the pin was low, make it high.
     else (output_low(pin_a1));         //if the state of the pin was high make it low
}
```

Figure 3

The function parameter tbyte is either the byte 0000 or the byte 0001 corresponding to the value of the bit that is currently being transmitted. I have used three assembly language instructions in the beginning of this function (between #asm and #endasm) because there is no simple means of rotating a 16 bit value in the CCS C implementation. These three assembly language instructions simply move the 16 bit value one place to the right, with the previous least significant bit being placed in bit 0 of the status register in the PIC chip. The next line (the line that begins with if) performs an exclusive-or (XOR) on this bit from the status register and the bit that is being transmitted (tbyte). If the result is equal to 1, the FCS is XOR-ed with the hex value 8408. If the result is equal to 0, this latter step is not performed. Either way, the new value of the FCS is preserved in fcshi and fcslo. This procedure may seem rather arcane, but it does work. For a discussion of the theoretical reasons behind this procedure see the 1986 Byte Magazine article.

A PIC-based TNC?

From the above discussion it is clear that it is not all that difficult to send AX.25 frames using a PIC chip. There are a wide range of beacon type applications where such a device could be very useful. To go one step further, does this mean we could build a PIC-based TNC for very little money? Unfortunately this is not a trivial matter. My near term goal is to find a way to build a 1200 baud transmit module that will take a continuous data stream, convert it into packets and transmit it. A similar module on the other end of the link would undo the process. Using this mechanism you could transmit virtually any

```
void fcsbit(byte tbyte)
{
#asm
   BCF     03,0
   RRF     fcshi,F        // rotates the entire 16 bits
   RRF     fcslo,F        // to the right
#endasm
   if (((status & 0x01)^(tbyte)) ==0x01){
         fcshi = fcshi^0x84;
         fcslo = fcslo^0x08;
      }
 }
```

Figure 4

serial data stream from one point to many points using existing amateur radio transceivers without conventional TNCs. My intermediate term goal is to build a full duplex stand-alone 1200 baud (and then 9600 baud) KISS TNC using these inexpensive chips. If this could be accomplished it would have a myriad of applications including very inexpensive 9600 baud amateur satellite modems.

There are a number of hurdles to be overcome before any of this is possible. Starting on the transmit side, the basic problem is that the device must receive data via a serial link at the same time that it is transmitting data over the radio. The beacon style device discussed in this article takes existing data that it obtained from whatever source and transmits it using AX.25. For the purposes of this device, it is assumed that the data stream is not continuous. If the input data stream is continuous, however, while it is transmitting the first packet, it must also be accumulating data for the second packet over the serial link. Some buffering would also be required since there is not a bit for bit correspondence between the serial data stream (which includes start and stop bits) and the radio data stream (which includes no start and stop bits, but does include addresses, PID and control bytes, the FCS, etc.).

The receive side may actually be a bit more difficult. This is because the incoming packet must be received in its entirety in order to calculate the FCS before it is forwarded out the serial port since packets with incorrect FCSs should be discarded. Thus there must be enough buffer space to hold the entire packet. Most packet applications, including all of the amateur digital satellites, are limited to PACLENs of 255 characters. There are PIC microprocessors with enough on board memory to handle this available in the $10 range. However, some protocols are now using packet lengths in excess of 1K. No PIC contains this much on-board storage, so some external SRAM would be required. This, in turn would involve using a PIC with a substantial number of I/O pins (for both the data and address lines).

Conclusion

Contrary to popular opinion, the most significant limitations in packet radio today are not technological. Amateur radio operators are only beginning to scratch the surface of the range of things that can be accomplished with the technology that is already available. In addition to the quest for faster speeds, we should also focus on new applications that can be developed with the slower speed digital technologies that can piggyback on conventional FM radio channels. One key to doing this is to develop extremely inexpensive packet radio interfaces. PIC chips can provide a means of doing this.

Notes

1. Hansen, John A., Using PIC Microcontrollers in Amateur Radio Projects (QST, October, 1998).
2. Fox, Terry L., AX.25 Amateur Packet-Radio Link-Layer Protocol, Version 2.0 (Newington, Ct: ARRL, 1984). ibid., p.4.
3. Morse, Greg, Calculating CRCs by Bits and Bytes (Byte, Sept 1986, pp. 115-124).
4. Microchip Application Note 655, D/A Conversion Using PWM and R-2R Ladders to Generate Sine and DTMF Waveforms .
5. www.microchip.com/Download/Appnote/Category/16CXX/00655a.pdf
6. For a more complete description see: McDermott, Tom, Wireless Digital Communications: Design and Theory. (Tucson:TAPR, 1966) pp. 121-126.

Update on GPS and APRS related offerings

It has been a while since we touched on all the various GPS and APRS stuff at TAPR. Here is the overview:

The APRS SIG has a new web page: http://www.tapr.org/tapr/html/aprssig.html On this page you will find links to all the APRS-related software stored on TAPR. We are now providing file repositories for:

APRSdos
MacAPRS
WinAPRS
javAPRS
APRS PLUS
Waypoint
palmAPRS

As for GPS Kits/Products

Garmin GPS-25 (OEM engine)
Motorola Oncore VP (OEM engine)
TAC-32 Software Registration
Garmin Data Cable (for the GPS-25)
MCX RF Pig Tail (to get from your engine to the rest of the world)
Garmin GA-27 Antenna with MCX connector
Motorola Antenna 97 with BNC connector
TAPR DGPS Differential GPS
Garmin GPS-20/25 Power and Signal Interface
Oncore VP Power and Signal Interface
APRS MIC-E Radio Mic Encoder
TAPR PIC-E PIC-Based Encoder
TAPR TAC-2 Totally Accurate Clock

All these can be found at: http://www.tapr.org/tapr/html/gpsf.html

Spread Spectrum with a PIC

Steve Bragg, KA9MVA
steveb@skymaster.c2-tech.com

PIC DS1

This is one of the whacky spread spectrum transmitter experiments I did one day back in September, 1997.

In this experiment, I had the following goals:

- Transmit direct-sequence spread-spectrum of a Manchester-encoded data stream
- 1 MHz chip rate with a 4 MHz PIC16F84
- Manchester encoding in software, at a rate of 3.89k bits per second (1/254th of the chip rate). Data rate is half that.
- Burst mode -- data sent in short packets (so that there would be time to load new data before SS transmitter turned back on)
- Pseudorandom binary spreading (PRBS) code generated by 7-bit linear feedback shift register (LFSR), using the taps specified for ham use (127-bit maximal-length sequence).

Hardware Notes

The hardware implementation is quite simple. There is a PIC16F84 in a fairly standard configuration. The PRBS sequence is output on RB0, and the Manchestered TX data is output on RA0. There's a debug pin which toggles for each data byte sent, providing a convenient trigger for the scope.

The PRBS code was XOR'ed with the Manchester-encoded data using an external XOR gate. I could have performed this function in software, too, but I wanted to see the PRBS code on the oscilloscope.

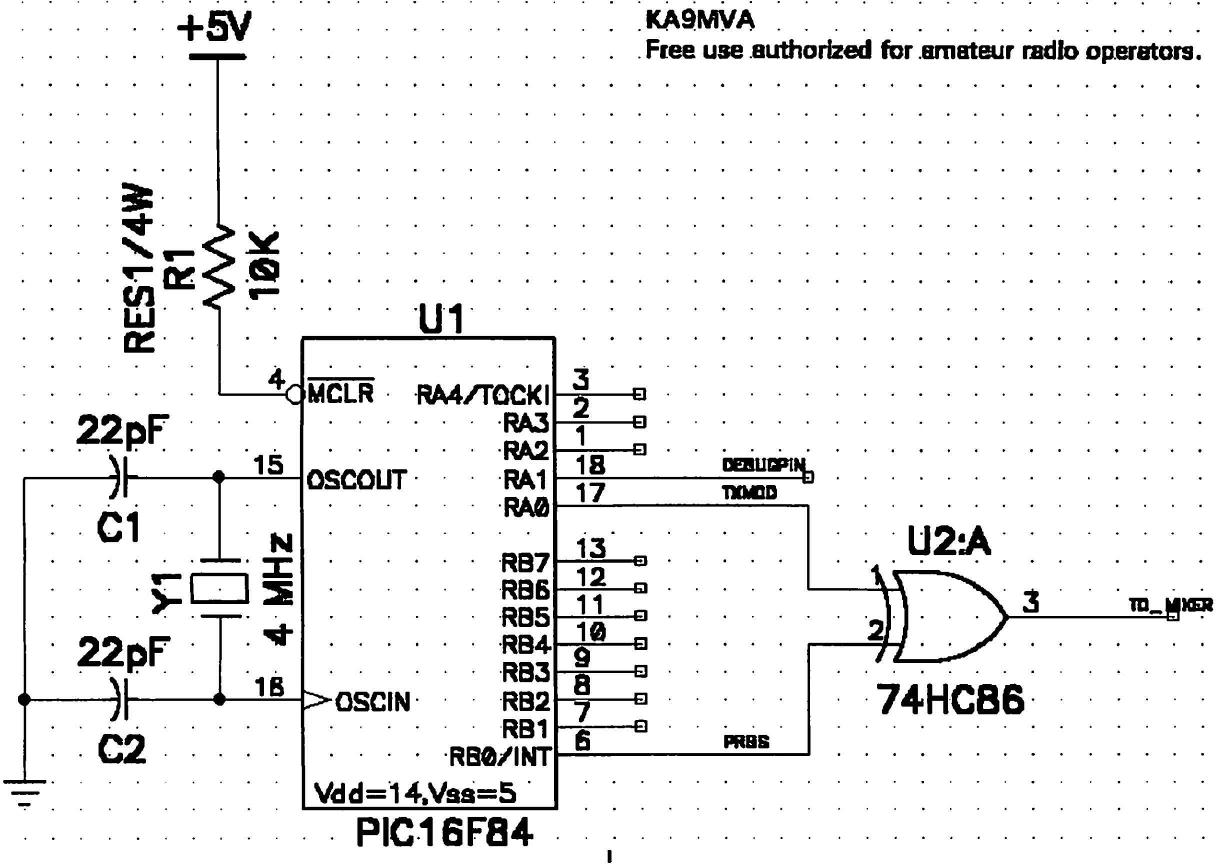
The spread Manchestered data stream is then mixed with the carrier frequency, using a double-balanced mixer from Mini-Circuits. Sorry, it's been awhile, and I forget the part number. I used 2.4 GHz for testing, although this could work at 900 MHz, too.

I didn't give much thought to how this stream would be synchronized to or received. I think that the 4 MHz PIC crystal should be used to reference the transmitter PLL, so that the transmit carrier frequency is some multiple of the PRBS bitrate. This allows a "sliding correlator" (frequency-locked-loop) to be used on the receiver. After the data is de-spread, it can be de-Manchestered fairly easily.

Firmware Notes

The PIC software implements the 7-bit LFSR by "unrolling" the code and "bit-banging" it on one of the port pins. There is one instruction (at least) for each bit of the PRBS code.

I used a BASIC program to generate the "unrolled" PRBS code. I misplaced that, but you can implement a linear-feedback shift register, right?



KA9MVA
Free use authorized for amateur radio operators.

The firmware depends on there being "wide places" in the PRBS code -- several bits of the same value in a row. It uses these places to perform the Manchestering, and the required looping and braching. It depends on the very deterministic timing of the PIC processor's instructions.

The firmware sends a fixed-length packet (in the demo, it was 8 bytes of the constant 0x55). It repeats it a fixed number of times for redundancy. The Manchester clock is toggled once per 127-bit run of the PRBS sequence. The data is clocked at half that rate.

Since the PRBS code is fixed in the firmware, you really couldn't call this Code-Division Multiplexing. However, it is still Direct-Sequence Spread Spectrum.

Here is the heavily-commented source code which is also available at:
www.c2-tech.com/~steveb/ka9mva/picds1.asm

```
; Spread-spectrum-on-a-PIC demonstration
; September 17, 1997
; Steven D. Bragg
;
; This program outputs Manchester-encoded synchronous data while at the same
; time implementing a 1-megachip-per-second pseudonoise generator in SOFTWARE !
; This is the definition of the PIC's ultimate capability at 4 MHz.


LIST    P=PIC16F84
#include <P16F84.INC>

__CONFIG _XT_OSC & _PWRTE_ON


outbit equ H'000C'      ; Used to output Manchester data
mclk equ H'000D'        ; Manchester clock state
bitcnt equ H'000E'      ; number of bits sent in the byte so far (counts down)
rptcnt equ H'000F'      ; number of times message has been repeated for i
                        ; redundancy (counts down)
msgbuf equ H'0010'      ; start of message buffer:  8 bytes long
msgend equ H'0018'      ; end of message buffer
repeat equ 3            ; number of times to repeat message for redundancy
#define TXMOD PORTA,0
#define PRBS PORTB,0
#define DEBUGPIN PORTA,1


        org     0       ;start address 0
        goto    start


        org         0x08
start   clrf    PORTA               ; Clear port A
        clrf    PORTB               ; Clear port b
        bsf     STATUS,RP0
        clrf    TRISA               ; Make port A outputs
        clrf    TRISB               ; Make port B outputs
        bcf     STATUS,RP0


; Direct-sequence spead-spectrum packet transmit
; The code will Manchester-encode and transmit a fixed length, 8-byte message
; along with the ham 127-bit sequence required to spread it at a 1-megachip rate


;       7-bit linear feedback shift register (127-bit maximal-length sequence)
;
;       01000001101010001100101110110000111010111100111110111111
;       00000001010101100110001000101101001110010000100101001001;
;       101101110001111
;
;       The entire pseudonoise sequence is implemented as a list of
;       bsf's and bcf's.  The only way that other work gets done
;       (i.e., Manchestering/sending data) while generating PN is
;       that there are bunches of 1's and 0's I call "wide places".
;       These wide places only reqire an instruction to initiate and
;       terminate them; during this time, the code can do other work.
;
;       Note!  This code assumes:
;       1.  No interrupts occur during transmit.
;       2.  No subroutines are called during transmit (careful modifying!)
;       3.  It has exclusive use of the file select register (FSR).
;       4.  The PIC is run at 4 MHz for 1 megachip/s.
;       5.  The Manchester encode rate is 1/254· the chip rate.
;
;       This code is EXTREMELY time sensitive.  Be careful modifying it.  Even 1
;       cycle off will cause the psedorandom chipping sequence to be incorrect.
        movlw   H'55'               ; 55 will be the test send pattern
        movwf   msgbuf              ; load it into the message buffer
        movwf   msgbuf+1
        movwf   msgbuf+2
        movwf   msgbuf+3
        movwf   msgbuf+4
        movwf   msgbuf+5
        movwf   msgbuf+6
        movwf   msgbuf+7
        movlw   msgbuf                  ; Get address of message buffer
        movwf   FSR                     ; to file select register
        clrf    mclk                    ; Reset Manchester clock
        movlw   8
        movwf   bitcnt                  ; 8 bits per byte
        movlw   repeat
        movwf   rptcnt                  ; reset repeat count
        bsf     DEBUGPIN                ; pulse debug pin so we know
        bcf     DEBUGPIN                ; it's the start of the sequence
        bsf     TXMOD                   ; Set transmit high to get ready

;       If this were a real transmitter, this would be the place to
;       turn on the PA, PLL, etc.

;       This is the beginning of the sequence
seq     bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1

;       Here is a "wide place" in the sequence.  There are five (5) zero bits.
;       There are four (4) instruction cycles available to do something.  We
;       Manchester-encode the data bit to be sent (LSB first) with the clock.
wide1   bcf     PRBS                    ; 0
        nop                             ; (1 cy)
        movf    INDF,w                  ; (1 cy) Get byte that's being xmitted
        xorwf   mclk,w                  ; (1 cy) Manchester encode
        movwf   outbit                  ; (1 cy) Data bit in LSB of outbit
;       Continue sequence
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0

;       This is a "wide place" in the sequence.  There are five (5) one bits.
;       There are four (4) instruction cycles available to do something.
;       If clock state is zero, prepare the next bit in the current byte
;       to be sent.  We actually shift the data in the buffer, which does not
;       destroy it since a byte rotated 8 bits is unchanged.
;       Note that the clock toggles at twice the data rate.
wide2   bsf     PRBS                    ; 1
        btfsc   mclk,0                  ; (2 cy) is clock state currently zero ?
        goto    wide2a                  ; (2 cy) no, just continue
        rrf     INDF,f                  ; (1 cy) yes, move next bit into place
        decf    bitcnt,f                ; (1 cy) Decrement bit count

;       Continue sequence
wide2a  bcf     PRBS                    ; 0
;       This is a "wide place" in the sequence.  There are six (6) one bits.
;       There are five (5) instruction cycles available to do something.
;       Send inverted Manchester-encoded data to the TXMOD port bit.
;       This section takes 5 cycles in any case.
wide3   bsf     PRBS                    ; 1
        btfsc   outbit,0                ; (2 cy) Is it a one?
        goto    wide3a                  ; (2 cy) Yes, jump
        bsf     TXMOD                   ; (1 cy) No, send a 1
        goto    wide4                   ; (2 cy) and continue
wide3a  bcf     TXMOD                   ; (1 cy) send a 0

;       This is a "wide place" in the sequence.  There are seven (7) zero bits.
;       There are six (6) instruction cycles available to do something.
;       Here, we decrement the bit counter.  If it zero, it is reloaded,
;       and the message pointed is incremented.
wide4   bcf PRBS                        ; 0
        nop                             ; (1 cy)
        movf    bitcnt,f                ; (1 cy) Check bit count
        btfss   STATUS,Z                ; (2 cy) Is it zero ?
        goto    wide4c                  ; (2 cy) No, just continue
        bsf     bitcnt,3                ; (1 cy) Yes, reload bitcnt to 8
                                        ;        (by setting bit 3)
        incf    FSR,f                   ; (1 cy) next byte
;       Continue sequence
wide4c  bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
;       This is the beginning of the sequence (again)
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
;       Here is a "wide place" in the sequence.  There are five (5) zero bits.
;       There are four (4) instruction cycles available to do something.
;       Here, we toggle the Manchester clock, and check for the end of message.
wide12  bcf     PRBS                    ; 0
        movlw   1                       ; (1 cy)
        xorwf   mclk,f                  ; (1 cy) Toggle Manchester clock
        movf    FSR,w                   ; (1 cy) Get file select register
        sublw   msgend                  ; (1 cy) Check for end of message
;       Continue sequence
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0

;       This is a "wide place" in the sequence.  There are six (6) one bits
;       There are five (5) instruction cycles available to do something.
;       Following a"zero" bit, there is another "wide place" in the
;       sequence.  There are seven (7) zero bits.
;       There are six (6) instruction cycles available to do something.
;       Here, we decrement the message repeat count.  If we have transmitted as
;       many redundant copies of the message as desired, we exit to shut the
;       transmitter down.
wide22  bsf PRBS                        ; 1
        nop                             ; (1 cy)
        btfss   STATUS,Z                ; (2 cy) Is the message finished ?
        goto    wide32                  ; (2 cy) No, continue
        decf    rptcnt,f                ; (1 cy) Decrement repeat count
        nop                             ; (1 cy)
```

```
wide32  bcf PRBS                        ; 0
        btfss   STATUS,Z                ; (2 cy) Will we repeat the message ?
        goto    done                    ; (2 cy) No, we're outta here!
        movlw   msgbuf                  ; (1 cy) Yes,
        movwf   FSR                     ; (1 cy) Reset message pointer
        nop                             ; (1 cy)
        nop                             ; (1 cy)
wide32c bsf PRBS                        ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
        bsf     PRBS                    ; 1
        bsf     PRBS                    ; 1
        bcf     PRBS                    ; 0
;       there are 3 ones here
        bsf     PRBS                    ; 1
        bsf     DEBUGPIN
        bcf     DEBUGPIN                                ; For debug
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0
        bcf     PRBS                    ; 0

;       Here is the last "wide place" in the sequence.  There are four (4) one bits.
;       There are three (3) instruction cycles available,
;       enough time to jump to the top of the loop.
        bsf     PRBS                    ; 1
        nop                     ; (1 cy)
        goto    seq                     ; (2 cy)

;       We are finished sending.  Time to stop spreading and turn the TX off.
;       (debug:  do it all again!)
done    goto start
        end
```

DeLorme Earthmate GPS Units Unuseable for APRS

James Alderman, KF5WT
james.alderman@ericsson.com

Attention fellow APRS enthusiasts and amateur balloon chasers: The new DeLorme Earthmate GPS unit is useless for our purposes. That's because the data from this unit comes out in a non-standard proprietary format -- not the standard NMEA format.

I got two of these units to use for APRS, only to find out that they are of no use to me. The Tripmate, which has been discontinued and has been replaced by the Earthmate, did send its data out in the NMEA format.

The fact that this unit uses a non-standard data format isn't mentioned in the sales literature, or on DeLorme's web site (www.delorme.com). I only found out by reading a product review in the Dallas Morning News a couple of days after receiving my units. Then I called DeLorme's tech-support people just to be sure.

To their credit, DeLorme did agree to take the units back and refund my money when I explained that the units were useless to me. I'm sure the Earthmate is a good product to use for planning trips, it just can't be used for APRS. The mapping CD that comes with the Earthmate (Street Atlas 6.0) will work with both standard NMEA data and this proprietary format as well.

For more information, consult the DeLorme web site or call them at 1-800-452-5931. The technical support number is 207-846-8900.

And as for me, I'm still looking for an inexpensive and lightweight GPS engine that outputs data in the standard NMEA format. If anybody has suggestions, I would love to hear from you.

Changes at Agrelo

[From the ARRL Letter]

Changes at Agrelo: Effective immediately, Agrelo Engineering of Pattersonville, New York -- manufacturer of the DF Jr direction-finding unit--says it has appointed SWS Security of Street, Maryland, as exclusive distributor for Agrelo's DF and transmitter products. "SWS will be responsible for all sales and technical support for these items and will be setting up the dealer network, handling advertising, and all other responsibilities," say a statement on Agrelo Engineering's Web site, http://www.agrelo.com. Agrelo President Joe Agrelo, N2OOC, apologized for problems with delivery and support of Agrelo Engineering amateur products and says his company in the future will concentrate on the commercial market and "divest ourselves of sales and support" for its amateur line. In business since 1972, SWS Security is run by Steve Uhrig, WA3SWS, whom Agrelo described as "a ham, an experienced international businessman and a designer of professional DF systems." Agrelo asked for the amateur community to be "patient" during the transition. He said any product or service "still owing as of November 1998" would be shipped from Agrelo "as time permits," but that all inquiries should go to SWS, 1300 Boyd Rd, Street, MD 21154-1836; tel 410 879-4035; e-mail sales@swssec.com; http://www.swssec.com. SWS says it is now taking orders for Agrelo products. Agrelo said SWS would be releasing upgrades, options, and enhancements to the DF Jr as well as new accessories and complete DF systems.

Kenwood TH-D7A Firmware Upgrade

Kenwood Communication Corporation
http://www.kenwood.net

In an ongoing effort to provide industry leading support for our products, Kenwood has released Service Bulletin ASB-1063. This bulletin can be found on Kenwood's FTP site. Kenwood would like to express appreciation to customers who have assisted in discovering software improvements that have been fixed in the new firmware. We are happy to report that new TH-D7A radios recently shipped to Authorized Kenwood Dealers have the new firmware upgrade already.

Existing TH-D7A owners can send radios in for the upgrade to Kenwood Service Center West ONLY, since special equipment is required to perform the work. If your normal operation pattern is not affected by situations needing an upgrade, you may consider waiting for any future upgrades that may be considered.

Kenwood realizes the potential of customer participation for improving our products. Comments on the newsgroups and listservers are welcomed.

You can download Service Bulletin ASB-1063 using the direct path listed below:
ftp://ftp.kenwood.net/e%3A/kenwood/kenftp/kenwood /Amateur/AmateurServiceBulletins/

If you are using Microsoft's Internet Explorer and have difficulty getting to the FTP site try this address:
http://208.197.91.81/Amateur/AmateurServiceBulletins/

This latest bulletin is in the .PDF format and requires the Adobe Acrobat Reader to view it. In the near future all currently listed bulletins will be converted to the .PDF format for faster downloads and clearer printable copies. This bulletin only applies to TH-D7A transceivers sold in the U.S.A. This information may not be valid for other TH-D7 equipment sold in Europe or other markets.

APRS Needs Your Help

Bob Bruninga, WB4APR
bruninga@nadn.navy.mil

"Up Front" in the March *QST* published a natiownide map of the 500 APRS digipeaters. But the whole meaning/intent of the original submission was lost in the editing. The title and content was supposed to be:

APRS Needs Your Help!

Even if you are not into APRS, you might be able to help your fellow HAM travelers through your area of the country. An analysis of all the APRS digipeaters in the USA shows the following statistics:

EAST of Mississippi:
82% of population is covered by APRS
52% of the area is covered

WEST of Mississippi:
57% of the population is covered by APRS
28% of the area is covered.

Everyone is going Wireless! APRS is "wireless data" for HAM Radio.

For travelers, as long as you are in the APRS coverage area, you have nationwide tracking and 2-way messaging capability (Even in your pocket with the Kenwood Data HT!). Our OBJECTIVE is to at least cover the major thoroughfares for travelers. Notable weak areas in the east are:

South Georgia
E.Central VA (Richmond, Norfolk)
Eastern NC
Western PA
Mississippi,
Southern IL/IN
Western KY
North Central NY state
Maine

All it takes for an APRS digipeater is a TNC and a radio on 144.39. If packet has died in your area, or for any other reason you have an available high antenna and TNC, we could sure use you on 144.39. Come join us. For more info see www.aprs.org or www.aprs.net

Also, the statement in the photo about using Satellites in the West was originally written as an objective, not a fait accompli. Although with last week's APRS/MIR School tests, many stations exchanged packets via MIR using only their HTs and rubber duck antennas.

APRSat

Bob Bruninga
bruninga@nadn.navy.mil

Well until 10 minutes ago, there was an APRSat called NATSWEB scheduled for Launcher integration in 2 days and launch in 3 weeks. We have been working almost around the clock since XMAS when we were offered a free Launch if we could build a fully qualified satellite in 6 weeks... We did!

Here is what it had:

A launch to geostationary transfer orbit on the equator at 36,000 km APogee by 660 km perogee that would give 12 hour a day coverage to anyone anywhere (except the poles). The mission was to provide an APRS mobile position and status reporting link from mobiles anywhere back into the internet linked worldwide APRS infrastructure. Our satellite consisted of:

Two Digipeaters

Two completely separate but identical KPC-3+ TNCs and 2 meter radios. One was on a vertical whip and one on a horizontal whip. Power out was a whoping 10 watts each! Since these were omni antennas and we had no attitude control, 2 meters was the only option. (70 cm requires 9 more DB for both up and downlinks). Each digipeater was backed up with a failsafe timer circuit that would power-cycle the TNCs if there was no PTT activity in over a minute. Kantronics had burned us a pair of ROMS with our defaults, so we came up ready to go after each power cycle. We also added a touch tone reset circuit as an added precaution.

Power System And Structure

Each system had its own battery and 6 solar panels for nearly 100% sun illumination. One of the TNC remote control lines on each TNC was used to cross connect the power systems in case one system failed; then we could use its battery for the other one. Dr. Bill Clapp at Weber State in Ogden, Utah, was doing the power systems, and we at the Naval Academy were doing the COMM package. We would attach to a 9 foot by 3 foot square open box truss made out of 3000 lbs of steel built by Boeing as their dummy mass. This mass was a vibration mass test unit never intended for launch, but since there were no other payloads, it became the prime payload. Unfortunately for us, it was painted all WHITE, so its average termperature would be a very cold -60 deg C. We were using a thermal insulator and then thermal coatings on our boxes to achieve a nice 0 to 30 deg C operating temperature for us. This thermal design was a non-trivial exercise!

Telemetry

We used the LEDS ON/OFF switch in the TNC to switch the 5 available analog inputs to two banks of sensors, 4 currents, 4 temperatures, battery voltage and RF power out. This for each system. Thus a total of 20 telemetry values from our off-the-shelf KPC-3+s! (This satellite has no other command/control system other than the TNCs).

Link Budget

Anyone could hit the digipeaters with a 160W mobile amp and a mobile antenna (optimized for peak power up at an angle of about 40 degrees) (36,000 km is a long way away). Anyone could hear the downlink with an OSCAR-10 class station. But the intent was to just have a half dozen permanent ground stations feed the downlink into the APRServe network. Thus, eveyone everywhere could see all satellite mobiles via the existing APRS worldwide infrastructure.

Uplink

One uplink was on 144.39 since it was already authorized over the entire North American continent. And since target mobiles would be radiating mostly upward and all other terrestrial users would be radiating on their horizons, we calculated we could get a 10 to 16 dB SNR so that the mobiles would be heard above all other users. (AMSAT also gave us a possible alternate uplink in the amateur satelltie segment in case APRS caught on in Europe where 144.39 is a meteor scatter freq.)

Downlink

We could also downlink on 144.39 just fine, since with 6 ground stations all listening all the time and feeding their combined signal into APRServe, the chance that all 6 stations would have local QRM at any instant is 0.01 %, this gave us a 99.99% probability of success on every packet. AND our signal from 36,000 km would be so weak to all other APRS users, they would not even be able to hear it except with a beam antenna pointed UP... Thus we could operate fully on 144.39. BUT 144.39 as a space downlink is not politically correct in the gentlemens agreement to keep all such activity in the AMSAT segment, and would cause discontent in Europe. This is the primary reason we had completely dual redundant systems. One for North America, and one for Europe if APRS caught on over there.

But alas, the STATE DEPARTMENT just killed us! Our FREE ride was on the launch of a totally passive dummy 3,000 lb MASS being launched to test the SEA LAUNCH system. Since the launch will be in international waters, using a Russian Rocket, on a Scandinavian ship with system integration by Boeing, BOEING had to get a Technology Export License from the State Department for the Launch (This takes months if not years!). Now, since Boeing was going to add us (an active payload) to their dummy mass, STATE DEPARTEMNT said NO. They say, "this is a change to your export license which will have to be modified and resubmitted."

[For those not familiar with the Sea launch program here is the link that describes this commercial endeavor: www.boeing.com/defense-space/space/sealaunch/ -Ken Mirabella, KM6YH]

Boeing was actually getting excited about our system, since it would give them attitude and temperature information which they had no other means for getting from their totally passive mass. Integration of our NATSweb satellite was on their work plan for integration and they have had engineers working with us in preparation for the clean room evolution for the last month.

Boeing has been fighting for us now for a month trying to convince State Department that this is a trivial change and that it doesn't need this kind of bureaucratic process. But with their BILLION$$$ launch on line and only weeks away, you can see why they cannot re-do their paperwork (which has been the subject of lots of State Department oversight of late) just for our $2000 attachment. So here, Friday afternoon, we are bumped. And their 3000 lb. chunk of steel goes to orbit (for hundreds of years) with no telemetry system at all.

But we DO now have a satellite READY TO GO for the next opportunity! (anyone know of another free launch?)

OH YES, it also had a NASA modified GPS unit attached to one of the TNCs so that we could see how well GPS worked from space! ARGH!

For details see our web page:

http://web.usna.navy.mil/~bruninga/NATSat.html

The WEB page is a week or so out of date, since we were concentrating more on building it, than documenting it, but we will clean that up later...

So THANKS to all in Long Beach that had offered to help us when we flew out for integration next Wednesday, no time now to send you all personal messages, I have got to go spend some family time, since I have been working 16 hour days since XMAS and sleeping here overnight in the office about 30% of the time...

It was fun!

de WB4APR, Bob Naval Academy Satellite Lab

Followup on the EVM Radio Interface Kit

[From the TAPR DSP SIG]

Filter Files

If you recently put together the very nice TAPR EVM enclosure kit and downloaded the filter files from TAPR as collected by KC7WW, but were unable to get them to run, you may want to read this message.

I couldn't get any of the Alef Null files to run. It turned out that there were some simple instances of variable names (e.g. buflen for one) being the same in leonid.asm and the application .asm files, and this kept things from compiling.

With a few simple variable changes, I was able to get bandpass, corelflt, fft-cut, inout, and talk to work, and used them for noise reduction on some EME contacts this evening. And I think rtty works though I didn't look for a rtty station. QRMQRN howls for some reason and wasn't useable here.

If you want the modified .asm files, or the .cld files, just let me know and I'll email as attachments.

When you compile, remember to include the needed auxiliary .asm files in the same directory. If you are getting strange error messages when you compile after you correct the duplicate variable names, its probably because of not having the necessary files in the directory.

Roger Rehr, W3SZ ex-AA3QK ex-WA3JYM
rrehr@epix.net http://www.epix.net/~rrehr

Software

A couple of years ago, I uploaded a collection of stuff for the EVM called n1owu_11.zip. This package has versions of Leonid, several modems and other EVM applications, DLEXE, SPY.EXE, and one or two other goodies.

This file is at: ftp://ftp.tapr.org/dsp/recent_uploads/n1owu_11.zip. If you try to find it by simply browsing through the TAPR site, you may miss it, because it is in the "recent uploads" area, not with the other EVM software.

This package should have all the source files you need to compile the included applications without any errors. They should also run (on my hardware setup, at least). However, if your radio hookup is different than mine, you may have to change the conditional compilation lines for the CODEC channels, etc.

If you go to www.erols.com/dougbert, you will find a couple of updates to some of the files, such as bios2.asm.

When I get a chance, I ought to review the package and update it by cleaning out any obsolete or useless stuff, and making any changes to the applications that are needed to make them work without modification with the TAPR interface circuit. If anyone has any suggestions in this regard, please let me know.

Douglas Braun, N1OWU
doug.braun@intel.com

EVM Enclosure Headphones

I recently received the enclosure for the EVM. It is now up and running and I do not find any problems running any of the software in the KC7WW collection. What I did miss when running the filters is the headphones socket, so I added the headphones socket to the back panel of the enclosure. I think this is something to have if you intend to experiment with filters.

Ray, 9H1EJ

If anyone else wants to do this, be aware of one thing:

The "ground" connection of the EVM headphone jack is NOT at ground potential. It has something like 2.5 volts on it. If you stick a stereo headphone socket in your enclosure's front panel and connect it to the EVM's socket, you could create a short from the headphone pseudo-ground to the real ground. I guess the solution is to find a chassis-mount headphone socket that does NOT have an electrical connection between its common terminal and its mounting hardware. Read the CODEC hardware data sheet for more info.

My solution was to add a dedicated audio amp to one output channel and hook it to a speaker. I do not normally use the headphone output, but it can be handy for debugging software.

Douglas Braun, N1OWU
doug.braun@intel.com

Web Page Editor for APRS SIG Page Wanted

Greg Jones, WD5IVD
wd5ivd@tapr.org

TAPR is looking for someone who might be interested in taking on the job of keeping up the TAPR APRS SIG web page. With all the sites coming on-line these days with lots of neat information, we are looking for someone to keep the page current.

Our current page reflects the page I did over two years ago which is terribly in need of regular updating as the activity in APRS continues to increase.

If you are interested, please let me know.

You can view the current page at
http://www.tapr.org/tapr/html/Faprssig.html

***Connect Request

I have a TAPR TNC 2, with an HF-split mod installed. I'm looking for a tuning indicator addition for this TNC. Something driving a bar-graph LED would be nice.

Karl Fraser, KK1A
kk1a@efn.org
PO Box 41273
Eugene, OR 97404-0326

I need help on locating any group still working with the L.L.Grace DSP-12 platform.

Sydney F. Chiswell, W2ICZ
W2ICZ@buffnet.net
76 Chardon Dr.
Buffalo, NY. 14225-2314

I am from Melbourne, Australia and have a DRSI DPK-2 TNC. DRSI is no longer in business and I am trying to locate some upgrades (new or used) for the above. They include, Open Squelch DCD, High Speed Clock, and Time of Day. I would consider a TNC with mods fitted and am also seeking the DRSI 9600 modem. Thanks for your help.

Trevor Smith, VK3TI
nash072@vicnet.net.au

Personal Review of Radio Shack Simplex Repeater Controller

Ronald G. Parsons, W5RKN
w5rkn@amsat.org

Radio Shack has a new item, Simplex Repeater Controller, #19-345. In theory, this box digitally records up to 30 seconds of audio from a receiver and then automatically keys the transmitter and repeats the audio. It will run for several weeks on 4 AAA batteries and comes with a patch cable to connect to Icom, Yaesu, Alinco, etc. HTs. I thought this might be a neat gadget to fill in dead areas of a digipeater for race routes, etc.

Unfortunately, the unit has two major flaws.

1. There is a delay of up to two seconds (by specification, in practice, somewhat less) before the digital recording starts. But it is enough to cutoff the first word or so of a repeated transmission. To get a packet through, TXDelay has to be upwards of 400 msec. Short packets seldom are decodeable, even with PASSALL ON. I didn't investigate the problem with packets.

2. There is no provision for an ID, either in voice or Morse code. To be legal, the HT would have to have an auto-ID feature.

The unit does function as described in the manual, and if you don't mind the repeated transmission getting clipped at the beginning, it may work for you.

Silent Key: King Hussein, JY1

[From the ARRL Pacific Division Update and the ARRL Bulletin and Letter.]

Radio amateurs around the world joined in mourning the death of Jordan's King Hussein, JY1. Hussein, 63, died Sunday morning, Feb. 7. The Middle East's longest-reigning ruler, he'd been Jordan's king for 47 years, taking the throne when he was just a teenager. His son, Abdullah, 37, succeeds him.

Hussein had earned a reputation as a catalyst for peace and as a conciliator in the Middle East.

Hussein was a Life Member of the ARRL. ARRL Executive Vice President David Sumner, K1ZZ, called him "an enthusiastic radio amateur whose support was invaluable to us in obtaining new amateur bands at the 1979 World Administrative Radio Conference." Sumner recalled that in May 1979, International Amateur Radio Union President Noel Eaton, VE3CJ, was invited to Amman to meet with King Hussein.

"Jordan's support of the Amateur Service was much in evidence at the conference that fall, and was a crucial element in our success," Sumner said. The WARC-79 resulted in Amateur Radio's gaining the 30, 17, and 12-meter bands. That same year, JY1 was featured in the film, "The World of Amateur Radio." Hussein regarded his 1983 contact with Owen Garriott, W5LFL, on board the Space Shuttle Columbia, as a high point in his Amateur Radio activity.

King Hussein also participated in the historic 1995 joint Israel-Jordan JY74X operation on Mt. Nebo, where hams from both countries participated in a Field Day-like operation. The King put in appearances both on the air and in person, much to the delight of the participants and those waiting to work him and JY74X.

All members of the Jordanian royal family automatically have Amateur Radio privileges in Jordan. Although the new king does not appear to have a call sign, King Hussein's widow--the American-born Queen Noor--is JY1NH. King Hussein's brother, the former Crown Prince Hassan, is JY2HT, while his cousin, Prince Raad, JY2RZ, is chairman of the Royal Jordan Radio Amateur Society.

APRS Emergency Planning

Bob Bruninga
bruninga@nadn.navy.mil

I recently posted a message to the SIG explaining how APRS attempts to maintain a "net-cycle" time of 30 minutes during benign operations and a 10 minute net-cycle time during local or dynamic emergency operations so that users can have a common expectation about the latency of data on their screens. *[See the article "APRS Network Congestion" on page 9.]* Mark Sproul, writer of WinAPRS, sent me a message supporting this philosophy so now it can become a common denominator for normal APRS planning.

Thus, those of you who write emergency operations plans can include these defaults in your planning documents. WinARPS allows the user to set any rate at any time. But, since the change between 30 minutes and 10 minutes in APRSdos is automatically adjusted based on the length of the PATH, these plans should be aware of this in recommending path lengths for emergency operations.

APRSdos will switch to 10 minute rate on direct and one-hop packets. If you need more hops and still also need the 10 minute rate, you must override this by manually editing the CFIG8xx.APR file in your root directory. The default is 30 minutes (1800 seconds). But please remember to return it to the 1800 second rate for benign operations.

FCC's Hollingsworth is Net Guest

[From the ARRL Pacific Division Update and the ARRL Letter, Feb. 5, 1999.]

The FCC's point man for Amateur Radio enforcement, Riley Hollingsworth, K4ZDH, put in a guest appearance January 29 during a session of the South Carolina Single Sideband Net on 75 meters. "They contacted me and asked me if I'd take a few minutes to talk with them," said Hollingsworth, who's a South Carolina native.

Hollingsworth said he told those gathered for the net "where we've been, where we're going, and what we're trying to accomplish" in terms of ham radio enforcement. He said he expects to return to the airwaves from time to time and welcomes invitations from other organized nets. (His e-mail address is rholling@fcc.gov, phone 717-338-2502. Hollingsworth's unannounced and unexpected appearance on 3894.5 MHz January 13 to discuss compliance and enforcement created a stir within the Amateur Radio community.

PSR Deadlines

Check page two for upcoming *PSR* deadlines. If you have something for publication, please contact Bob Hansen, *PSR* editor at psr@tapr.org. TAPR is looking for technical and introductory articles on the following subjects: information on general digital communications, applications using digital communications, equipment hints or modifications, future directions and standards, tutorials, and any regional packet news or information.

Mail-in Ballot for TAPR Board of Directors Election

All Information Must be Provided:	Vote for up to three:
Name (printed):	_____ Steve Bible, N7HPR
Member Number:	_____ Steve Dimse, K4HG
Check Number (from mailing label):	_____ Bob Hansen, N2GDE
Signature:	_____ Steve Stroh, N8GNJ

Mail to: TAPR, 8987-309 E. Tanque Verde Rd. #337, Tucson, AZ 85749-9399. Must be received by March 31,1999.

Kits/Publications Update

TAPR Member's Mug, Shirts, and Badges

Don't forget that TAPR has mugs, shirts, and callsign badges available. The TAPR mug is an 11oz white Porcelain Mug which has TAPR logos in both Black and Microwavable Gold. TAPR shirts are available in a variety of sizes, styles, and colors with the TAPR logo stitched into the material. You can also have your name and callsign added underneath. And as always TAPR badges are still popular. Let everyone know you're a member and supporter of TAPR!

Shirts:
www.tapr.org/tapr/html/shirts.html
Mugs:
www.tapr.org/tapr/html/taprmug.html
Badges:
www.tapr.org/tapr/html/badge.html

TAPR GPS Update

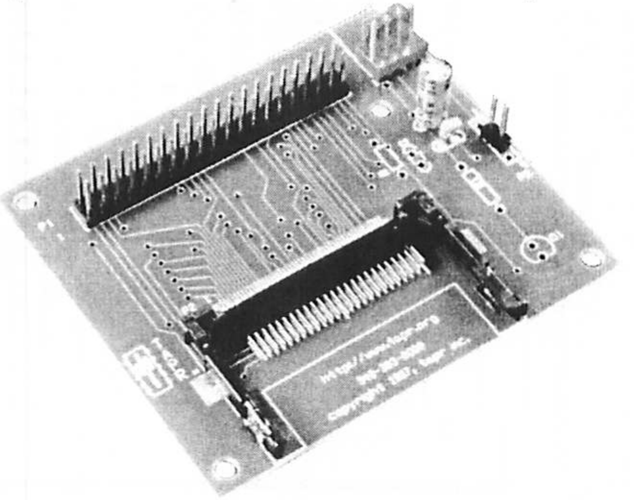
The TAPR GPS Page has been updated to include the Garmin GPS-25 OEM board, Garmin GA-27, and several new Motorola GPS units. See http://www.tapr.org/gps/ Steve Bible, N7HPR, has taken over the task of keeping up with various GPS technologies that can be made available to the membership through TAPR.

TAPR offers both the Garmin GA27 and Motorola Antenna97. This provides good choices to the membership for antennas, something that has been lacking.

On a last note regarding the GPS35TP. We are examining possibilities on how we can provide this unit and still provide something more than just getting it directly from Garmin.

CompactFlash Card Adapter

The new CFA kit is available at the office. For details check out the last issue of the *PSR* or on: http://www.tapr.org/tapr/html/cfa.html



TUC-52 and METCON-II personality board.

The METCON-2 is currently in first stage beta testing. Ron Parsons, W5RKN, has been banging away at the TUC-52/METCON-2 along with Paul Newland, AD7I. As soon as parts arrive we should be sending out the rest of the beta build.

TAC-2

The TAC-2 has been shipping at a steady rate since it became available. The TOC development seems to have stalled. As of this *PSR*, no date is defined on when we might see a TOC extension for the TAC-2. For more details on the TAC-2, just subscribe to the TACGPS mailing list. There is an excellent discussion on XNTP and other issues on the TACGPS list.

MIC-E

Another 50 kits are being completed and should be available through the office the first of February. If you have questions about what the MIC-E does or how people are using it, you can use the MIC-E mailing list.

TAPR Publications

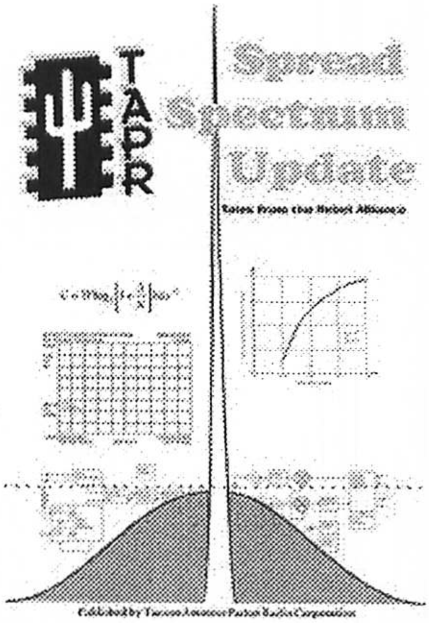
Networking Without Wires: Amateur Radio TCP/IP The first 10 chapters are completed and we are working on getting the last few chapters put in before we go to a another review of the book. We might be looking at a DCC publication date if there are not too many revisions left to do.

1999 CD-ROM

The TAPR 1999 CD-ROM should be hitting the production house sometime in March/April and be available at Dayton. We have a fix for the IE4.0 browser problem we had on the 1998 CD. With the amount of data on this upcoming CD, we might have to move to a two CD package. Keep an eye on TAPR-BB and the web page for information before the next *PSR*.

Spread Spectrum Update

The Spread Spectrum Update publication is in at the office. You can order your copy via the web page at:
www.tapr.org/tapr/html/pub.ss.html



Tucson Amateur Packet Radio

8987-309 E. Tanque Verde Rd #337
Tucson, Arizona • 85749-9399
Office: (940) 383-0000 • Fax: (940) 566-2544
Internet: TAPR@TAPR.ORG • www.tapr.org
TM Non-Profit Research and Development Corporation

January 1999

www.tapr.org • ftp.tapr.org • tapr@tapr.org
Office Hours: Tue-Fri 9am-12pm, 3pm-5pm CT

Kits

Kits	Price	Qty	Total	Kit Code	Information
CompactFlash Card Adapter	$59.00			8	A FlashCard is not included in the kit
Motorola EVM56002 Interface	$150.00			16	http://www.tapr.org/tapr/html/dsp56002evm.html
TAPR APRS™ MIC-Encoder™	$149.00			28	
TAC-2 (Totally Accurate Clock)	$139.00			10	Requires a GPS to operate
Differential GPS (Member Price)	$179.00			16	Requires a GPS to operate
DAS (DTMF Accessory Squelch)	$68.00			3	As seen in Dec 95 QST
AN-93 HF Modem .	$90.00			4	limited kits available.
TAPR 9600 bps Modem	$80.00			6	
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	

Publications

TAPR Spread Spectrum Update	**$18.00**			8	**200+ pages on ARS Spread Spectrum Topics**
1998 TAPR CD-ROM	$20.00			4	ISO 9660, 650 Megs of info! w/ html nav pages
1997 TAPR CD-ROM	$10.00			4	ISO 9660, 650 Megs of info! w/ html nav pages
Wireless Digital Communications	$39.99			8	300+ pages w/disk by: Tom McDermott, N5EG
Packet Radio: What? Why? How?	$12.00			5	130 pages. TAPR's Packet Radio book.
BBS Sysop Guide	$9.00			4	60 pages. by: Barry Buelow, WA0RJT
NOSIntro, Intro to KA9Q NOS	$23.00			8	Ian Wade, G3NRW, TCP/IP over Packet Radio
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			8	
PSR Set Vol 2 (#18 - #36 '86 - '89)	$20.00			8	
PSR Set Vol 3 (#37 - #52 '90 - '93)	$20.00			8	
PSR Set Vol 4 (#53 - #68 '94 - '97)	$35.00			16	
ARRL/TAPR 1998 DCC	**$15.00**			4	**1998 DCC Proceedings Chicago, IL**
ARRL/TAPR1997 DCC	$15.00			4	1997 DCC Proceedings Baltimore, MD
ARRL CNC Proceedings 1st - 16th	call			.	Individual Proceedings, call for prices
Entire Set ARRL DCC 1st - 16th	$120.00			144	12 Proceedings from 1981 to 1997

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
3 1/2" Disk from TAPR Library	$3.00			0	$3 per disk. See TAPR Software Library List

GPS

TAC-32 Software Registration	$55.00			0	
Garmin GPS-25 (Member Price)	$179.00			28	No Discount. (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 VP GPS (Member Price)	$269.00			28	No Discount.
Oncore VP Interface/Power Kit	$40.00			8	
Oncore GT+ GPS (Member Price)	$169.00			28	No Discount.
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.

Membership

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

○ Renewal ○ New Member

☐ **SubTotal**

– **Membership 10% Discount**

Except were noted **Member #: ______ (Place new if joining)**

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

______ **Texas Residents (7.75% tax)**

______ **Membership (New or Renewal)**

______ **Shipping and Handling**

For Total Kit Codes Between

1 - 3	4 - 7	8 - 15	16 - 27	28 - 55
Add $3	Add $5	Add $6	Add $7	Add $8

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: