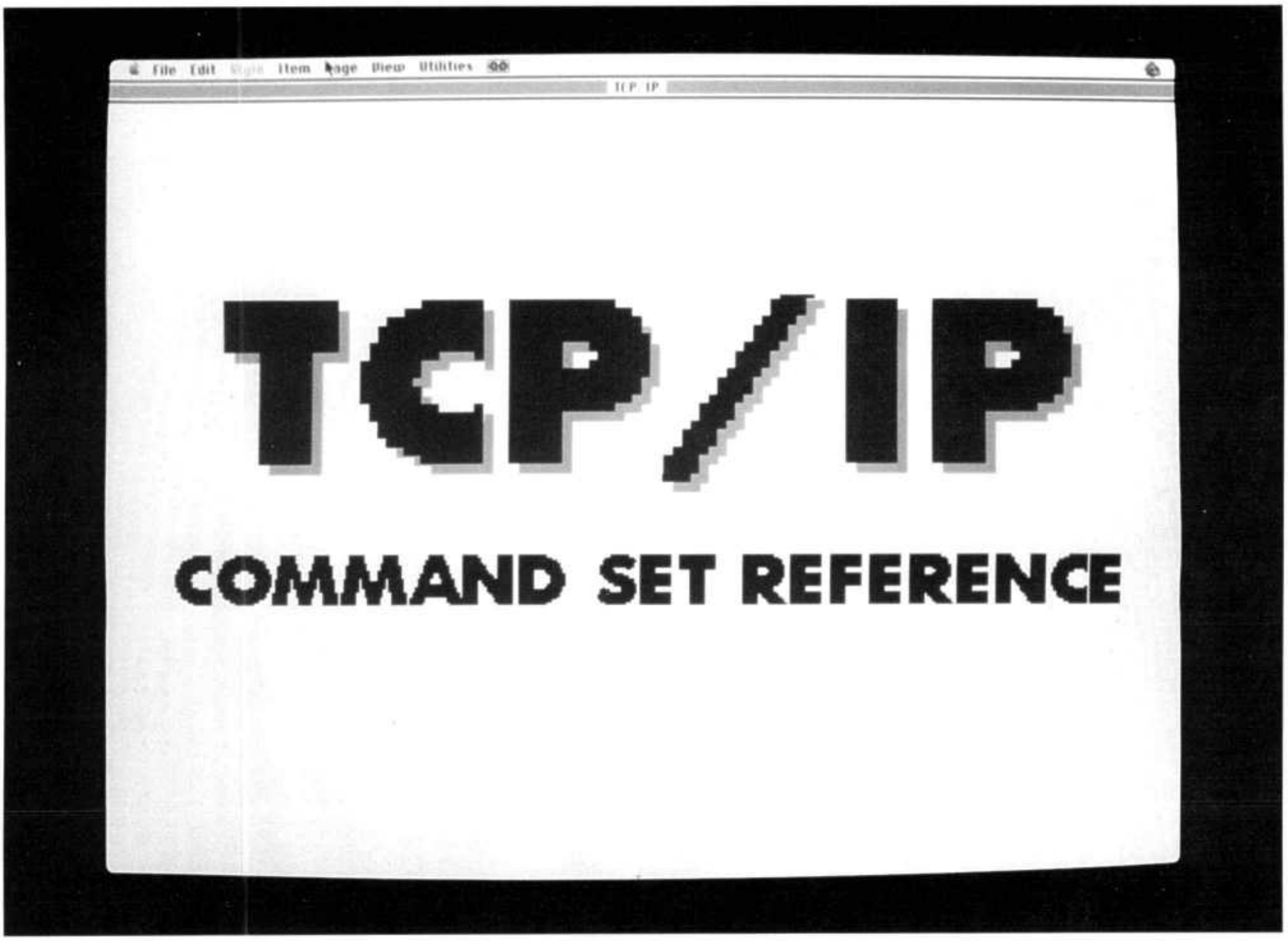
$1.75

QEX 109 WITH Gateway

ARRL Experimenters' Exchange

MARCH 1991



EX: The ARRL
:perimenters' Exchange
nerican Radio Relay League
!5 Main Street
ewington, CT USA 06111

Non-Profit Org.
US Postage
PAID
Hartford, CT
Permit No. 2929

QEX (ISSN: 0886-8093) is published monthly by the American Radio Relay League, Newington, CT USA.

Offices

225 Main St, Newington, CT 06111 USA
Telephone: 203-666-1541
Telex: 650215-5052 MCI
FAX: 203-665-7531 (24 hour direct line)
Electronic Mail: MCI MAIL ID:215-5052
(user name ARRL)
Telemail address: ARRL

Subscription rate for 12 issues:
In the US by Third Class Mail:
ARRL Member $12, nonmember $24;

US, Canada and Mexico by First Class Mail:
ARRL Member $25, nonmember $37;

Elsewhere by Airmail:
ARRL Member $48, nonmember $60.

QEX subscription orders, changes of address, and reports of missing or damaged copies may be marked: *QEX* Circulation.

Members are asked to include their membership control number or a label from their *QST* wrapper when applying.

TABLE OF CONTENTS

TCP/IP COMMAND SET REFERENCE — 3
By Ian Wade, G3NRW

A quick-reference list of commands for TCP/IP NOS packages KA9Q/PAØGRI and KA9Q/G1EMM.

DIRECT DIGITAL SYNTHESIS — 7
By Dr P. H. Saul, G8EUX

DDS, the latest feature offered in top-of-the-line rigs. G8EUX explains this new technique and how it is used in amateur radio.

POWER SUPPLY FOR GaAS FET AMPLIFIER — 10
By Zack Lau, KH6CP

This circuit provides an active supply to compensate for device and temperature variations and fits in well with the no-tune concept of microwave circuitry.

COLUMNS

CORRESPONDENCE — 12

Comments on the Safari-4 QRP transceiver; more on the "Fourth Method" and; a call for more experiments combining microwave and CDMA technology.

COMPONENTS — 14
By Mark Forbes, KC9C

This month's feature: Digital memory—volatile and nonvolatile. Also, information on 900-MHz power amplifiers, high-capacity NiCd batteries, flush-mounted nuts and 20-bit audio DAC.

GATEWAY — 16
By Stan Horzepa, WA1LOU

FCC cites PBBS business communications; San Francisco to be site of 10th ARRL Computer Networking Conference; input sought for new 222-225 MHz band; space station packet radio operational; RUDAK-2 launched; LO-19 PBBS operational soon; new packet-radio software available; home is where your TNC is. . ., and; CAPRA information packet available.

MARCH 1991 QEX ADVERTISING INDEX

American Radio Relay League: 9, 15
Communications Specialists Inc: 15
Down East Microwave: 19
Henry Radio Stores: 20
Jacob Handwerker, W1FM: 15
Kantronics: Cov III
L. L. Grace: Cov II
P. C. Electronics: 13
Rai Enterprises: 19
Sinclabs Inc: 15
Yaesu USA Inc: Cov IV

THE AMERICAN RADIO RELAY LEAGUE, INC

The American Radio Relay League, Inc, is a noncommercial association of radio amateurs, organized for the promotion of interest in Amateur Radio communication and experimentation, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of the radio art and of the public welfare, for the representation of the radio amateur in legislative matters, and for the maintenance of fraternalism and a high standard of conduct.

ARRL is an incorporated association without capital stock chartered under the laws of the State of Connecticut, and is an exempt organization under Section 501(c)(3) of the Internal Revenue Code of 1986. Its affairs are governed by a Board of Directors, whose voting members are elected every two years by the general membership. The officers are elected or appointed by the Directors. The League is noncommercial, and no one who could gain financially from the shaping of its affairs is eligible for membership on its Board.

"Of, by, and for the radio amateur," ARRL numbers within its ranks the vast majority of active amateurs in the nation and has a proud history of achievement as the standard-bearer in amateur affairs.

A bona fide interest in Amateur Radio is the only essential qualification of membership; an Amateur Radio license is not a prerequisite, although full voting membership is granted only to licensed amateurs in the US.

Membership inquiries and general correspondence should be addressed to the administrative headquarters at 225 Main Street, Newington, CT 06111 USA.

Telephone: 203-666-1541 Telex: 650215-5052 MCI.
MCI MAIL (electronic mail system) ID: 215-5052
FAX: 203-665-7531 (24-hour direct line)

Canadian membership inquiries and correspondence should be directed to CRRL Headquarters, Box 7009, Station E, London, ON N5Y 4J9, tel 519-660-1200.

Officers

President: LARRY E. PRICE, W4RA
PO Box 2067, Statesboro, GA 30458

Executive Vice President: DAVID SUMNER, K1ZZ

Purposes of *QEX*:

1) provide a medium for the exchange of ideas and information between Amateur Radio experimenters
2) document advanced technical work in the Amateur Radio field
3) support efforts to advance the state of the Amateur Radio art.

All correspondence concerning *QEX* should be addressed to the American Radio Relay League, 225 Main Street, Newington, CT 06111 USA. Envelopes containing manuscripts and correspondence for publication in *QEX* should be marked: Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be typed and double spaced. Please use the standard ARRL abbreviations found in recent editions of *The ARRL Handbook.* Photos should be glossy, black-and-white positive prints of good definition and contrast, and should be the same size or larger than the size that is to appear in *QEX.*

Empirically Speaking . . .

HF Packet Radio R&D Project Update

The news is that the Federal Emergency Management Agency has deleted the requirement in their grant to ARRL that the government would own any intellectual property (IP) resulting from the grant. Thus, prospective participants who were discouraged by the IP requirement should reconsider joining in the program. Just to be clear about it, there are no longer any strings attached which would give the government any ownership of any designs, inventions or copyrights resulting from work done under this program.

The dropping of the IP strings was raised with FEMA in the fall of 1990. By the end of the year, it became clear to everyone involved that the IP problem had prevented the very thing the grant was meant to do—stimulate HF amateur packet-radio development. In the discussions, FEMA asked "whatever happened to the idea that hams were willing to share their designs with other hams?" Our answer was that this is still true, and Amateur Radio magazines are testimony to that. What has changed, however, is that Amateur Radio systems are much more complex than they were decades ago and may take thousands of hours of individual or team labor to complete. With such an investment, many designers feel that they should have a "piece of the action," or at least IP rights control, if their design goes into commercial production.

Another point brought up by several prospective participants was that their pre-employment agreement stipulates that their current employer owns any IP developed by them on duty or off. So, they were unable to get involved in anything that said that the government would own the rights. Thus, that's resolved now, as the IP problem is left strictly to the individual or between the individual and his or her employer. Neither FEMA nor ARRL will be involved in any IP considerations. If you want to sell or license your design to a commercial manufacturer, that's no longer any concern of FEMA—in fact they have come to realize that this would be a good outcome for them as they could just buy completed units off the shelf.

So, what next? ARRL is ready to receive written proposals for funding assistance to stimulate development of HF amateur packet radio. There is $9450 of spending authority still left in the FEMA grant. The money can be spent to reimburse out-of-pocket expenses but not labor. For example, if a prospective participant proposed to write multitone modem software for a DSP board, this money could be used to pay for a pair of DSP boards to prove the software. Another example might be that money could be used to pay for hardware used to develop software for some type of majority voting scheme for packet decoding.

If you might be interested in participating in this program, please first read "The Great 1989 HF Packet Design Quest" in the May 1989 issue of *QST*, and "Empirically Speaking" in the January 1990 issue of *QEX*. Then contact me or *QEX* Assistant Editor Lori (Maty) Weinberg at ARRL HQ for additional information. We will be sending a letter to those who have indicated an interest.

New Amateur Contributions to CCIR

Those who have been following the ARRL and IARU preparations for WARC-92 are aware that the International Radio Consultative Committee (CCIR) provides each WARC with a report which provides the technical basis for the Conference. But there is another aspect of CCIR's work—their routine business, that of establishing international standards for radio services and systems. The CCIR conducts this standards development by means of Study Groups (arranged by radio service) and does so in four-year study periods. The study period just concluded was 1986-1990, and the documents containing all the Questions, Recommendations and Reports for that period should be available in the near future. The amateur service and amateur-satellite service contributed to that effort, and the approved papers will appear in the volume for Study Group 8.

1990-1994 is the next study period, and work is now underway. Digital Committee member Paul Newland, AD7I, and Paul Rinaldo, W4RI, drafted three documents that, if approved, would modify Recommendation 625 (the specs for TOR), Report 540 (the basis for NAVTEX) and Report 1027 (a description of a TOR adaptive ARQ technique developed by the USSR). On February 7, these documents were introduced to the US working group responsible for contributions relating to the maritime mobile service. Copies were also circulated to US manufacturers of AMTOR equipment. The maritime mobile working group will meet again on April 15, May 13 and June 17. Please let Paul Newland or me know if you would like to review these draft contributions.

In addition, we are planning to introduce one or more draft contributions concerning the amateur and amateur-satellite services into the US working group that deals with the amateur services. That group meets next on March 25 and will likely meet several times in the following months in preparation for the Study Group 8 meeting in Geneva December 11-20, 1991.—*W4RI*

TCP/IP Command Set Reference
Version 1.6: 23 January 1991

By Ian Wade, G3NRW
7 Daubeney Close
Harlington Dunstable
Bedfordshire LU5 6NF
United Kingdom

I originally started using amateur TCP/IP about 18 months ago, but found that the "net" package wasn't particularly exciting compared with the commercial versions I had been using elsewhere, so I didn't pursue it.

Then, a few months ago, I decided to take another look at amateur TCP/IP and discovered that Phil Karn and company had obviously spent some time sprucing it up.

I decided I needed a quick-reference list of commands for the NOS package; hence this document. I have found it very useful in trying out new commands.

Abstract

This document contains details of all of the commands to be found in the following TCP/IP Network Operating System (NOS) packages:

KA9Q/PAØGRI: version 900819
KA9Q/G1EMM: version KH113014

Familiarity with the basic functions of the NOS packages is assumed. This document is intended only as a quick reference to the TCP/IP command set, and does not describe how individual commands work.

Rationalization of Parameter Descriptions

Because the NOS packages contain software modules originating from several different sources, the documentation which describes them inevitably contains a number of inconsistencies. For example, the words "label" and "interface" apparently describe different objects, whereas in actuality they are the same thing. On the other hand, the meaning of the word "address" used in many commands differs from command to command.

In this document an attempt has been made to rationalize the meaning of these parameters to produce a consistent command set within and across the two NOS packages.

The principal parameters which have caused confusion in the past are to do with names, addresses and interfaces. These are now defined as follows:

<callsign>	an AX.25 MYCALL callsign (e.g. "G3NRW-1")
<hostname>	a computer name (eg "g3nrw" or "g3nrw.ampr.org")
<ipaddress>	an Internet address (eg "44.131.5.2")
<host>	<hostname> or <ipaddress>
<username>	a user at a computer (eg "ian")
<interface>	a device interface name (eg "pk0" or "ax0")
<ioaddress>	a device I/O base address (eg "0x3f8")
<vector>	an IRQ level (eg "4")

(Note that the word "hostid" is not used in this document to avoid confusion with the Unix command of the same name.)

TCP/IP NOS Command Set

The rest of this document specifies the complete NOS command set for the KA9Q/PAØGRI and KA9Q/G1EMM packages.

The extreme left-hand column of each line is coded as follows:

i	signifies a command in the KA9Q/PAØGRI version of NOS
m	signifies a command in the KA9Q/G1EMM version of NOS

The extreme right-hand column shows the default values of command parameters.

TCP/IP "NOS" COMMAND SET REFERENCE

The rest of this document specifies the complete "NOS" command set for the KA9Q/PA0GRI and KA9Q/G1EMM packages.

The extreme left-hand column of each line is coded as follows:

```
i |     signifies a command in the KA9Q/PA0GRI version of "NOS"
 m|     signifies a command in the KA9Q/G1EMM version of "NOS"
```

The extreme right-hand column shows the default values of command parameters.

```
Version|                                                        |Default

im| ?        (help: list of top-level "nos>" commands)
im| !        (break out to shell)
im| #        (comment line)

im| abort                [<session #>]                 (FTP)

im| arp

i | arp add              <host> ether|ax25|netrom
                                 <ether_addr>|<callsign>

 m| arp add              <host> ether|ax25|netrom|arcnet
                                 <ether_addr>|<callsign>

i | arp drop             <host> ether|ax25|netrom

 m| arp drop             <host> ether|ax25|netrom|arcnet

im| arp flush

i | arp publish          <host> ether|ax25|netrom
                                 <ether_addr>|<callsign>

 m| arp publish          <host> ether|ax25|netrom|arcnet
                                 <ether_addr>|<callsign>

im| asystat

i | attach 3c500         <ioaddress> <vector> arpa <interface>
                              <buffers> <mtu> [ipaddress]

i | attach asy           <ioaddress> <vector> slip|ax25|nrs
                              <interface> <buffers> <mtu>
                              <speed> [ipaddress]

 m| attach asy           <ioaddress> <vector> slip|ax25|nrs|ppp
                              <interface> <buffers> <mtu>
                              <speed> [options]

 m| attach drsi          <ioaddress> <vector> ax25 <interface>
                              <bufsize> <mtu>
                              <chan_a_speed> <chan_b_speed>
                              [<ipaddress_a>] [<ipaddress_b>]

 m| attach eagle         <ioaddress> <vector> ax25 <interface>
                              <buffers> <mtu> <speed>
                              [<ipaddress_a>] [<ipaddress_b>]

 m| attach hapn          <ioaddress> <vector> ax25 <interface>
                              <rx_bufsize> <mtu> csma|full
                              [<ipaddress>]

im| attach hs            <ioaddress> <vector> ax25 <interface>
                              <buffers> <mtu> <txdelay>
                              <persistence> [<ipaddress_a>]
                              [<ipaddress_b>]

 m| attach kiss          <asy_interface> <port> <interface>
                              [<mtu>]

im| attach netrom

im| attach packet        <vector> <interface> <buffers>
                              <mtu> [<ipaddress>]

 m| attach pc100         <ioaddress> <vector> ax25 <interface>
                              <buffers> <mtu> <speed>
                              [<ipaddress_a>] [<ipaddress_b>]

im| attach scc           <devices> init <ioaddress> <spacing>
                              <Aoff> <Boff> <Dataoff>
                              <intack> <vector> [p]<clock>
                              [hdwe] [<param>]

im| attach scc           <chan> slip|kiss|nrs|ax25
                              <interface> <mtu> <speed>
                              <bufsize> [<callsign>]

 m| attended             [off|on]                              |on

 m| ax25 bc              <interface>
 m| ax25 bcinterval      [<seconds>]                           |0
 m| ax25 bctext          ["<broadcast text>"]
im| ax25 blimit          [<val>]                               |30
im| ax25 digipeat        [on|off]                              |on
im| ax25 flush
im| ax25 heard
im| ax25 irtt            [<millisecs>]                         |5000
im| ax25 kick            <&AXB>
im| ax25 maxframe        [<window_size>]                       |1
im| ax25 mycall          [<callsign>]
im| ax25 paclen          [<bytes>]                             |256
im| ax25 pthresh         [<bytes>]                             |128
im| ax25 reset           <&AXB>
im| ax25 retry           [<n>]                                 |10
```

```
im| ax25 route
im| ax25 route add    <target_callsign> [<digi_callsign> ...]
im| ax25 route drop   <target_callsign>
 m| ax25 route mode   <target_callsign> [vc|datagm|interface]

im| ax25 status       [<&AXB>]
im| ax25 t3           [<millisecs>]                            |0
 m| ax25 t4           [<seconds>]                              |300
 m| ax25 timertype    [original|linear|exponential]            |exp
i | ax25 version      [1|2]                                    |1
 m| ax25 version      [1|2]                                    |2
im| ax25 window       [<bytes>]                                |2048

im| bbs

im|        Help         ? (command list)
im|        Area         A
im|        Bye          B
im|        Chat         C
im|        Download     D <filename>
im|        Escape       E [<esc_char>]                         ^X
im|        Finger       F [<username>]
im|        Gateway      G <interface> <callsign> [<callsign>...]
im|        Help         H                     (full helpfile)
 m|        Information  I
im|        Heard        J
im|        Kill         K <n> ...
im|        List         L [<n> ...]
im|        Netrom       N
im|        Read         R [<n> ...]
im|        Send         S <username> [@ <host>] [< <from_addr>]
im|                                       [$<bulletin_id>]
im|        Telnet       T [<host>]
im|        Upload       U <filename>
 m|        Verbose      V
im|        What         W [<directory>]
im|        Zap          Z <filename>

im|        Remote       @
im|        Expert       [<string>]

i |        (unknown)    >
im|        (unknown)    "
im|        (unknown)    #

im| cd            [<directory>]

im| close         [<session #>]

 m| comm          <interface> <string>

im| connect       <interface> <callsign> [<digi_callsign> ...]

im| delete        <filename>

im| detach        <interface>

 m| dialer        <interface> <interval> <ping_target>
                       <script_filename>

im| dir           [<directory> | <file>]

im| disconnect    [<session #>]                        (AX.25)

im| domain addserver  <host> [<host> ...]

i | domain addserver  <own_host>                 (starts server)

i | domain bootp      <domainserver_ipaddress> [filename]

 m| domain cache clean [off|on]                                |off
 m| domain cache list
 m| domain cache size  [<entries>]                             |20
 m| domain cache wait  [<seconds>]

i | domain debug      [off|on]                                 |off
im| domain dropserver <host> [<host> ...]
 m| domain list
i | domain listservers
i | domain load       [<filename>]
i | domain loopback   [off|on]                                 |off
 m| domain maxwait    [<seconds>]                              |60
i | domain nslookup   <domainserver_host> <record_type> <name>
i | domain retries    [<n>]                                    |0
 m| domain retry      [<n>]
i | domain save       <filename>
im| domain suffix     [<domain_suffix>]
i | domain timeout    [<seconds>]                              |30
im| domain trace      [off|on]                                 |off

 m| domain translate  [off|on]                                 |off
 m| domain verbose    [off|on]                                 |on

 m| drsistat

 m| dump        <hex_memoryaddress> | <.> [<decimal_range>]

 m| eaglestat

im| echo        [accept|refuse]                      (telnet)  |accept
```

continued on next page

```
i  | eol         [unix|standard]                     (telnet) | standd
m  | eol         [standard|null]                     (telnet) | standd

m  | escape      <hex_character>          (always F10 on PC) | 0x1d

i  | etherstat

im | exit
```

```
im | finger      [<username>]@<host>
   |                    (no spaces between parameters)

i  | fkey
i  | fkey        <key_number> [<value> | "<string>" ]
   |                    (terminate string with ^M for CR)

   | f1   59    sf1   84    cf1   94    af1   104    pgup  73
   | f2   60    sf2   85    cf2   95    af2   105    pgdn  81
   | f3   61    sf3   86    cf3   96    af3   106    home  71
   | f4   62    sf4   87    cf4   97    af4   107    end   79
   | f5   63    sf5   88    cf5   98    af5   108    arup  72
   | f6   64    sf6   89    cf6   99    af6   109    ardn  80
   | f7   65    sf7   90    cf7   100   af7   110    ar l  75
   | f8   66    sf8   91    cf8   101   af8   111    ar r  77
   | f9   67    sf9   92    cf9   102   af9   112    ins   82
   | f10  68    sf10  93    cf10  103   af10  113    del   83

im | ftp <host>

   |      ascii
   |      batch   [off|on]
   |      binary
   |      cd      [<dir>]
   |      dele    <remote_file>
   |      dir     [<filename> | <dir> [<local_filename>] ]
   |      get     <remote file> [<local file>]
   |      h
   |      list    [<filename> | <dir> [<local_filename>] ]
   |      ls      [<filename> | <dir> [<local_filename>] ]
   |      mget    <remote_file> [<remote_file> ...]
   |      mkdir   <remote_dir>
   |      mput    <local_file> [<local_file> ...]
   |      nlst    [<filename> | <dir> [<local_filename>] ]
   |      pass    <password>
   |      put     <local_file> [<remote_file>]
   |      pwd     [<dir>]
   |      quit
   |      rmdir   <remote_dir>
   |      type    [a | i | l <bytesize> ]                 | a
   |      user    <username>

im | F10                (to escape to "nos" command level)
```

```
m  | hapnstat
im | help               (list of top-level "nos" commands)

im | hopcheck check     <host>
im | hopcheck maxttl    [<hops>]                          | 30
im | hopcheck maxwait   [<seconds>]                       | 5
im | hopcheck queries   [<count>]                         | 3
im | hopcheck trace     [off|on]                          | off

im | hostname     [<hostname>]

im | hs
```

```
im | icmp echo    [off|on]   (must be "on" for one-shot pings) | on
im | icmp status
im | icmp trace   [off|on]                                  | off

im | ifconfig  [<interface>]
im | ifconfig  <interface> broadcast      <bcast_ipaddress>
im | ifconfig  <interface> encapsulation  none|ax25|slip|netrom
im | ifconfig  <interface> forward        <ipaddress>
im | ifconfig  <interface> ipaddress      <ipaddress>
im | ifconfig  <interface> linkaddress    <callsign>
im | ifconfig  <interface> mtu            <bytes>
im | ifconfig  <interface> netmask        [0x]<hexmask>
im | ifconfig  <interface> rxbuf

i  | info

im | ip address   [<ipaddress> | <hostname>]
im | ip rtimer    [<seconds>]                               | 30
im | ip status
im | ip ttl       [<hops>]                                  | 255

m  | isat         [off|on]                                  | off
```

```
im | kick         [<session #>]
```

```
im | log          [<log_filename> | stop]
```

```
m  | mail

im | mbox
m  | mbox attend       [off|on]
m  | mbox kick
m  | mbox maxmsg       [<n>]                                | 200
m  | mbox motd         ["<string>"]
m  | mbox status
m  | mbox timer        [<seconds>]
m  | mbox tiptimeout   [<seconds>]
```

```
im | mem free
im | mem garbage
m  | mem ifbufsize   [<bytes>]                              | 2048
m  | mem nibufs      [<n>]                                  | 5
im | mem sizes
im | mem status
im | mem thresh      [<bytes>]                              | 8192

im | mkdir       <directory>

im | mode        <interface> [vc|datagram]          (AX.25)

im | more        <filename> [<filename> ...]      (q: quit)
   |                                     (space: next page)
   |                                      (CR: next line)

m  | motd        ["<string>"]

m  | multitask [off|on]                                     | on
```

```
im | netrom acktime    [<millisecs>]                        | 3000
im | netrom bcnodes    <interface>
im | netrom connect    <node_callsign> | <node alias>
im | netrom choketime  [<millisecs>]                        | 180000
m  | netrom derate     [off|on]                             | on

im | netrom interface  <interface> <alias> <quality>
im | netrom irtt       [<millisecs>]                        | 15000
im | netrom kick       <&CB>
m  | netrom minquality [<n>]                                | 10

im | netrom nodefilter
im | netrom nodefilter add  <neighbour_callsign> <interface>
im | netrom nodefilter drop <neighbour_callsign> <interface>
im | netrom nodefilter mode [none | accept | reject]

im | netrom nodetimer  [<seconds>]                          | 0
im | netrom obsotimer  [<seconds>]                          | 0
m  | netrom promiscuous [off | on]                          | off
im | netrom qlimit     [<bytes>]                            | 2048
im | netrom reset      <&CB>
im | netrom retries    [<n>]                                | 10

im | netrom route
im | netrom route add  <alias> <destination> <interface>
   |                            <quality> <neighbour>
im | netrom route drop <destination> <neighbour> <interface>
im | netrom route info <destination>

im | netrom status
m  | netrom timertype  [linear | exponential]               | exp
i  | netrom ttl        [<hops>]                             | 64
m  | netrom ttl        [<hops>]                             | 10
im | netrom user       [<username>]
im | netrom verbose    [off | on]                           | off
im | netrom window     [<frames>]                           | 4

im | nntp addserver    <nntpserver_host> <interval_in_seconds>
   |                   [<time_range>] [<group> [<group> ...]]

im | nntp directory    [spool | control <directory>]
im | nntp dropserver   <nntpserver>
im | nntp groups       [<newsgroup name> ... ]
im | nntp kick         <nntpserver>
im | nntp listservers
im | nntp trace        [<trace #>]     (0: no trace)          | 1
   |                                   (1: serious errors)
   |                                   (2: transient errors)
   |                                   (3: session progress)
   |                                   (4: received articles)
   |                                   (5: errors)

im | nrstat
```

```
im | param <interface> <param> [<param> ...]
```

```
im | param <KISS_interface>   0 <data frame>
im | param <KISS_interface>   1 <TX_delay>       (10 mS units)
im | param <KISS_interface>   2 <persistence>         (0-255)
im | param <KISS_interface>   3 <slot_time>      (10 mS units)
im | param <KISS_interface>   4 <TX_tail>        (10 mS units)
im | param <KISS_interface>   5 <n>         (n=0: half duplex)
   |                                       (n>0: full duplex)

im | param <KISS_interface> 255                (exit KISS mode)

im | ping        <host> [<len> [<seconds> [<incflag>]]]

im | pop mailbox       <mbox>
im | pop mailhost      [<host>]
im | pop kick
m  | pop quiet         [off|on]                             | off
im | pop timer         [<seconds>]                          | 0
im | pop userdata      [<username> <password>]

im | ps

im | pwd               [<directory>]
```

```
im | record            [<filename> | off]                   | off

im | remote            [-p port] [-k key] [-a kickaddr]
   |                        <host> exit|reset|kick

im | remote            -s <key>

im | rename            <old_filename> <new filename>

im | reset             [<session #>]

im | rip accept        <incoming_gateway_host>
```

continued on next page

```
im| rip add           <destination_host> <secs> [<flags>]
                                   (1: include route to self)
                                   (2: split horizon)
                                   (4: triggered update)

im| rip drop          <destination_host>
im| rip merge         [off|on]                              | off
im| rip refuse        <incoming_gateway_host>
im| rip request       <incoming_gateway_host>
im| rip status
im| rip trace         [<n>]              (0: no trace)       | 0
                                         (1: changes only)
                                         (2: full trace)

i | rlogin            <host>

im| rmdir             <directory>

im| route
im| route add         <dest_host>[/<bits>]|default
                        <interface> [gateway_host [<metric>]]

im| route addprivate  <dest_host>[/<bits>]
                        <interface> [gateway_host [<metric>]]

im| route drop        <dest_host>[/<bits>]
im| route flush
im| route lookup      <dest_host>

m | rspf interface    [<name> <quality> <horizon>]
      (needs ifconfig <interface> broadcast 44.255.255.255)

m | rspf message      [<"message_string">]
m | rspf maxping      [<n>]                                 | 5
m | rspf mode         [vc | datagram | none]                | none
m | rspf rrhtimer     [<seconds>]                           | 0
m | rspf routes
m | rspf status
m | rspf suspecttimer [<seconds>]
m | rspf timer        [<seconds>]                           | 0

im| sccstat

im| session       [<session #>]

im| shell

m | smtp batch        [off|on]                              | off
im| smtp gateway      [<host>]
im| smtp mode         [ queue | route ]                     | route
im| smtp kick
im| smtp kill         <jobnumber>
im| smtp list
im| smtp maxclients   [<n>]                                 | 10
i | smtp mxlookup     [off|on]                              | off
m | smtp quiet        [off|on]                              | off
im| smtp timer        [<seconds>]                           | 0
im| smtp trace        [<n>]              (0: trace off)      | 0
                                         (1: trace on)

im| socket        [<socket #>]

m | source        <script_filename>

im| start         ax25|discard|echo|finger|ftp|netrom|pop
                        remote|rip|smtp|telnet|ttylink

m | start tip     <interface>

m | status

im| stop          ax25|discard|echo|finger|ftp|netrom|pop
                        remote|rip|smtp|telnet|ttylink
m | stop tip      <interface>

m | tail          <filename>

im| tcp irtt      [<millisecs>]                             | 5000
im| tcp kick      <&TCB>
im| tcp mss       [<bytes>]                                 | 512
im| tcp reset     <&TCB>
im| tcp rtt       <&TCB> <millisecs>
im| tcp status    [<&TCB>]
m | tcp syndata   [off|on]                                  | off
m | tcp timertype [linear | exponential]                    | exp
im| tcp trace     [off | on]                                | off
im| tcp window    [<bytes>]                                 | 2048
```

```
im| telnet        <host> [<well_known_port_number>]         | 23
m | third-party   [off | on]                                | on
im| tip           <interface>
i | ttylink       <host> [<well_known_port_number>]         | 87
im| trace     <interface> [<BTIO_flags> [<trace_filename>]]]

    BTIO FLAGS:

        B=0   Broadcast filter off (trace all packets)
        B=1   Broadcast filter on (ignore broadcasts)

        T=0   Display protocol headers only
        T=1   Display headers + ASCII text
        T=2   Display headers + ASCII text + hex

        I=0   Ignore input packets
        I=1   Trace input packets

        O=0   Ignore output packets
        O=1   Trace output packets

im| udp       status

im| upload    <filename>

im| watch     [off|on]                                      | on
m | watchdog  [off|on]                                      | off
```

```
NOS STARTUP OPTIONS
===================

  nos [-t]                          (trace startup)
      [-d <root_directory>]
      [-s <socket_array_size>]
      [-m <heap_memory_in_KB>]
      [ <nos_autoexec_filename> ]

/ftpusers PERMISSIONS
=====================

    1    read file              )
    2    create new file        )  "ftp" and "telnet"
    4    write/delete file      )

    8    AX.25 Gateway access   )
   16    Telnet Gateway access  )
   32    NET/ROM Access         )  "telnet" only
   64    Remote control         )
  128    Disallow access        )

WELL-KNOWN PORT NUMBERS (decimal)
=================================

    0  reserved       23  TELNET            79  FINGER
  1-4  unassigned     25  SMTP              87  TTYLINK
    5  RJE            37  TIME              95  SUPDUP
    7  ECHO           39  RLP              101  HOSTNAME
    9  DISCARD        42  NAMESERVER       102  ISO-TSAP
   11  USERS          43  NICNAME          109  POP-2
   13  DAYTIME        53  DOMAIN           113  AUTH
   15  NETSTAT        67  BOOTPS           117  UUCP-PATH
   17  QUOTE          68  BOOTPC           119  NNTP
   19  CHARGEN        69  TFTP             513  RLOGIN
   20  FTP-data       75  private dialout
   21  FTP            77  private rje
```

Bits

6th Annual Satellite Colloquium

The 6th Annual Satellite Colloquium, sponsored by AMSAT-UK, will be held July 25-28, 1991. As in past years, the University of Surrey will be the meeting place for this Colloquium on OSCAR satellites and digital communications. Question-and-answer sessions with various amateur satellite experts are planned, as are tours of the UoSat command station. Cost is expected to be the same as last year, approximately US $84 per day, which includes the full day's events and accommodations. The following is the preliminary line-up of programs:

Thursday, July 25—International Satellite Day
Friday, July 26—Satellite Sessions
Saturday, July 27—Satellite Sessions
Sunday, July 28—Satellite Sessions

Those interested in attending this year's Colloquium are urged to contact Ron Broadbent, G3AAJ, as soon as possible to insure that the proper amount of space is reserved. Ron's address is: 94 Herongate Road, Wanstead Park, London E12 5EQ England. Phone: [from US] 011 (44) 81-989-6741, or fax: [from US] 011 (44) 81-989-3430.

Direct Digital Synthesis

What Is It and How can I Use It?

By Dr. P. H. Saul, G8EUX

Reprinted from the December 1990 issue of *Radio Communication*

'DIRECT DIGITAL SYNTHESIS' (DDS) has appeared several times recently in *RadCom*, both in *Technical Topics* [1] and in a review [2]. Direct Frequency Synthesis (DFS) and Numerically Controlled Oscillator (NCO) are synonymous terms. It is the latest feature offered in top-of-the-range rigs such as the Icom IC-781 and the new Yaesu FT-1000, which has, according to the advertisements, no less than five direct digital synthesisers. What then is this apparently new technique, and what is its relevance to amateur radio?

TERMINOLOGY

FIRST, SOME TERMINOLOGY explained. Direct synthesis (without the digital bit) simply means the production of an output frequency, either a transmitter output or a local oscillator for a receiver by the addition of several, sometimes many oscillators, usually by a combination of mixing, multiplying, re-mixing etc., with many stages of filtering. A truly analogue technique, and horrifically expensive to carry out really well. Examples were primarily limited to specialist, often military, synthesizers in the fifties. Outputs could be made very clean, especially in respect of close-to-carrier phase noise, but only by massive amounts of filtering, usually racks full of it! Of course, synthesizers of this type never made it into amateur production equipment.

A simpler version did go into quantity production, primarily, at least in numbers, in the early U.S. market C.B radios. This was the 'crystal bank' synthesizer, shown in **Fig 1**. This arrangement used two arrays of crystals, selected in appropriate combinations to give the required coverage and channel spacing. It worked very well over a restricted frequency range, could offer very low phase noise since crystal oscillators are inherently 'quiet', and could be made acceptably (for the time) compact, and low power. A familiar example on the UK market some 15 years ago was the very well known 'Liner 2'. This used crystal bank synthesis, with a front-panel tuned VXO on a further crystal oscillator for the conversion up and down from the effective 'tuneable IF' at 30MHz to the working frequency of 145MHz. The need for the VXO illustrates the weakness of this type of synthesizer; close channel spacing is essentially prohibited by the very large number of crystals which would be needed. Triple bank versions were described in the professional literature, but were not produced for amateur purposes. The other disadvantage of this class of synthesizer is the high cost of the crystals, but it remains an interesting technique capable of very high performance with reasonable design. The well-documented 'faults' of the Liner 2 were not connected with the synthesizer, but were due to overdrive of a later mixer stage.

Most of the earlier commercial amateur bands HF radios overcame the need for frequency stability by the use of a tuneable IF system and a carefully engineered VFO. Examples are in the KW range, the Yaesu/Sommerkamp FT-DX series and, in homebrew form, the G2DAF. This technique survived in solid state form, with among others, the FT101. However, in parallel with these came the early attempts at combining the best VFO characteristics with synthesis in order

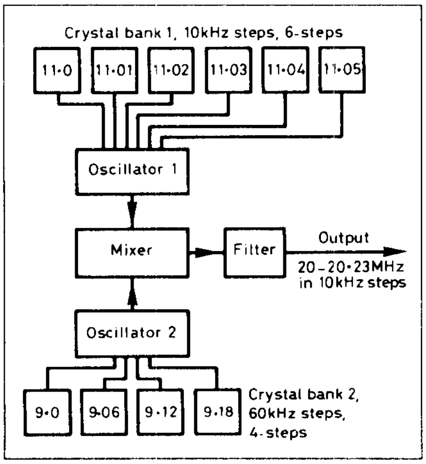


Fig 1: Simplified crystal bank synthesizer

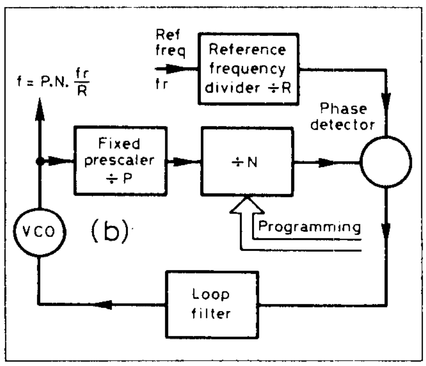


Fig 2a:Direct division

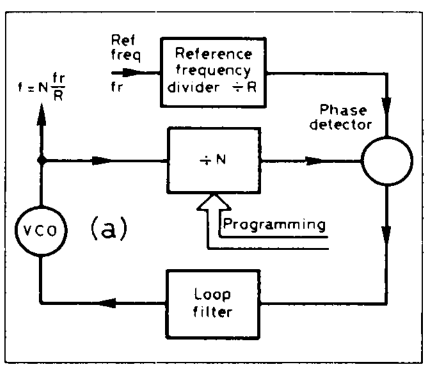


Fig 2b: Fixed prescaling

to build a single conversion receiver, with the objective of achieving exceptionally wide dynamic range. An example of this was G3PDM's classic phase-locked oscillator as used in his receiver [3], This was a true synthesizer, albeit analogue, based primarily on valves, with a VFO using a FET and two bipolar transistors which is still regarded as the standard to beat. The author was present at a demonstration of this receiver where a signal below 1 microvolt suffered no apparent degradation from a 10volt signal 50KHz away, an amazing 140+dB dynamic range made possible by the extremely low phase noise of this synthesizer, and of course the superb linearity of a beam-deflection mixer run with 100 volt push-pull local oscillator drive!

About this time (1969), the first truly digital (but not direct) synthesizers were beginning to appear, chiefly in military equipment; the outstanding example is the UK sourced 'Clansman' series of radios, still in service with UK forces.

BASIC SYNTHESIZER

A BASIC SINGLE LOOP digital synthesizer is shown in **fig 2**. Actually, this is very simplified. Fully programmable dividers of appropriate frequency range were not immediately available. Two solutions were proposed for this; the 'dual modulus' prescalers and a variant of the crystal mix scheme. The first of these uses the properties of a dual modulus, or in some cases even a triple modulus, counter to overcome the difficulty of producing a really high frequency fully programmable divider. The scheme is shown in **fig 3**. It consists of a VCO, usually in discrete component form, a dual modulus counter, a main counter, a reference oscillator and divider chain, a phase detector and a loop amplifier/filter. This type of synthesizer has been described many times, both in the professional and amateur fields. In some variants, it probably represents the current status of PLL frequency synthesis, especially for very wide frequency range applications such as TV receivers and satellite tuners.

The second method, that of crystal mixing, is probably the most widely used technique in amateur and PMR equipment, where wide operating frequency range is not needed. It offers advantages of simplicity and low component count, especially when the digital functions of the system can be contained in a single chip. The scheme is shown in **fig 4**. Unlike the dual modulus scheme, no prescaler is needed, but a high frequency mixer is necessary, usually with two crystals , one for mixing down and one for the reference.

Almost all the commercially available phase locked loop synthesizers use an external VCO, which must be designed by the equipment builder; not too difficult to a commercial concern, but no mean task for the amateur building a one-off. The problem is that the VCO is critical to achieving good performance from the synthesizer, especially in respect of phase noise , and hence reciprocal mixing in a receiver. While the lock loop can take care of phase noise close to the carrier, a badly designed circuit or a badly laid out PCB can cause a synthesizer to 'feature' sidebands on the output spectrum at the comparison frequency and sometimes harmonics of it. The VCO dominates the noise away from the nominal frequency, ie outside the PLL bandwidth, which may be only a kHz or so, less in the case of very narrow frequency increments. Phase detector design is another critical area; the most recent PLL chips [5] offer two or more phase detectors to cover the far-from-lock and locked-in cases. All in all, PLL synthesizer design is a very demanding and essentially analogue task, as anyone who has built one and carefully examined the output on a spectrum analyser will testify. Nevertheless, truly excellent performance can be achieved, by the professional engineer in factory built equipment and by the very dedicated and well equipped amateur who does not cost his time into the project. It should be said that for any synthesizer work, a spectrum analyser, or at least a continuously tuneable receiver covering the frequency of interest and a harmonic or two, is essential.

LOCK-UP TIME

PHASE LOCKED LOOP synthesizers have one further disadvantage. The lock-up time is inherently related to the smallest frequency increment. For an FM only rig, with say 5KHz minimum frequency step, this is of no real consequence. For SSB, however, even 100Hz steps are disconcertingly large to most operators, and 10 Hz or better is preferred for the 'analogue feel'. Professional systems overcome this by complex, multiple loop synthesis, which can be both expensive, and when the filters are taken into account, bulky.

Amateur equipment practice has tended to take a more pragmatic route. Minimum step size in the synthesizer is usually 1KHz, while increments between are achieved with a separate control knob on older equipment, or by a digital analogue converter (DAC), operated by the last digit of the frequency set. The analogue voltage tunes a VXO in the rig, which may either be the conversion crystal in the synthesizer, or may be the reference crystal itself. Care must be taken to ensure that the pulling range is accurate or the '100Hz' steps will show a jump in one direction or the other at the 1KHz increments. Actually, many rigs do show this if observed carefully; the reason is that, until recently the DAC was a fairly simple affair consisting of a resistor array and switching transistors. More recently, real DACs have been used, but this is not a complete solution, since the VXO is unlikely to be linear to the required degree, so some step nonlinearity is inevitable. At least if the end points are not seriously wrong, this should not be a problem. Some rigs do tune in 10Hz steps, especially on HF; this is an extension of the technique to 100 steps instead of just 10. Again, there is the issue of the 1KHz crossover points; but with 10Hz steps this can be made less noticeable on a well designed and adjusted rig. This is the final point on PLL synthesis; properly designed, it works very well indeed, but who has not heard of 'synthesizer whine', or read reviews of otherwise excellent radio limited in performance by synthesizer noise causing reciprocal mixing? My own experience includes a rig which preferred not to go mobile; the synthesizer lock loop would go out of tune after a surprisingly repeatable mileage, only finally cured by the application of much wax to the VCO coil.

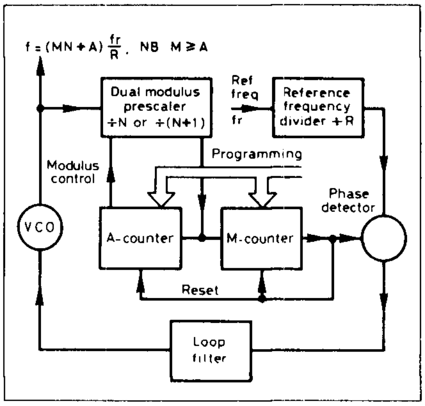


Fig 3: Dual Modulus prescaling

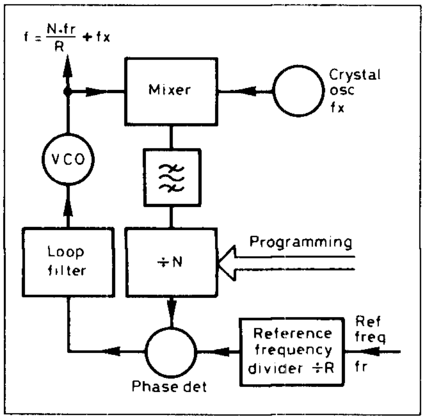


Fig 4: Mixing in the loop

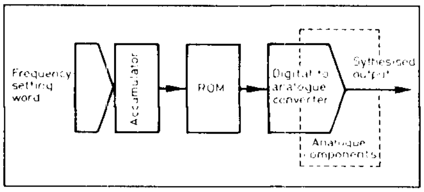


Fig 5: Direct frequency synthesizer

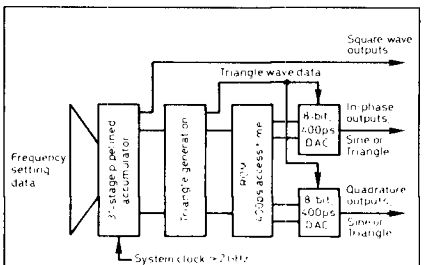


Fig 6: System block diagram of SP2002

A CURE-ALL?

WILL DIRECT DIGITAL synthesis cure all these ills? The answer is "no, not yet", but in the fairly near future this may turn into a qualified "yes". The basic direct digital synthesizer consists of an arrangement to generate the output frequency directly from the clock and the input data. The simplest conception is shown in **fig 5**. This consists of a digital accumulator, a ROM containing the pattern in digital form of a sinewave, and a digital to analogue converter. In general, the DAC will be followed by some filtering usually a low-pass filter somewhere just below half the clock frequency.

Dealing with the accumulator first, this is simply an adder with a store at each bit. It adds the input data word to that in the store. The input data word only changes when the required frequency is to be changed. In the simplest case the length of accumulator is the clock frequency divided by the channel spacing, although it is usually calculated the other way round, ie if a 5kHz channel spacing up to 150MHz is needed, a clock frequency of at least 300MHz will be required, due to the Nyguist sampling theorem, and 300MHz divided by 5KHz is 60,000. More conveniently, a 16 bit accumulator gives a 65,536 steps. If the step size is to be 5KHz, then a clock frequency to the accumulator of 65,536 x 5KHz is needed, i.e. 327.68MHz.

These numbers are the basic values chosen for the first commercially available VHF DDS, the Plessey SP2001. What the numbers mean is that by applying a binary number multiplied by 5KHz, ie the number to be applied is the 'channel number'. This makes programming particularly easy, especially as the device has a parallel input data format: DIP switches were used on the evaluation boards. The device produces any frequency in the range covered, ie 5KHz to over 100MHz in a single range, without any tuned circuits, although of course it does depend for stability on the clock source, which would normally be crystal controlled. Other frequency increments are available; any multiple of 5kHz by selection of input data, and others by choice of clock frequency, eg 6.25kHz requires a clock at 204.8MHz, with a two channel (2 x 3.125kHz) program word. Of course, frequency multiplication or mixing can be used to take the output to higher frequencies.

NEW DEVICE

RECENTLY, A NEW DEVICE (SP2002) has been announced [6]. This device, which is shown in block diagram form in **fig 6**, is more complex than the SP2001, since it has on chip DACs and facilities for square, triangle and sine outputs. Two DACs are needed because both phase and quadrature signals are available in true and complement form. The most significant bit (MSB) from the accumulator feeds the square wave output buffer direct. In parallel, the next 7 bits from the accumulator, which digitally represent a sawtooth waveform, feed a set of XOR gates, under control of the MSB, so that a triangle output is generated, in digital form, at this point in the circuit. Actually, two triangles in quadrature are generated. These can be digitally steered to the output DACs, or can be used to address a ROM containing data for sine and cosine waves; only 90 degrees is needed, since all four quadrants can be generated from one. Finally, if selected, the digital sine/cosine is fed to the DACs for conversion to analogue form.

This device was designed to operate at up to 500MHz output frequency, with 1Hz steps, so the full range is 1Hz to 500MHz, again in a single 'range' with no means of or requirement for tuning. The clock frequency is necessarily very high; the quadrature requirement adds a further factor 2, so nominal clock frequency is 2^31Hz, i.e. 2.147483648Ghz. This must be supplied with crystal controlled stability, although since phase noise is effectively divided down in the synthesizer, and the frequency is fixed, it can be generated by a simple multiplier from a crystal source. Output spectra are shown in **figs 7 and 8**. Fig 7 shows a clean output at exactly one quarter of the clock frequency. Noise floor is essentially generated by the spectrum analyser. Fig 8, however, is more representative of the general case; it shows a spectrum a 225MHz, a frequency not integrally related to the clock frequency. Here the spurs have come up to a level about 50db below the carrier. This is the fundamental limitation of the DDS technique, and where most development work is going in the future. The limit comes from the finite word size and accuracy of the DAC, and incidentally the ROM, although this could have been made bigger fairly easily.

Fast DACs are difficult to make accurately, for two reasons. The first is technological; IC processes in general limit to a component matching accuracy of about 0.1%, ie 9 or 10 bits accuracy in a careful design. Other techniques, such as laser trimming of the resistors, can be used on some processes. However this tends to use older, slower processes, and especially requires large resistors for trimming, which slow the DAC settling time. The second problem is simply the requirements on fast settling; the DAC is required to get to the final value quickly, and this will in general happen with the smallest number of bits. An eight bit system was chosen to fit the process capabilities and the device requirements, but this leads to a limitation in the high level of spurious signals present. Basically, although some frequencies are very clean, in the worst case the spurious level is 6N dB below the carrier, where N is the number of effective DAC bits. For an 8 bit system, this give -48dB. Oversampling, ie running the clock at more than twice the output frequency gives some improvement, at 3dB per octave, by improvement of the DAC resolution, but only up to a limit of the DAC's accuracy. Typically this is about 9 bits or -54dB. In some applications, this may not be serious, since filtering can remove all but the close in spurs; phase locked translation loops into the microwave region also act as filters of relatively narrow band, while retaining the 1Hz step capability.

In the amateur rigs available using DDS, the synthesizers operate at relatively low frequencies and are raised to the working frequency by PLL techniques; complicated, but still making the fine frequency increments available without compromise of PLL design. The devices used are probably CMOS types, with DAC accuracy around 12 bits. This gives spurious signals which should theoretically be -72dB referred to the carrier, which is adequate in most cases especially with some filtering from the PLL.

What of the homebrewer? Where does he get the devices? Does he want them anyway? All the devices mentioned by part number are commercially available; although the exact types in the Japanese equipment are not known to the author, US types of similar performance are also available. However, prices are likely to be high, at least for the next year or two. This is especially true for the VHF and UHF types, which are not really ideal anyway for single frequency (as opposed to 'hopping') radio receiver applications.

PRACTICAL DDS

A VERY USEFUL ARTICLE on a practical DDS appeared some years ago [7]. This showed how a synthesizer could be made using available TTL and a ROM. However, the output frequency was limited to 1MHz. Real DDS systems could be built by the amateur, and I intend to explore this for my own use. This is not intended to be a constructional article, so the comments made here are more in the nature of hints, without guarantee of success, but they outline a possible route to follow.

First, what do we need? An output up to say 10MHz would be useful for further conversion upwards in a mixer or PLL. Steps of 10Hz or better are essential. This determines the clock frequency (>20MHz) and the accumulator size (21 bits). For convenience, the clock should therefore be 20.971520MHz. Standard TTL will work at this frequency although I would prefer HC CMOS.

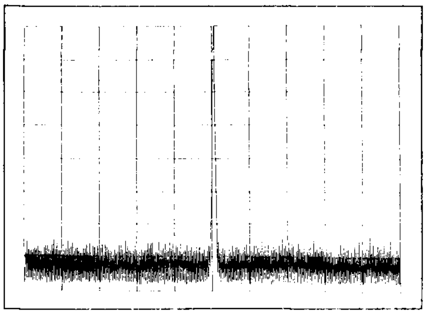


Fig 7: Output spectrum at 250MHz, with a 1GHz clock. 10MHz/div, X axis; 10dB/div, Y axis.

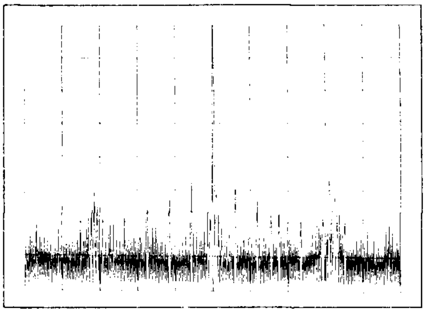


Fig 8: Output spectrum at 225MHz, with a 1GHz clock. 10MHz/div, X axis; 10dB/div, Y axis.

A square wave would fit most applications in radio; there is no real need for a sinewave in local oscillators. However, the MSB output mentioned above has one problem; even higher spurs! Think of it as a one-bit DAC. Then -6NdB is not very good spurious suppression. The reason is that the square wave can give only increments in time, limited to the clock period in resolution. The sinewave also interpolates in amplitude, to the degree offered by the DAC accuracy, and so gives much better spurious control. However, if we are prepared to accept harmonics, which can be filtered out later if required, then a triangle is just as good in this respect as a sinewave. Since most commercial ROMs have access times of around 100ns, 10MHz is the limiting update rate, so a triangle has the further advantage of not needing the ROM at all. The DAC remains the major problem. Current low cost DACs are limited to 8 bits and settling times of over 1 microsecond.

A BETTER APPROACH

A MUCH BETTER APPROACH would be to use a current output DAC such as the SP9768 (8 bits, 5ns) or SP9770 (10bits, 12ns). These are ECL input devices, so will need interfaces to the CMOS accumulator and XOR gates; an ECL accumulator and XOR array should be considered. The latter device would give some 60db spurious suppression alone. Alternatively, it should be possible to add more bits using external discrete components, probably one further LSB and possibly two MSBs. Although temperature tracking of these components would not be good, the system would be operating in a fairly benign environment, so this is not critical. The DAC so produced would therefore be 12 or 13 bits accurate, the accuracy of the added bits coming from preset 10 turn potentiometers and some careful DC measurements. A full 4 digit DVM (or better) will be required for this operation. Of course, a better solution would be a fast 12 or more bit DAC; these are becoming available, but at a price.

Putting the system together should be fairly simple, using good VHF practice in order to avoid accidental cross talk onto the output lines. Power supplies need good decoupling for the same reason. A clock oscillator at this frequency should not pose problems. The only set up adjustment is the DAC if built up as above; ideally, DDS systems are set up free. Finally a check on a spectrum analyser will show how well it works.

CONCLUSION

IN CONCLUSION, DIRECT frequency synthesis is a promising technique for the future. There are certainly obstacles to overcome before it is generally used without a phase locked loop for filtering and conversion in HF radio applications, but it is already in use in amateur equipment. Eventually, we may see faster, more accurate DACs and new filtering techniques combine to produce a 100dB clean synthesized oscillator, with no set-up procedures at all and with extremely narrow channel spacing at low cost.

REFERENCES

1) Pat Hawker, 'Technical Topics' *RadCom*, December 1988 pp 957 - 958
2) Peter Hart 'ICOM IC-725 HF Transceiver' *RadCom* September 1989 pp 56 - 58
3) Peter Martin 'A receiver with noise immunity and Frequency Synthesis', *Radio Communication Handbook* (RSGB), fourth Edition pp 10.104 - 10.108.
(4) Plessey Semiconductors 'Frequency Dividers And Synthesizers IC Handbook', especially pp 304 - 306.
5) Plessey Semiconductors Data Sheet SP8853
6) Plessey Semiconductors Data Sheet SP2002
7) JHJ Dawson 'Direct Digital Frequency Synthesizer' Wireless World December 1981 pp 40 - 43.

Power Supply for GaAs FET Amplifier

By Zack Lau, KH6CP
ARRL Laboratory

I have used the following circuit to supply numerous GaAs FET low-level amplifiers with good results. It's an active supply to compensate for device and temperature variations. It fits in well with the no-tune concept of microwave circuitry. After all the work of coming up with RF circuitry that doesn't need tuning, why should you need a set of trimmers to adjust DC bias parameters?

Circuit Description

This circuit isn't original—a similar circuit appeared in the May 1989 issue of *QST* in a GaAs FET preamplifier article by Al Ward. Unfortunately, the author didn't supply a printed circuit-board pattern, which is the need this article fulfills. In addition to Al's article, there is also AN-S003, put out by Avantek, which gives more information on this type of biasing circuit. Another useful reference is a paper by Gary Franklin, "RF and Microwave, Transistor Bias Considerations," that appeared in the November 10-12, 1986 *RF Expo East Proceedings.* This conference was sponsored by *RF Design Magazine.*

The negative bias generator is a simple NE555 oscillator circuit feeding a diode rectifier. The use of high speed rectifier diodes allows a higher oscillator frequency to be used, allowing smaller filter capacitors. Besides, you want the negative supply to come up as quickly as possible, reducing the stress on the GaAs FETs. Common switching diodes like 1N914s will work if you don't exceed their limited current capabilities. Three terminal regulators provide stiff voltages for the active bias circuitry that don't exceed the FET limits.

The active circuit uses R3 to set the desired GaAs FET drain current. R1 and R2 act as a voltage divider that sets the drain voltage. Don't forget that in addition to the emitter base voltage drop, there is often a drain resistor that adds yet another voltage drop.

The Hand-Waving Explanation of Circuit Operation

The circuit is based on a PNP transistor which tries to maintain 0.7 volts across the base-emitter junction. If this voltage is too low, the current across the base is reduced, making the FET gate voltage more negative. The more negative gate-source voltage reduces the drain current, which increases the emitter voltage. Since the base voltage is approximately fixed, the base-emitter voltage is therefore increased until it's approximately the voltage drop for a saturated transistor. Similarly, too low a drain current increases Vbe, increasing the current across the base, making the FET gate voltage more positive, which increases the drain current.

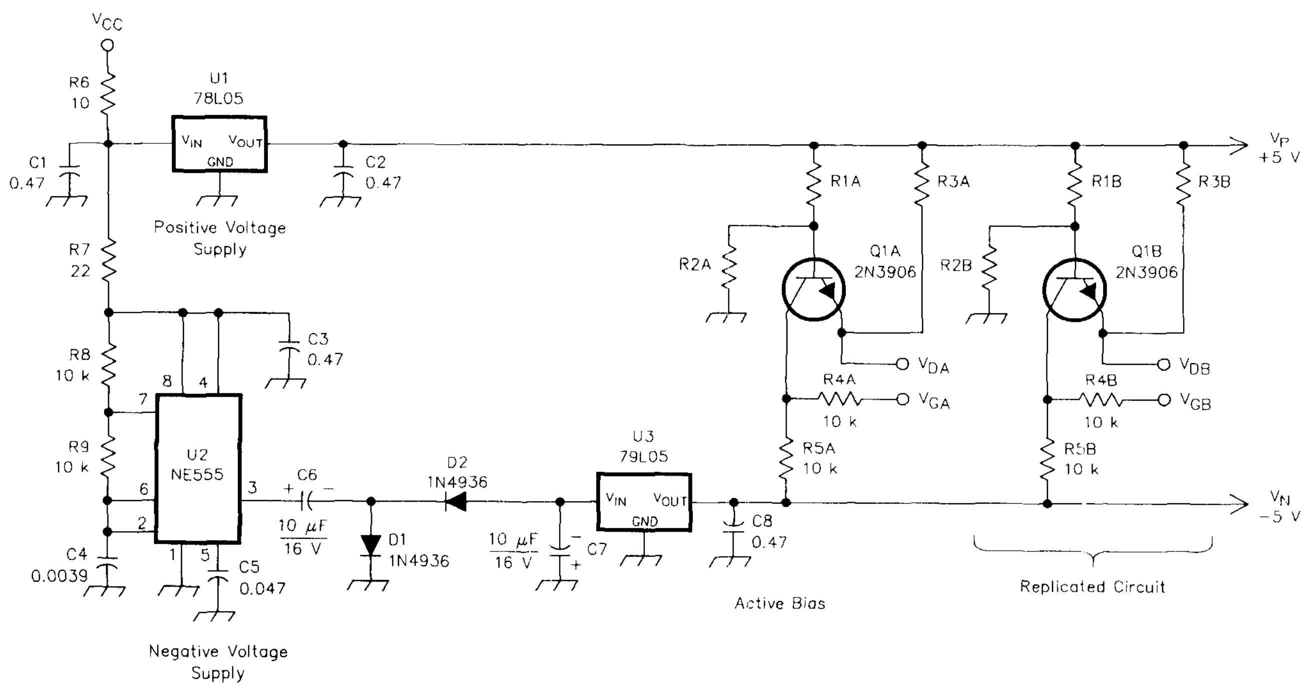


Fig 1—Schematic.

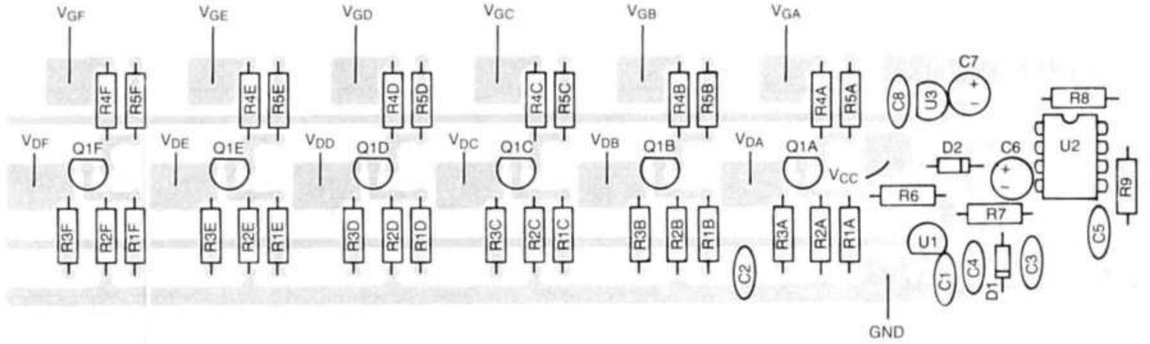


Fig 2—Component Layout

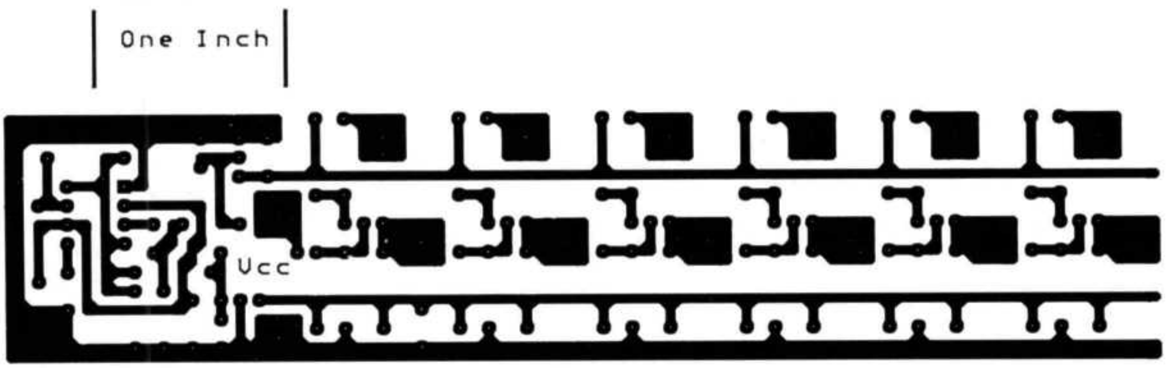


Fig 3—Etching Pattern

Choosing Values for R1, R2, R3

Definitions

Vp = The positive supply bus
Vdd = The drain voltage supplied to the GaAs FET amplifier (don't forget any voltage drops from series resistors)
Id = The drain current
Vbe = The base-emitter drop (0.7 V, but temperature sensitive)

(1) R3 = (Vp-Vd)/Id
(2) R1 = R2*(Vbe + Vp-Vdd)/(Vdd-Vbe) approximate equation

I have had pretty good luck using equation 2 with R1 and R2 values in the low to mid kΩ range. These equations seem to be close enough for picking standard values.

Experimental Results

The resistance values are assumed to be those marked on 5% quarter-watt resistors. The voltages are measured with a Fluke 77 multimeter.

R1	R2	R3	V_{dd}	V_p
2k	3.9k	15 ohm	3.92 V	5.03 V
2.2k	4.3k	51 ohm	3.99 V	5.04 V

Construction

The printed circuit board is designed to allow a variable number of devices to be powered. If you only need to power a few devices, you might wish to cut the board short. With a bit more work, more than 6 devices can be accommodated by replicating the active bias pattern and extending the board. It may be necessary to heat sink or replace the 78L05 with a bigger device if you supply too many FETs. The pads are purposely made big enough for swaged terminals, which are ideal for this application if you have access to the expensive swaging tool ($300 + new).

Reliability

Since the negative supply comes on shortly after the positive supply, questions about damaging the FETs invariably arise. So far, out of over 20 devices operating from 2304 to 10368 GHz, only one failure has been noted. I suspect this reused 72084 failed as a result of excessive heat or static discharge, as it's not always easy to remove devices from circuits that didn't work as well as hoped. Actually, my biggest problem with microwave devices are the leads breaking off while attempting to reuse them!

There is a trade-off between current limiting and voltage limiting. By raising the open circuit voltage, you can increase the drain series resistance, improving the current limiting abilities of your supply. However, if the voltage is too high, you can exceed the V_{DS} rating of the device. There are situations where a more sophisticated supply is needed to avoid exceeding device ratings. An example might be a 5-volt maximum V_{DS} FET that needs a drain bias of 40 mA at 4 volts, as well as 51 ohms of series resistance for RF stability!

Correspondence

Comments on the October Safari-4 QRP Transceiver Article

I believe a correction is needed in Wayne Burdick's excellent article on the Safari-4 QRP transceiver as described in *QEX*, October 1990. This concerns the sidebar, "Why Dual Conversion," which is printed on page 16.

In the fifth paragraph of the sidebar, the statement is made, "A direct-conversion receiver cannot provide single-signal reception, because there is no way to prevent signals on both sides of the VFO from producing AF outputs." This is untrue in both theory and in practical application. The phasing method of receiver architecture can provide a direct-conversion, single-signal receiver. There have been several articles on this method, beginning with the "original" articles by Villard and Thompson, and Norgaard in *QST*, 1948. More recent application work has been described in *QST*, September 1969, by Taylor (W1DAX) and also by Shubert (WA0JYK) in *Ham Radio*, August 1973.

Although both of these articles used components commonly available at the time of publication, they both accomplished quite good single-signal reception with direct-conversion receivers. The phasing architecture is even more practical with the components we have now, such as high-speed, low-power CMOS ICs for quadrature local oscillator injection sources over a reasonably wide band of frequencies. They could easily generate the 8.80 to 8.87 MHz signal needed for N6KR's frequency scheme.

I've always considered the phasing method much more elegant than the brute force crystal filter method for single-signal reception. With today's modern components, this approach is not only theoretically possible, it can be readily implemented with commonly available components, even in a QRP transceiver. I believe a correction to the article would be helpful for readers who are not familiar with the phasing method so that they are not led to false conclusions that would discourage experimentation along these lines. —*John H. Klingelhoeffer, WB4LNM, 1500 Kingsway Drive, Gambrills, MD 21054*

[Reply] *Mr. Klingelhoeffer is correct in pointing out that the phasing method of direct conversion can yield single-signal reception, and that it should be considered as a design option even for QRP. In fact, the use of this technique should result in fewer spurious responses, since only one signal source would be required to generate and detect the 8.8-MHz IF range.*

I have two concerns, however. First, the Safari-4 in particular is intended for outdoor use, and the low VFO range obtained with my mixing scheme (3.9 MHz) offers better temperature stability than a VFO operating at 8.80 MHz. Second, more importantly, is circuit complexity. Even the most widely read amateur reference—the Handbook*—expresses doubts, especially in this passage:*

*"Some designers have contrived elaborate circuits that, by means of phasing networks, provide single-signal reception. Unfortunately, the circuit becomes nearly as complex as that of superheterodyne. The benefits obtained are probably not worth the effort." (*The 1990 ARRL Handbook for the Radio Amateur, *pg 12-8.)*

By contrast, the "brute force" method works well for CW. It is easy to construct a Cohn filter from microprocessor crystals, and the NE602 mixer requires very few components.

I hope that amateur experimenters will rectify the shortage of good information on using the phasing technique. I, for one, will be working on it, and I offer Mr. Klingelhoeffer the challenge of presenting a simple, low-current design suitable for QRP CW work.—Wayne Burdick, N6KR, 446 Mt. Hope Street, Unit 9, North Attleboro, MA 02760

Comments on "The Fourth Method"

"The Fourth Method: Generating and Detecting SSB Signals," by van Grass, PA0DEN, in September 1990 *QEX*, parallels, in some respects, work I have been doing over the past few years. I have just completed a 100-mW, 75-m SSB exciter based on a variation of the third, or Weaver, method of SSB.

PA0DEN's sidebar explanation of using analog multiplexers as frequency shifters is excellent. The analogy based on selsyns quickly gets to the heart of complex-signal (or quadrature-signal; or image-reject-mixer) techniques. I wish to make a correction: The device PA0DEN describes, with one rotor winding and two stator windings 90 degrees apart, is a resolver. A selsyn, or synchro, contains one rotor winding and three stator windings 120 degrees apart.

The two stator windings in a resolver require four wires, giving signals at 0, 90, 180 and 270 degrees, as described by PA0DEN. This leads to a four-position multiplexer cycling through the four positions in one cycle of the carrier frequency.

The signal at 180 degrees is −1 times the signal at 0 degrees, and the signal at 270 degrees is −1 times the signal at 90 degrees. Thus, since multiplying a signal by −1 is easy to do, only the 0-degree signals, in many cases, need to be processed. These two signals are often called I (in-phase) and Q (quadrature), and are often referred to as the components of a complex signal or a two-phase signal.

The three stator windings of a synchro require only three wires (because the sum of the voltages in the three windings is zero), giving signals at 0, 120 and 240 degrees. This leads to a three-position multiplexer cycling through the three positions in one cycle of the carrier frequency.

The three signals at 0, 120 and 240 degrees are often called A, B and C, and can be referred to as the components of a three-phase signal. Anything you can do with a complex signal you can do with a three-phase signal. Conversion from one to the other can be accomplished by a Scott-T transformer or circuit, well known in the electromechanical branches of electrical engineering.

In some cases, the three-phase approach leads to simpler circuitry than does the complex approach.

The Weaver method was published in 1956 ("A Third Method of Generation and Detection of Single-Sideband Signals", by Weaver, in the *Proceedings of the IRE*, December 1956) and in 1957 ("The Third Method of SSB," by Wright, in *QST*, September 1957), and then sank without a trace. The Weaver method is described in recent Signetics linear-integrated-circuit data books.

In essence, Weaver generated SSB by first down-converting the input audio signal into an array of dc-to-1500 Hz signals, low-pass filtering these signals, and then up converting to the final RF output frequency. Weaver used 0 and 90 degree signals, but the same method can be used with three-phase signals.

The need for sharp, matched audio low-pass filters made the Weaver method unattractive in 1957. but parts have come a long way since then. Modern parts make the Weaver method much more attractive than it once was. *—Peter Traneus Anderson, 990 Pine Street, Burlington, VT 05401*

A Call for More Experiments Combining Microwave and CDMA Technology

I read the comments by WA1OCK in the December 1990 *QEX* about an "Amateur Web." I have also been noticing other comments elsewhere about similar ham networks and plans, including a comment in *QEX* (or *QST*?) about something called "Dream-Net."

I also believe that the future of ham radio must bring faster and bigger ways to get all sorts of communications connected between area.

I note that the cellular providers, NYNEX and PACTEL (and now Europe's NOKIA) are rapidly experimenting with a more recent style of modulation called "CDMA" (code-division-multiple-access), which promises to allow segments of RF spectrum to support even more simultaneous conversations, and with no *absolute* limit to that number, except that the signal starts to degrade a bit if users exceed a specified quantity. One of the main providers of integrated circuits used by CDMA is Qualcomm, and chip sets will soon be both available and economical.

I am only one ham, but for the last two years I have really enjoyed experimenting with both 10- and 24-GHz microwave sources. I believe that Amateur Radio could benefit from more experiments that combine microwave and CDMA technology, and I intend to continue along those lines.

Once Qualcomm MMICs become available for CDMA modulation, and I get them to work with some of my existing 10- and 24-GHz wide-band equipment, I feel that the result may offer a different solution for getting many channels of voice, data and telemetry from one place to another.

Ham radio needs more than finding more repeater pairs, or boosting packet LAN speed to 9600 bauds. With CDMA and the frequencies we have, including microwave, we could have a communication system better than any. As an example, if only 1 MHz of our 2-meter band was converted to CDMA, it could support over 1400 simultaneous two-way conversations without interference. *[Ed. Note: CDMA, better known in the amateur community as spread spectrum, is authorized to US amateurs in bands only above 420 MHz.]* It would mean that rush hour in Chicago would allow every ham the ability to use the same single wide-area repeater at the same time, without any interference, and allow a similar group only 40 miles away to repeat the whole process with another repeater! To top it off, simple microwave equipment as I described above could easily link such repeaters, allowing expanded and much-improved mobile performance! No "remote receiver" needed! Just transmit instructions to the local CDMA repeater, and link up automatically (and privately) with a similar CDMA repeater user a couple of states away! We don't even come close, now.

With the bandwidth available at higher microwave frequencies (500 MHz at 10 GHz), hams could easily be sending thousands of simultaneous conversations or data streams between major areas, and the cost can easily be lower than the primitive linking techniques we presently use.

I am not sure if hams are interested enough, but we certainly have the talent to implement much better communication links. If private companies can provide instant computer terminal communications, airline reservations and mobile telephones, we hams can still do it *better!—Alan Rutz, WA9GKA, 7102 W 500 S, La Porte, IN 46350*

Components

By Mark Forbes, KC9C
PO Box, 445
Rocklin, CA 95677

BE CAREFUL!

Regular readers may have noticed that my January column ran in February instead of its scheduled time. The reason was that I had back surgery on a ruptured disk a few days before Christmas and wasn't in shape before that to pound the keyboard to produce the January words of wisdom.

The injury was ham related, so I wanted to warn you not to pull the same stunt I did. I got a great buy on a 40-foot push-up tower, rotor, and TA33jr beam...$150. Unfortunately, I couldn't wait for my friends to come over to help and proceeded to try to put it up myself. The first two sections went up fine, but the third got stuck about 8 feet up. I bent from the waist and gave a tremendous tug...which was followed by a loud "pop" from my lower back. In short, I now refer to my great bargain as my "$9,000 antenna." Moral: Wait for help, and always lift with your *knees* bent!

DIGITAL MEMORY

This month's treatment of digital electronics focuses on memory devices. There are basically two types of digital memory: volatile and nonvolatile. Volatile memory loses its contents when power is removed, and nonvolatile memory retains the stored data even when the power is turned off. Making a gross generalization, the most common volatile memory is RAM (random access memory), and the most common nonvolatile is ROM (read-only memory) or EPROM (erasable/programmable read-only memory). I'm not going to get into the intricate details of how the memory components work at a transistor level; instead I'll describe them from a functional standpoint.

RAM is what the majority of memory in your PC is. It actually comes in two flavors—static and dynamic. Static RAM is very fast, but fairly expensive. It is not used in great quantity because of its cost. Static RAM stores data for as long as power is applied, without any attention. It is frequently used where a small amount of memory is needed, such as in your HF or VHF transceivers. It is easy to manage. To store data, the "address" signal is placed on the RAM and then the data is stored in that location. The address is just like it sounds: It is a unique location on the chip, just like your house address specifies a unique location on your street. Retrieving data is done in the same way. You can read data from the memory as many times as you want, and it won't be lost as long as power is maintained.

Dynamic RAM is less expensive and is what the memory in your PC is. Dynamic RAM uses capacitive effects to store the data, and as such, the charge on the capacitor can bleed off. Unless it's recharged, the data will be lost. So, dynamic RAMs have to be refreshed periodically. This is done by the computer reading the data and then writing it back to the RAM. A tedious process, but it's worth it because of the cost differential.

ROM (and EPROM) are permanent memory, but they can't be rewritten. So, they are only used to store information that doesn't change. For example, part of the computer's operating system (that is, the program that tells the computer how to function) is in ROM since it never changes. Similarly, the memory that tells your transceiver how to operate is stored in ROM. EPROM can be erased and reprogrammed, so it's often used to develop projects so that the designer can change the operating characteristics.

There are other, more exotic types of memory, but those will have to wait for another month. Next time, I'll introduce microprocessors.

900-MHz POWER AMPLIFIERS

Motorola has a new series of integrated power amplifiers that can be used in the 902-MHz ham band. The series, the MHW807, is designed for cellular telephone applications, but can easily reach the low end of the 902-MHz band. It can be extended somewhat upward with potentially reduced gain.

The '807 (an appropriate number for an amplifier!) gives a power gain of 35 dB, and is specified to operate from 870 to 905 MHz. With only 1 mW of drive, the chip will deliver 6 watts of output power.

A second amplifier product is the MRF10120 microwave power transistor. This is intended for long-pulse applications, so the ratings are *not* for continuous service such as FM. To use for FM, you must downgrade the maximum power ratings.

These transistors cover 960 to 1215 MHz, so are applicable in both the 902-MHz band and the 1.2-GHz band. With 15 watts of drive they deliver 120 watts of output power. Power supply requirement is 36 V dc. To complement the MRF10120, Motorola has also introduced the MRF10030 driver and MRF10005 predriver, with 30 and 5 W output, respectively.

For more information on any of these products, contact Motorola Semiconductor Products, PO Box 20912, Phoenix, AZ 85036.

HIGH CAPACITY NiCd BATTERIES

Gates Energy Products has just introduced a new series of nickel cadmium batteries that offer the highest capacity-per-size available. The batteries are available in AA, C, D, and some specialty in-between sizes. As an example of the capacity of these batteries, the AA cells' capacity is 800 mA/h. Typical AA NiCds are generally 500-600 mA/h. The D-size battery has a capacity of 5.0 A/h!

The peak current is also very impressive. The AA cell can deliver 18 amps for one second—the D cell 110 amps! The price is also right on these batteries. The AA cell costs about $1.25, the C cell $3.50, and the D is approximately $6.00. Contact Gates at PO Box 667650, Charlotte, NC 28226.

FLUSH-MOUNTED NUTS

For a real professional finish to your projects, Precision Metal Products now offers flush-mounted stainless-steel nuts. These nuts press-fit into a hole drilled in the chassis, providing a similar function to a tap, but with greater strength. The nuts can mount into panels from 0.04 to 0.312 inches. Contact Precision Metal Products, PO Box 6026, Peabody, MA 01961. They will send literature plus free samples.

20-BIT AUDIO DAC

If you're interested in experimenting with digital audio signals, the Analog Devices AD1862 should be fun to work with. This 20-bit DAC is specially designed for audio work. This chip has an impressive dynamic range of 102 dB, and a signal-to-noise ratio of 119 dB. The digital interface of the DAC receives serial data in 2's complement format. Power is supplied with +12 V dc and -12 V dc. This impressive chip is priced at less than $20. Contact Analog Devices at 804 Woburn Street, Wilmington, MA 01877, or phone 617-937-1428.

Conducted by: Stan Horzepa, WA1LOU
75 Kreger Drive
Wolcott, CT 06716-2702

FCC CITES PBBS BUSINESS COMMUNICATIONS

On January 25, the engineer in charge at the FCC's Norfolk Office of the Field Operations Bureau sent notices of apparent violation to ten PBBS operators indicating that they may have operated their Amateur Radio stations in violation of Section 97.113(a) of the FCC's Rules because they apparently used the Amateur Radio Service to facilitate the business activity of an organization. Shortly thereafter, $300 notices of apparent liability were sent to WA3QNS, the alleged originator of the message, to N3LA, the first PBBS on which the message allegedly appeared, and to KJ4LQ, operator of the PBBS from which a complainant read the message.

The offending "ALL@USA" message, originated by WA3QNS and relayed by a number of PBBSs in early January, asked recipients to phone a "900" number. The packet-radio message "was defective on two counts," ARRL Executive Vice President David Sumner wrote in March *QST*. By promoting a "900" number, the message facilitates business activities, and having nothing to do with Amateur Radio, the message fails to qualify as an information bulletin, thus making it "broadcasting," Sumner said in an editorial.

According to PBBS operator Tom Clark, W3IWI, "The implications of the action. . . are absolutely appalling. What is implied is that each and every station in a store-and-forward network is responsible for the actual message content passing through each node. The PBBSs were cited because their calls were in the message header 'audit trail.' The FCC's action states that each PBBS SYSOP is personally responsible for the 'correctness' of all messages merely passing through his system."

In his *QST* editorial, Sumner notes the understanding reached with FCC staff in 1986 when automatic control of packet retransmissions was permitted. The editorial is firm in its support for "downstream" packet-radio operators; the League's belief is that the Commission does not intend that individual PBBS operators "put their licenses on the line for something over which the Commission has acknowledged they have no control." However, the editorial does suggest that the packet-radio community "take heed and pay more attention to the appropriateness of 'broadcast' packet-radio traffic."

SAN FRANCISCO TO BE SITE OF 10TH ARRL COMPUTER NETWORKING CONFERENCE

Meeting on January 18 and 19, the ARRL Board of Directors selected the San Francisco area as the site for the 10th ARRL Amateur Radio Computer Networking Conference. The Northern California Packet Association will host the conference which is scheduled for the weekend of September 27-29. *Gateway* will publish further information when it becomes available.

INPUT FOR NEW 222-225 MHz BAND PLAN SOUGHT

The ARRL VHF Repeater Advisory Committee (VRAC) and VHF-UHF Advisory Committee (VUAC) seek input from 222-225 MHz users, including packet-radio operators, in order that the ARRL Membership Services Committee may coordinate a recommendation for a new 222-225 MHz band plan.

SPACE STATION PACKET RADIO OPERATIONAL

Musa Manarov, U2MIR, on board the Soviet space station Mir, received a packet-radio station by way of a cargo rocket that was launched on January 14 and had the station on the air by January 20. Musa operates in two packet-radio modes: live keyboard-to-keyboard operation using call sign U2MIR-Ø and mini-PBBS operation using call sign U2MIR-1. Virtually all of the packet-radio operations occur on 145.55 MHz.

The U2MIR-1 PBBS commands are as follows:

B	(bye) disconnects you from the PBBS.
H	(help) displays a help file.
J	(log) displays a list of call signs heard.
K n	(kill) deletes your message numbered *n*.
KM	(kill mine) deletes all of your messages that you have read.
L	(list) lists the 10 latest messages.
M	(mine) lists the 10 latest messages to/from you.
R n	(read) reads message numbered *n*.
S call	(send) sends a message to *call*.

RUDAK 2 LAUNCHED

Radio M-1/RUDAK-2 was launched from the Plesetzk spaceport in the USSR on January 29 and, now that this Amateur Radio payload is on orbit, it has been renamed AMSAT-OSCAR-21 (AO-21). AO-21 was built by AMSAT-U-ORBITA in the Soviet Union and AMSAT-DL in Germany. This cooperative effort began in the summer of 1989 when an agreement was reached between representatives of both organizations. AMSAT-U-ORBITA designed and built the Mode B linear transponder, the command receiver, the telemetry systems and the power supplies, while AMSAT-DL designed and developed the digital transponder known as "RUDAK-2."

The first version of the RUDAK 2 Bulletin Board System (RBBS) will be loaded as soon as the initial on-orbit check-outs are completed. It will be a preliminary release capable of storing personal mail and messages of common interest. The following instructions from DK1YQ describe how to use RBBS.

After connecting RUDAK, you will receive a log-on message similar to the following:

Welcome to the RUDAK II Bulletin Board System V1.00
Logged in at yy-mm-dd hh:mm:ss, x Users
This is a preliminary release.
Please report deficiencies to DL2MDL.
73 de AMSAT-U-ORBITA/AMSAT-DL/RUDAK-Group.

Enter H for Help
UA3CR de RUDAK>
The following commands are presently supported:
H outputs a summary of available commands.

L call m-n shows the list of messages addressed to *call*. *M-n* (eg, 1-5) determines a range of message numbers. Messages are numbered individually for each addressee starting with 1 for the newest message. *Call*, *m-n* or both can be omitted. The default values are your call sign for *call* and 1-5 for *m-n*. Ranges are always truncated to a maximum of 10. *Call* can be either an Amateur Radio call sign or the name of a board.

R call n reads message *n* addressed to *call*. The default value for *call* is your call sign.

E call n erases message *n* addressed to *call*. The default value for *call* is your call sign, *n* is mandatory. You can only erase messages created by you or addressed to you. If you get the 'File busy: Try later!' message, someone else is currently reading or listing the message.

S call subject sends a message to *call*. *Subject* can be any text string delimited by a carriage return <CR> describing the message content. It will be printed by the L command and shows up at the beginning of the message. Both *call* and *subject* are required. They can be entered on separate lines, in which case they are prompted as follows:

S
To> ALL
Subject> Important News!

A message can be ended by disconnecting, by timing out after loss of signal, by entering <CTRL-Z> or by entering <CR>.<CR> at the end of the message.

—from AMSAT News Service

LO-19 PBBS OPERATIONAL SOON

The command stations of LUSAT-OSCAR 19 (LO-19) are in the process of loading the PBBS software into the satellite. At the present time, the PBBS is not open to the public as testing of the system is still ongoing. Once system shakedown is complete, AMSAT-LU will publish the access code to be entered into the PACSAT ground station software configuration file to allow general use of the PBBS. Among the first messages to be posted on LO-19 will be a message from the President of Argentina, who is an Amateur Radio operator.

—from AMSAT News Service

NEW PACKET-RADIO SOFTWARE AVAILABLE

New packet-radio software has become available during the past month. The following information comes from *CompuServe's HamNet* and other sources. Although, all of the delineated software may be downloaded from *HamNet*, some of the software may be downloaded from other sources such as other on-line services, as well as, landline and packet-radio BBSs.

ARES/Data V1.5, file name: ARES15.EXE, binary, 169752 bytes

ARES/Data is a multi-user, general purpose database program. Users connect to the database with any type of AX.25 packet-radio station. The main database runs on an IBM PC.

This version of ARES/Data now works as an application under the G8BPQ Switch. As a result, your PC can serve as a network switch, like NET/ROM, and simultaneously offer service as a database. It also means that you can use any type of KISS TNC for the database (a WA8DED firmware-equipped TNC or DRSI PC*PA is not required, but is supported).

BAYCOM.ZIP, file name: BAYCOM.ZIP, binary, 143104 bytes

BAYCOM is the IBM PC TNC emulator software from Germany that was described in the February installment of *Gateway*. The English translation of the BAYCOM manual is contained in the accompanying file named BAYMAN.ZIP (binary, 41984 bytes).

BCAST20.SIT, file name: BCAST2.SIT, binary, 89600 bytes

This version of the Macintosh PACSAT ground station software now supports extracting PFH and BODY to permit you to view the file without saving.

DX Cluster Monitor Program, file name: DXCMON.ZIP, binary, 65086 bytes

This IBM PC program monitors a DX Cluster node and displays DX spots on your monitor without connecting to a node.

G1YYH V1.9 AmigaNOS Extras and LS Program, file name: AMIGAN.LZH, binary, 20352 bytes

This lzh file contains the missing program LS for G1YYH AmigaNOS V1.9.

G8BPQ Node Software V4.02, file name: BPQ402.EXE, binary, 111293 bytes

This is the latest version of the G8BPQ node software for the IBM PC.

HIGHTIME.PG Hex Date Calculator, file name: HIGHTI.ARC, binary, 12519 bytes

This program calculates the hex number to be placed in your HIGHTIME.PG IBM PC PACSAT ground station software file for use with the Microsats and UO-14.

HIGHTIME.PG Modifier, file name: HIGHFI.ZIP, binary, 5617 bytes

This program changes your HIGHTIME.PG IBM PC PACSAT ground station software file to the current date or a modified date.

Mac TLM PACSAT KISS File Decode, file name: KISSTL.SIT, binary, 30976 bytes,

This version of the Macintosh PACSAT TLM program uses saved KISS text files and includes satellite time with each record.

MSYS Version 1.10 Files 1 and 2, file name: MSY110.EXE and MSYOPT.EXE, binary, 357376 and 164864 bytes respectively

MSYS version 1.10 (an IBM PC PBBS) by Mike Pechura, WA8BXN.

Packet-GOLD 1.22 Test Drive and Documentation, file name: GOLDTD.EXE, binary, 180003 bytes

This is a "test drive" of Packet-GOLD, an IBM PC digital communications terminal emulator that now supports the latest AEA mail-drop firmware, conference mode, binary file transfers, brag files, large scroll back buffers, continuous monitoring, DOS Shell, 43/50 line EGA/VGA modes and user selectable colors.

Packet Radio TNC Driver, file name: TNCV13.ZIP, binary, 235520 bytes

A packet-radio TNC driver for the IBM PC that includes split or full screen operation, a mini-PBBS, script language, YAPP/XMODEM/YMODEM file transfers, UUENCODE, UUDECODE, FAX, SSTV and RTTY receive, connect directory, directory browser, message editor, conversation monitor and more.

PFHADD.EXE v910124m, file name: PFH124.ZIP, binary, 18314 bytes

This IBM PC program adds a PACSAT header to messages before they are uploaded to a satellite. This version adds switches that aid terrestrial PBBS operators.

PG.EXE, file name: PG0206.ZIP, binary, 64382 bytes

This version of the IBM PC PACSAT ground station software includes directory maintenance, selective downloading and listing capabilities and other new features for interacting with the MicroSats and UO-14.

SuperQK Version 0.0a, file name: SQK00A.ZIP, binary, 58267 bytes,

This is a beta version of an IBM PC program that replaces the QK program provided with Quiktrak Version 4.0. It clears, loads, sorts and updates the Quiktrak data files and allows entries to be printed or sent in text, AMSAT and NASA formats.

TexNet Applications Version 1.60, file name: TEXAPP.ZIP, binary, 81280 bytes

Version 1.60 has many new features, changes, and corrections. Two generic image editors are provided for customizing the generic image for new nodes and creating the associated firecode image.

TrakSat V2.30, file name: TRAK23.EXE, binary, 310260 bytes

An IBM PC satellite tracking program that provides color graphics with output in text format or by means of a graphical representation of the satellite's location on a map of the world. The program can tracks up to six satellites simultaneously.

VE4UB NTS Utility, file name: NTSVE4.EXE, binary, 105911 bytes

The VE4UB NTS packet-radio traffic generation program for the IBM PC correctly generates and prepares NTS traffic for sending via packet radio.

HOME IS WHERE YOUR TNC IS...

As a frequent business and pleasure traveler based in Silicon Valley and now in Palm Beach, Florida, I felt "disconnected" when I couldn't check into my regular PBBS, have keyboard-to-keyboard QSOs with friends, or just simply play. I always carried a Model 100 computer and used its convenient, yet slow built-in modem for communicating with my company. Also, I managed to occasionally drag an HK-21 and an HT to explore the local traffic in new areas. I sent packet-radio messages through the network, however, I usually arrived back home long before the message propagated through...often in time to tell the addressee that a message was on the way.

To solve this self-imposed dilemma, a telephone linked system was devised. The basic solution was to tie an auto-answer modem back-to-back with the TNC in the shack. I could then dial the auxiliary phone number at home from anywhere and use the packet-radio system as if I was actually there. The technique afforded more freedom and spontaneity and was a lot of fun. As the system was being assembled and configured, various parameters and conditions were found that tended to optimize the system, as well as its use and safety. I'd like to share these findings in the following notes.

First of all, a 300-baud auto-answer modem was chosen. This speed was adequate for most of the anticipated exchanges and provided a reliable link. At 1200 bauds, more lost data was experienced mostly due to lack of error correction between the modem and TNC. The slower speed was an acceptable trade-off for this simplified system.

Configuring the Modem

Use a terminal or computer to set-up the modem.

Set the auto-answer parameter(s). Eight (8) rings were chosen to discourage all or most erroneous calls.

If possible, disable any "RING" or "INCOMING" annunciators that the modem may send when detecting an incoming call.

Disable any echo features.

Just before exiting the modem set-up session, make sure that the serial communication parameters match those of the TNC.

Configuring the TNC

Using a terminal or computer, set the parameters in the TNC to the desired "default" values. The following settings have been found to be useful in averting potential problems during non-linked periods:

CONOK OFF avoids reverse access to your phone line from the radio side when using a modem with auto-dial features.

BEACON 0

ECHO OFF helps to avoid choking the serial ports with excess characters.

MSTAMP ON logs the most recent MHEARD stations.

MONITOR OFF helps to keep the modem from getting choked, especially with control sequences from network node interchanges.

Connecting the TNC/Modem

Connect the TNC and modem with a "null modem" connection between the serial ports since both devices are typically "DCE." Connect your phone line and radio and your ready to go.

After you call your TNC through the modem, you can set CONOK ON and ECHO ON for normal operation. Setting MONITOR ON is not always recommended when occupying busy packet-radio channels. At 300 baud, buffers take a while to clear. At this point, you will be controlling your TNC as if you were sitting in your shack. You may want to incorporate battery back-up operation in both TNC and modem to avoid problems should power be interrupted. When exiting a session, set MONITOR OFF, BEACON 0, then ECHO OFF. Just before hanging up, send a <CTRL-C> and carriage return to make sure the TNC is left in the command mode.

There are other techniques and features that can be incorporated; it is left up to you to custom design your own system. You may also find other parameters relative to your particular TNC that require attention.

The system has been used as described with a great deal of pleasure and a minimum of problems. It was even used in emergency training exercises during the hurricane season in Florida this year. I had an opportunity to experiment with the system while on a speaking engagement in London two years ago and was successful. Not bad DX, mate!

A data base is being compiled that lists operating notes and parameters for various systems. If you'd like to share ideas, enhancements, etc, I'd like to hear from you.

—by Marcello Soliven, KJ6QA@N4JOA.#WPBFL.FL, from NCPA Downlink

CAPRA INFORMATION PACKET AVAILABLE

Chicago Area Packet Radio Association (CAPRA) has compiled information about packet-radio operations into an "INFO Pack," which includes operating tips, user guides for such things as NET/ROM, KA-Node, DX Packet Cluster, etc. It also has material about operating modes such as AMTOR and HF packet radio. An excellent packet-radio operating tutorial is provided. There are local listings of resources such as digipeaters, NET/ROM nodes, PBBSs and maps for the network in Metropolitan Chicago, call area 9 and the West Coast.

The INFO Pack has almost 200 pages of information and costs $7.50. If you need further information send an SASE to CAPRA, PO Box 8251, Rolling Meadows, IL 60008.

—from CompuServe's HamNet

***GATEWAY* CONTRIBUTIONS**

Submissions for publication in *Gateway* are welcome. You may submit material via the US mail to 75 Kreger Dr, Wolcott, CT 06716, or electronically, via CompuServe to user ID 70645,247, or via Internet to horzepa@gdc.portal.com. Via telephone, your editor can be reached on evenings and weekends at 203-879-1348 and he can switch a modem on line to receive text at 300, 1200 or 2400 bit/s. (Personal messages may be sent to your *Gateway* editor via packet radio to WA1LOU@N1DCS or IP address 44.88.0.14.)

The deadline for each installment of *Gateway* is the tenth day of the month preceding the issue date of *QEX*.