

# VK1CAE Experimental FlexDigi Node

I have installed a FlexDigi node for experimentation in VHF multi user access to an AX-25 node.








This page is intended to give readers an understanding of how to use the FlexDigi node and an overview of how it works.

Detailed configuration information is also given so that you could replicate the whole node, or parts of it.

I welcome feedback on the content of the document, or the node operation and performance, and suggestions for improvement. You can also [download](#) this document in Acrobat PDF format.

I have setup a news group on a local news server as a venue for shared discussion, <news://news.oda.com.au/od.comms.radio> , or you can email me at [owen@oda.com.au](mailto:owen@oda.com.au) .

Links to other resources (software, etc) are [here](#).

-  [Overview](#)
-  [FlexDigi Routing](#)
-  [Overview of FlexDigi Commands](#)
-  [DAMA](#)
-  [Connect Examples](#)
-  [Node configuration](#)
-  [SSID use plans](#)



## Overview

### Purpose of the Flexnet trial

The VK1CAE Experimental FlexDigi node has been established to trial some newer node / routing / channel access techniques.






### Flexnet

Flexnet is a routing protocol that was developed in Europe to overcome some of the perceived shortcomings of NET/ROM. Flexnet was first implemented on the RMNC node, a dedicated multiprocessor microcomputer based node.

Flexnet is firstly a routing protocol, not a software platform.

### Flexnet features

Flexnet has a number of features that are designed to overcome the problems that we experience every day in multiple access channels on VHF packet:

-  hop to hop recovery of lost / damaged frames
-  simple specification of the digi route
-  automatic adaptive routing
-  improved adaptive channel access method
-  support for Demand Assigned Multiple Access (DAMA) channel access method

### PC/Flexnet

Flexnet was later implemented on the IBM/PC platform as PC/Flexnet which is suitable for use in 'end-stations'. PC/Flexnet can run under Windows, and the latest version has improved integration with Win95 (or Win98). PC Flexnet still uses real mode drivers which are loaded prior to starting Windows, and which can use 150K to 250K of 'below the line' memory prior to Windows starting... so will be unattractive to some.

You can take advantage of many of the benefits of a Flexnet network without running PC/Flexnet on your own

machine, Flexnet was in use for years in RMNC nodes before it was available on PCs.

## FlexDigi

FlexDigi is the 'node' module of PC/Flexnet. FlexDigi uses additional memory and will usually be run on a machine dedicated to the node function.

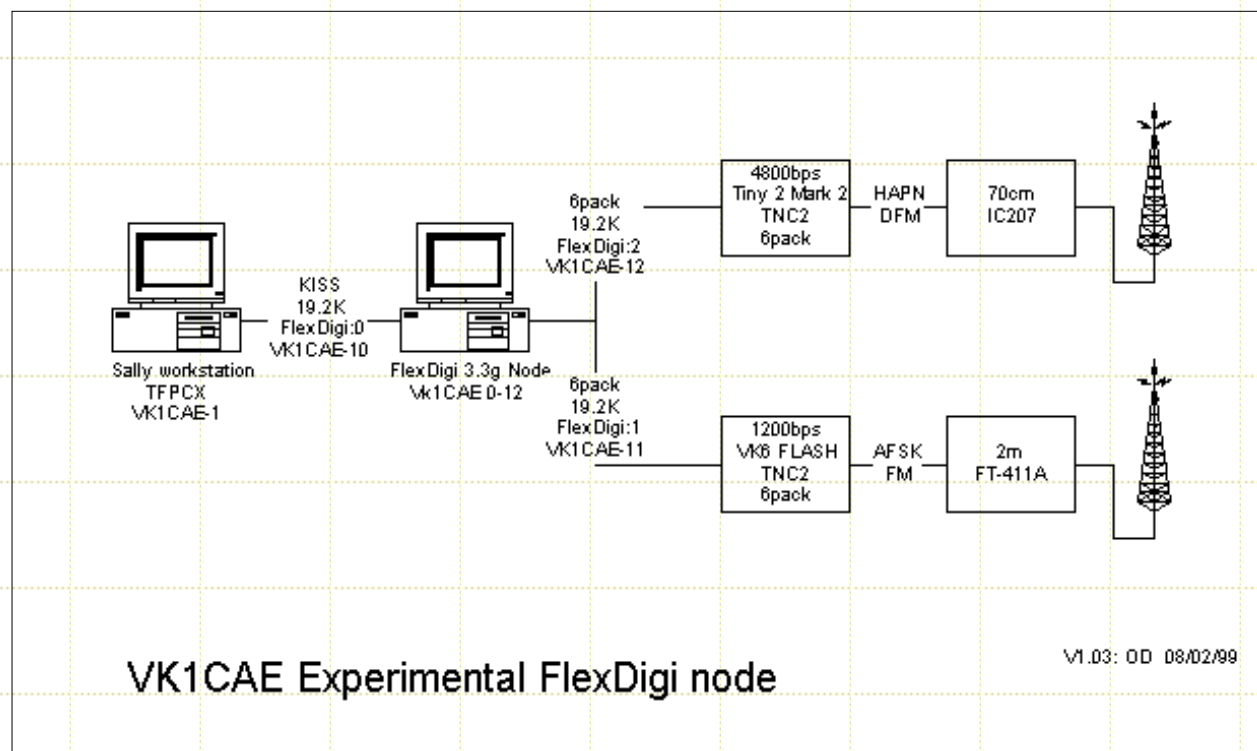


Figure 1: Block diagram of the node configuration. Key elements of the system are described in sections of this document.

## FlexDigi Routing

Using a FlexNet (FlexDigi) node is very similar to using a traditional digipeater system and can be used in exactly the same way.

However, one of the main features of FlexDigi is its autorouter, and adaptive router that can automatically route packets without you needing to specify all the FlexNet nodes along the way, you only need to specify the first and last FlexNet node and FlexNet will automatically expand the address to include the additional transit nodes. The autorouter will dynamically update its routing tables and choose the best route for current conditions.

Not every digipeater in between needs to be named, only the entry and the destination digipeater must be known. There are 4 methods to route a frame to its destination. They are tried in the following order:

1. destination table routing
2. link table routing
3. Heardlist routing
4. SSID routing

### Destination Table Routing

The first method is based upon the callsign information. The digi tries to look up the destination callsign in a table maintained by the autorouter. If the callsign is found in this list, the frame is sent to the corresponding neighbour.

Example: DB0ODW has the following interlinks:

1. DB0KT
2. DB0AAC
3. DB0IE

and knows the destinations: DB0EQ, DB0ZDF etc.

The frame:

```
<fm DG3FBL to DK7WJ via DB0ODW, DB0ZDF>
```

gets expanded to:

```
<fm DG3FBL to DK7WJ via DB0ODW*,DB0AAC,DB0ZDF>
```

DB0AAC is now the next call in the digipeater field, and is known as a neighbour of DB0ODW on port 2. So the frame is sent on port 2 to DB0AAC which will then route it to DB0ZDF.

You can display the Destination Table on a node by connecting to the node and using the D \*, D , or D <call> commands.

## Link Table Routing

When the router does not find a matching entry in the destination table, the digi now tries to look up the call in the link table as set up by the sysop. If a route is found here, the frame is sent to the identified port.

Example: DB0ODW has the following interlinks:

1. DB0KT
2. DB0AAC
3. DB0IE
4. DB0AIS #

The frame <fm DG3FBL to DK7WJ via DB0ODW,DB0AIS> is transmitted on port 4, although there is no entry in the destination table. In this case, only the link table is used for routing.

You can display the Link Table on a node by connecting to the node and using the L command.

## Heardlist routing

The node remembers the last 200 stations heard in an internal list. The SSID of the station is taken care of, if possible. So it is possible to be standby on different ports with different SSIDs without any problems; you will always receive your frames on the right port. Incoming connection wishes and UNPROTOs are routed according to this list.

You can display the Heardlist on a node by connecting to the node and using the MH, MH <port number>, MH <call + wildcards> commands.

## SSID routing

The last chance to get the frame delivered is the SSID specified for the digipeater. The sysop should have assigned SSIDs to user ports. To make use of the SSID routing, the user just specifies the SSID of the port where he wishes his frame to be transmitted.

Example:

DB0ODW has the following mapping of the SSIDs to the port numbers:

port 1 has the SSID -0, port 5 the SSID -12, port 6 the SSID -3

When there is a connect request received which specifies a SSID for the node, the frame is sent on the port specified by the SSID:

```
<fm DG3FBL to DK7WJ via DB0ODW-12>
```

This frame is transmitted on port 5, which owns the SSID -12, provided there is no other way known to DK7WJ. On port 5 the following frame appears:

```
<fm DG3FBL to DK7WJ via DB0ODW-3*>
```

This ensures that someone on port 5 knows where the frame came from, i.e. the entry-SSID is put into the frame. This SSID change is needed to ensure that the receiver knows where to send his UA to, i.e. port 6 with the SSID-3. This is an important feature of FlexNet. All paths are reversible, i.e. transparent to the receiver, even if the connection came by connecting further on at a node. This routing method is used mainly on user ports.

When all these routing methods fail, the frame is discarded. If the frame was created by an "Connect" command at

the node, the user gets the message: "\*\*\*\* <call>: can't route"

You can display the SSIDs allocated to ports by connecting to the node and using the L commands, the SSID is in the "id" column.

## Connect Examples

### To connect from 144.700 to VK1BBS

C VK1BBS v VK1CAE

The autorouter will look after the details (and should route your packets at 4800 on 439.550 to Ginini, across to 144.8 on Ginini and 4800 down to VK1BBS (which is quite quick, quicker than uplinking on 144.800).

### To connect from 144.700 to VK1XYZ on 144.800

C VK1XYZ v VK1CAE GINI48

The autorouter will look after the details (and should route your packets via 4800 on 439.550 to Ginini, across to 144.8 on Ginini and down to VK1XYZ.

### To connect from 144.700 to VK1ABC explicitly on 144.700

C VK1XYZ v VK1CAE-11

The autorouter will look after the details (and should route your packets via 1200 144.700 (which is VK1CAE-11) to VK1ABC.

### To connect to VK1XYZ

C VK1XYZ v VK1CAE

Without an SSID, the autorouter will try to route the packet using its destination table, then link table, then the MH list. If the station you are attempting to connect to is not in the MH list, you will not connect, you will need to explicitly route the outgoing packet using the SSID as in the previous example. The MH list will contain stations that the node hears on 144.700 and 439.550 (but not 144.8).

So, if a station has been heard by VK1CAE, the autorouter will look after the details (and should route your packets via the port on which the station was last heard.

If the destination callsign specifies (ie non 0) SSID, the router will take that into consideration.

For information on usage, connect and use the LO, I, and A and H commands. You can have a look at the destination tables and link tables on the node and work out your own variations.

Try these examples and note the expansion of the address lines if you can hear the packets.

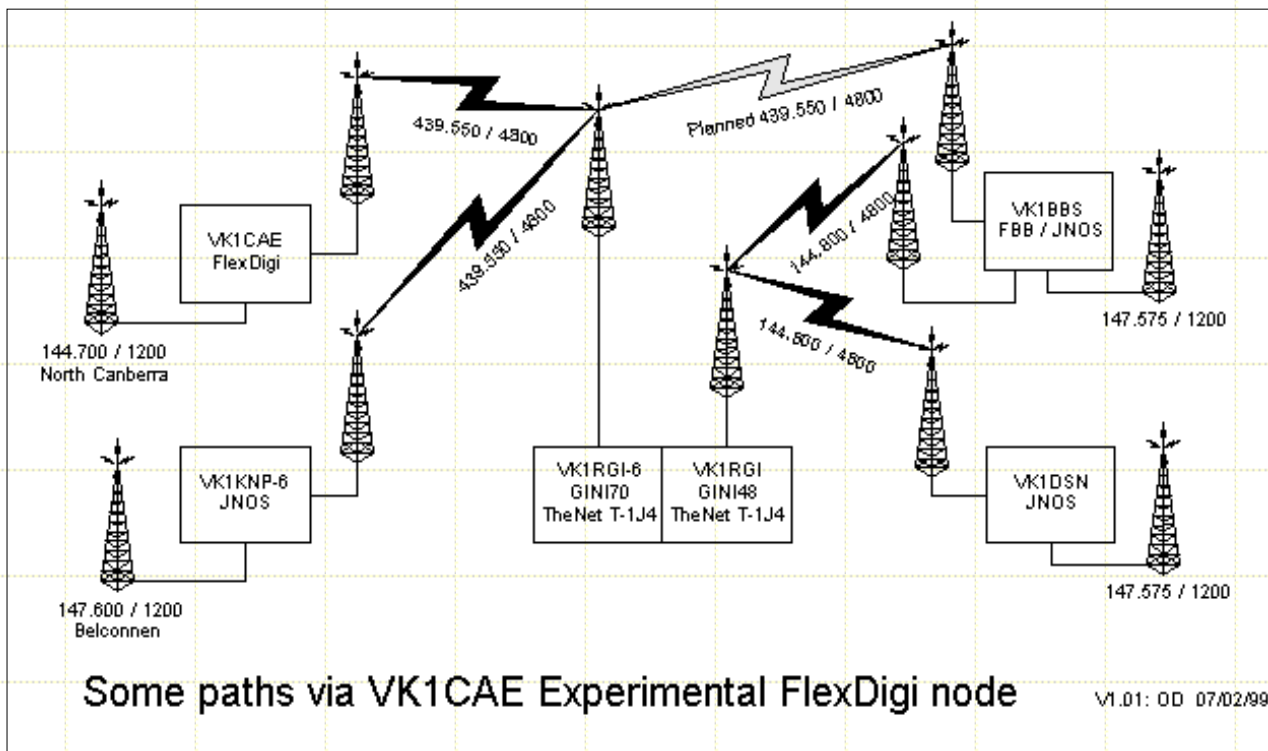


Figure 2: Some of the paths reachable via the experimental FlexDigi node.

## Overview of FlexDigi commands

This is a summary of the commands that the FlexDigi host accepts. (You do not need to logon to the FlexDigi node to route via it.)

- A - display text LATEST NEWS
- B - display beacon-file
- C - start conversation mode
- /w - list node users
- /w n - list convers users on channel n
- /c - display convers channel number
- /c n - switch to channel n
- /s call text - send private msg to user
- /q - quit conversation mode
- C call [v] [digi] - connect further on
- D [\*] [call] - display destination table, path to destination
- F <call> - FIND-command, look for <call>
- H - display text HELP
- I - display text INFO
- IO - In/Out - status
- L - display Interlink information
- LO - display text LOCAL
- M [?] - connect next BBS
- MH [...] - display Heard-list [selectively]
- MY - display Mycall and SSID-range
- P - display parameters and statistics
- Q - quit, end of connection
- S - SETSEARCH, display search-paths
- T <call> <text> - send text to other users
- U [\*] [port/"="] - display user table [selectively]

## DAMA

Demand Assigned Multiple Access is an alternative channel access method that is designed to overcome some of the problems in use of the commonly used Carrier Sense Multiple Access (CSMA) on multiple access channels.

CSMA suffers from serious throughput degradation at even moderate levels of channel utilisation (>30%) even when all stations can hear each other. The deadtime (between when your TNC (or equivalent) decides the channel

is clear and another TNC has detected data carrier is a contributor to collisions which waste channel capacity. This is then seriously aggravated when one station cannot even hear another station that is transmitting and collides, again wasting channel capacity. The CSMA algorithm falls down seriously when all participating stations cannot hear each other.

DAMA is intended to give all stations in session with the node fair access to node bandwidth, whether they are weak or strong. The FlexDigi node will be a DAMA master (when enabled) and will poll the stations its session partners in turn. DAMA capable session partners will not transmit until they are polled, thereby reducing the collision rate. This is effective even where two of the nodes session partners cannot hear each other, because they don't transmit until told so by the node.

One advantage of the DAMA method is that it does not require everybody to change everything all at once. However as additional users convert their TNCs to work with DAMA the more pronounced will become the increase of throughput. Even stations that are waiting to switch over could help to increase the areas throughput by changing a two operational parameters:

1. the delay between the reception of a frame and the TNCs response (sometimes called T2, TXDELAY or DWAIT) should be reduced to a value under 1 second;
2. the time interval from when an I-frame is sent to when the TNC sends an RR# to ask for a pending ACK (sometimes call RESPTIME), should be set to a value that is clearly higher than the time between two polls of the master (usually more than 30 seconds at 1200 bps).

To fully benefit from DAMA both the node and the user must work together in the master/slave relationship.

While a DAMA slave is connected to a DAMA master, it will only transmit when polled by the master, and it will transmit irrespective of DCD. This does not prevent the slave station having additional connections to other stations, just the slave will defer transmitting frames for all stations until polled by the master. If a station is not connected to the master, it is not controlled in any way by the master. For these reasons:

- ✚ only one DAMA master should be active on the channel;
- ✚ the channel should be reserved for connections to the node or via the node;
- ✚ stations should be DAMA compliant, or change their channel access parameters (as above) for best effect; and
- ✚ stations should disconnect from the DAMA master before changing frequency to ensure that their DAMA slave status is reset properly (rather than depending on the longish timeout).

The full DAMA channel access method is implemented in current versions of :

- ✚ PC/Flexnet
- ✚ TheFirmware TF2.7b) (WA8DED Hostmode for TNC2) as a DAMA SLAVE by NORD><LINK
- ✚ TFPCX (2.71\*\*\*), a resident AX.25 software TNC for the PC (only external Modem required, no TNC);
- ✚ TFKISS (3.01) (former TFPCR) which emulates a TNC with TheFirmware and any TNC can be used if it can be switched into KISS-Mode
- ✚ DIGICOM and BAYCOM Software
- ✚ TheNetNode (TheNet for PC) as a DAMA MASTER

(\*\*\* bugs detected in DAMA implementation, updated patches available from me.)

## Node Configuration

The Node workstation is an IBM PC compatible 486 DX33 Motherboard, 4MB RAM, 1.44MB FDD, 2 x 16550 COM ports.

The machine boots from a floppy disk and creates a ramdisk image of the files required to run the node, then changes to the ramdisk and starts the node.

### CONFIG.SYS

```
device=a:\dos\himem.sys
devicehigh=a:\dos\ramdrive.sys /E 2000 512 128
```

### AUTOEXEC.BAT

```
@ECHO OFF
set tz=EST-10
```

```

rem set tz=EST-11EDT
path=a:\bin;a:\dos

rem find the ramdrive
set FINDVOL=.
findvol -e -z
if errorlevel 1 goto nord
set TEMP=%FINDVOL%\
set TMP=%TEMP%
copy command.com %FINDVOL%\
set comspec=%FINDVOL%\command.com
xcopy /e a:\ %FINDVOL%\
path=%FINDVOL%\bin;%FINDVOL%\dos;%FINDVOL%\flexnet
doskey
%FINDVOL%:
set RAMD=%FINDVOL%
mblank 9
m n t
goto end

:nord
echo Cannot find RAMDRIVE
echo .
set
:end

```

## M.BAT

```

@ECHO OFF
:m
cd \
rem deal with arguments for auto execution of options
set cmd=%1
shift
menu /c%cmd% /t1 /a30 QqUuNnTtZzSs \bin\m.scr
if errorlevel 13 goto quit
if errorlevel 11 goto 11
if errorlevel 9 goto 9
if errorlevel 7 goto 7
if errorlevel 5 goto 5
if errorlevel 3 goto 3
if errorlevel 1 goto 1
if errorlevel 0 goto quit
rem *****
:11
rem Sync directories
if not "%RAMD%" == "" update a:\bin %RAMD%\bin
if not "%RAMD%" == "" update a:\flexnet %RAMD%\flexnet
goto m
rem *****
:9
rem unload Flex
FLEX /u
cd \flexnet
if not "%RAMD%" == "" update a:\flexnet %RAMD%\flexnet
boot
rem should never get here
pause
goto quit
rem *****
:7
rem Baycom terminal
bct /c /r0
goto m
rem *** help sub menu *****
:5
cd \flexnet
FLEXNET 150
if errorlevel 1 goto flexu
rem Port 0:
kiss 1
if errorlevel 1 goto flexe
rem Port 1,2:
6pack 2 /b=19200 /c=12
if errorlevel 1 goto flexu
flexdigi
if errorlevel 1 goto flexu

```

```

FLEX
rem Port 0:
FSET mode 0 19200ydc
FSET txd 0 0
rem Port 1:
FSET mode 1 1200cm
FSET txd 1 25
rem Port 2:
FSET mode 2 4800c
FSET txd 2 25
goto flexe
:flexu
rem something went wrong, so unload the lot
FLEX /u
:flexe
rem need to invoke the menu again
m %1 %2 %3 %4 %5 %6 %7 %8 %9
rem *****
:3
FLEX /u
cd \flexnet
if not "%RAMD%" == "" update a:\flexnet %RAMD%:\flexnet
rem need to invoke the menu again
m %1 %2 %3 %4 %5 %6 %7 %8 %9
rem *****
:1
:quit

```

## Ports

The following is a list of the port parameters from the p command:

po	id	td	qso	usr	tifr	rifr	tkby	rkby	qty	mode	links	ssids	time	
0	10	0	3	1	62	34	8	0	100	19200ydc	VK1CAE	1-1	2	-
1	11	40	1	2	15	17	0	1	100	1200cm				
2	12	40	7	2	2	2	0	0	100	4800c	VK1RGI	6-6	21	@
2											GINI70	0-15	20	@
15	--	0	0	0	3	1	0	0	100	y				
										via VK1RGI-6	VK1RGI	0-15	21	@
										via VK1RGI-6	GINI48	0-15	21	@
										via VK1RGI	VK1DSN			
										via VK1RGI	VK1BBS	0-15	44	@

Note: this was a captured at time of writing and may have changed since, if you are able, log onto the node and get a current list.

## Links

The following is a list of the links from the l command:

VK1RGI	6-6	21	P 2 @
GINI70	0-15	20	P 2 @
VK1RGI	0-15	21	via VK1RGI-6 @
GINI48	0-15	21	via VK1RGI-6 @
VK1DSN			via VK1RGI
VK1BBS	0-15	94	via VK1RGI @
VK1CAE	1-1	2	P 0 -

Note: this was a captured at time of writing and may have changed since, if you are able, log onto the node and get a current list.

## SSID plans

Inbound SSID plan		
Callsign	Destination	Comment
VK1CAE-0	FlexDigi node host	

VK1CAE-1	Sally PMS (if it is running)	

Explicit SSID routing		
SSID	Link	Comment
10	Sally	
11	144.700 1200bps AFSK	
12	439.550 4800bps HAPN	

Outbound SSID plan		
Callsign	Destination	Comment
0	FlexDigi	
1 - 4	Sally PMS	
9	FlexDigi local terminal	
10 - 13	Flexdigi link SSID	
14 - 15	Reserved	for connecting back in from a Net/Rom node for testing etc

## Change history

Revision	Date	Comment
1.00	08/02/1999	Original release
1.01	09/02/1999	Remove ZX, AG from link table, fixed printing
1.02	26/02/1999	Transposed ports 0 and 1 for ease of testing
1.03	01/03/1999	Changed frequency of Port 1

1.04

13/06/1999

Updated AUTOEXEC.BAT to use FINDVOL

# Intellectual property

This document may use trademarks in referring to some objects. All product names and trademarks are the property of their respective owners.

This work is copyright Owen Duffy 1999, all rights reserved.

Updated: 13/06/99 17:31:05 +1100, Hit count: **00136**

---

Can't find it? Try my [search](#) facility.

I appreciate your comments on this web, and advice of problems that you may have encountered. Try the [Feedback](#) form or email the [webmaster](#).

All content is Copyright: Owen Duffy 1995,1998. All rights reserved. No material may be copied or linked without express written permission.

---