

R M N C / F l e x N e t

and

P C / F l e x N e t

Sysop Manual Software

June 1995

Gunter Jost, DK7WJ

translated by Mario Lorenz, DG0JAB

This manual was written with great care. However, it may contain errors. Therefore, there is no warranty of any kind nor any other legal responsibility of the author or the translator.

Copyright (C) 1995 Gunter Jost, DK7WJ, Lichtenbergstr. 77, D-64289

All rights reserved. No part of this manual may be reproduced or duplicated without prior written consent of the author.

=====

1. News on Version 3.3

=====

1.1. Channel Synchronization

It is now possible to synchronize several channels by software. The DCDs of these channels are checked mutually, and only one channel may send at one time. The timing parameters are optimized for multimode user access ports and cannot be changed at the moment. Activation is done by an additional "s" in the MODE-command. All ports with the "s"-flag set are barred against each other. A hardware DCD conjunction may, but does not need to be removed. It is not possible to define more than one group of synchronized channels.

1.2. DAMA-Master

The node now is capable of the DAMA protocol, currently only simplex. On synchronized channels, the DAMA mode is synchron, too. The transmission time per QSO is currently limited to about. 4 seconds; parameters cannot be specified currently. DAMA mode may be activated by an "m" in the MODE-Command. It is possible to have several independent DAMA masters at the same time, e.g. for user ports on different frequencies. On multimode ports, the combination "sm" has to be used (not on RMNC fullmaster).

1.3. DAMA-Slave

When a channel hears another DAMA master, e.g. when using the system as user frontend, this channel is set to DAMA slave mode automatically. For network nodes this feature can be turned off.

1.4. Local C-Text

An additional C-Text can be set up for users which connect locally, i.e. without a digipeater in their path. This text can be written by using the command "WRITE L". Any user can read it with the help of "LOCAL". It is useful to move special hints, for example about the local user ports, away from the C-Text to the L-Text, if they are not of any interest when accessing the node via interlinks.

1.5. New Link Options

Subnet links (">" - links) can be hidden now. The ">" must be replaced by ")" to achieve this.

1.6. New Trace Options

```
#      suppression of RR/RNR/REJ-Frames
$      suppression of I- and UI-Texts, i.e. only the frameheaders are
      displayed
<call> trace only this callsign. Only check source and destination
      callsigns, digipeater callsigns are not checked. SSID is taken
      care of, if specified.
>      only frames sent
<      only frames received
```

The port number still needs to be defined.

1.7. Talk Command

With this command a line of text can be sent to another user who is connected to the infobox. The output is similar to the "/s"-command in convers mode. This command is not implemented on RMNC fullmasters.

Syntax: "TALK <call> <text>"

Instead of typing the TALK-command for each line, a durable TALK mode can be activated by "TALK <CALL>"; it is ended by giving "/q". This is similar to convers mode.

1.8. TxDelay Measurement

On any port TxDelay measurement of the other stations can be activated. If the measured TxDelay exceeds the needed TxDelay more than 100ms, the connection is interrupted. For stations without a digi in their path, a warning message is issued.

1.9. New MODE-command Parameters

- Turn off port
y On this port, you will always be sysop as long as there are no digipeaters in your path. For security reasons, on RMNCs this may only be set in the MRMNC compiler. On these ports the loop detector is inactive.
ATTENTION: This attribute should only be set on local(wired) links. Even here, it must be impossible to connect back to the node, otherwise everything can be reprogrammed freely.
C Only relevant in KISS mode: Force CRC mode.
On some PC/Flexnet drivers: Activation of software DCD.
M DAMA master (see separate discussion)
S Channel synchronization (see separate discussion)
U This port is a user port. Activates following features (on RMNC fullmaster only partially implemented):
* Port never goes into DAMA-Slavemode
* TxDelay-Measurement (see separate discussion)
* If in DAMA mode: Disconnect of users who poll the digi several times (i.e. they are no proper DAMA slaves)
If a connection is interrupted due to one of the conditions above, a warning message is issued, but only if the user is connected directly, i.e. without any digipeaters in the path.

1.10. New U-Options

= show only Infobox-QSOs (was: any character)
<call> show only QSOs from/to this callsign (solomaster only)
* as usual, display additional QSO data

1.11. New L-Option

With "L *", extended statistics are shown. This includes the uptime of the software and the up- or downtime of each link. Link resets will reset this time only if they are caused locally or the connection is interrupted.

1.12 Heardlist / MH-Command

The Heardlist, up to now only used internally for routing, can be displayed now. All heard callsigns are inserted into the list. This also improves the port routing. The list contains max. 200 callsigns and is saved to HD on PCs every ten minutes.

Options:

<port> Portnumber 0-15, list only callsigns heard on this port
<count> number of callsigns to be listed. Default is 30. <count> must be in the range 16 ... 200.
<call> Look for this callsign. SSID is ignored if not specified.

1.13. Better command line interpreter

The command line interpreter of the infobox now works line oriented instead of frame oriented. This makes uploading of parameters faster and the convers mode has fewer problems formatting the text. Not implemented on RMNC fullmasters.

1.14. Other changes to Version 3.2

* Baudrate no longer defaults to 1200 baud. It now needs to be explicitly set for each port. Therefore, the list file only shows the used ports now.
* The P-command accepts a port number; "P 1" now only lists the line for port 1
* On solomasters, text memory is 7.5KB now
* The autorouter has been modified slightly; routing-changes in the network are broadcasted faster now.

2. INTRODUCTION

2.1. The history of FlexNet

First ideas for the software came back in 1987, and the first version was developed in 1988. First tests were run at home and later at the digipeater DB00DW in Odenwald, Germany. This digi was equipped with a complete 6809-development system and made it possible to test new versions by uploading them. In 1989 the work on the RMNC version began. The RMNC (Rhein-Main-Network-Controller) was developed by the PR-Group of Frankfurt (RMPRG) and offered an optimal platform for FlexNet.

Since 1991 a version for MS-DOS-Computers has existed, but it has been used only internally and never been distributed. However, after a growing demand for such a PC/FlexNet, in 1994 a new modular driver concept was developed in cooperation with DL8MBT, the author of the BAYCOM software.

This makes PC/Flexnet usable for almost everybody. PC/FlexNet is freely configurable and may be used with any I/O-modules. It also offers an application interface for high level applications, such as terminal emulations. Development kits for this interface and the IO-drivers are available.

PC/Flexnet is usable also for end-users which do not need a whole node. User are offered by this an AX25-driver with the advantages of FlexNet, like the very easy parameter setting and the modular driver concept. It runs with almost every terminal program which supports the WA8DED Hostmode. (See File TFEMU.DOC for details) In this case, FLEXDIGI.EXE is not loaded, and therefore no work needs to be wasted in setting it up.

At this point we wish to thank everyone who helped us in developing the software. Our special thanks to all the sysops, who patiently tested the software again and again to get it stable. The goal of FlexNet was and is to develop a robust and easy-to-maintain node software, which should annoy sysops and users as little as possible.

2.2. Important features of FlexNet

RMNC/FlexNet V2 was a version which had many differences to previous versions for users and sysops, both operational & optical. The RMNC/FlexNet V3.0 had some new features, where the autorouter was the most important one. V3.2 was completely revised and became faster and more comfortable again. V3.3 brought the DAMA channel access method and is forerunner of a completely new access method which is also usable on echo duplex nodes. Maintenance of the RMNC was even more simplified, the change of the whole software of the node can now be done by changing one single EPROM. All L2- and L1-parameters except TxDelay are set adaptively to the current working conditions; this means that the sysop has almost no work in setting up parameters. We found it unnecessary to have parameters for every special cases. It was reached the point where the node runs without any maintenance, except for dropouts, of course. Some statistics are useful for the sysop in optimizing the interlinks and make the network transparent to the user. The permanent development and the very cheap hardware made RMNC/FlexNet to the most popular system within Germany, and the growth rates in foreign countries are impressive.

From now on FlexNet for MS-DOS is available. The RMNC is not intended to be replaced, it is still the preferred platform for unattended nodes. However, it is an alternative for existing PC-based nodes and makes it possible to experiment with FlexNet without prior high investments in hardware.

2.2.1. Hop-to-Hop-Acknowledgement

Since V2 an important feature has been the immediate acknowledgement of received frames of QSOs which run via the digi. This greatly improves the stability of the links. The possibility of a failure used to be increased exponentially to the number of digipeaters in the path. Now the whole link is as good as the worst interlink on the way.

2.2.2. Connectability

Since V2 the RMNC-digi has been connectable, i.e. QSOs are now possible with the digipeater itself (previously it was only a L2-digi). The digi offers several service functions, for instance convers mode, link information and some information pages.

2.2.3 Remote Control

The FlexNet-software makes it possible to enter sysop mode of the digipeater remotely. This allows changing of parameters, information pages, link information and so on after successful authorization as sysop.

2.2.4. Installation

Updating to a new RMNC/FlexNet-software version is easy, just one EPROM needs to be programmed and changed. To simplify the configuration of the node, an IBM-PC(and compatible) based program is available for doing this job. Please refer to the chapters "Installing RMNC/FlexNet and "Generating a MASTER EPROM".

The software is available at no costs for use in amateur radio and may be copied freely in binary form (Disk/EPROM) as long as you accept the legal terms. (see section 6)

3. Using FlexNet

=====

Using a FlexNet node is very similar to using any other digipeater system. However, you have to specify the callsign of the entry and the destination node. The hop-to-hop-acknowledgement which has been used since V2 should now be known to the users. The FlexNet auto-router should not cause problems anymore also. Surprises can occur from the loop detector only. It displays the message "loop detected" if the user tries to connect a node which would be routed back to the same port the user comes from.

3.1. Adaptive Parameter Settings

Most parameters which had to be set manually on traditional node systems are set automatically by FlexNet according to the actual conditions. After many experiments, RETRIES were set to 10, but this caused QSOs on fast links to break on short interrupts. To fix this, the node polls at least 90 seconds before giving up. The number of polls, therefore, indirectly depends on the data rate and the use of the channel. FRACK (Frame-Acknowledge-Time, T1) is permanently adjusted to the time the other station needs to acknowledge the I-Frames. MAXFRAME is set according to the reliability of the connection. When all frames are acknowledged reliably, MAXFRAME is quickly set to 7.

On 1200Bd, however, 7 frames are sent relatively rarely, since the maximum time of one transmission is restricted to 12 seconds. All QSOs are handled equally, i.e. they share the 12 seconds transmission time. If one station needs a retransmission of frames (e.g. by REJects), MAXFRAME is decreased instantly. In this case it is provable that the throughput is significantly higher when MAXFRAME is set between 1 and 3. Only the quality of the connection TO the other station is checked, not the connection FROM the other station. A poll which gets the acknowledgement of all frames or a lost acknowledgement does not affect MAXFRAME.

PERSISTENCE is the most critical parameter when many users are using the same channel. A fixed parameter setting is always only a bad compromise between collision probability and transmission willingness. FlexNet, therefore, makes use of a new access method, which works fully adaptively. All stations on the channel are permanently counted (column "usr" in the P-list). A waiting time is calculated from this value and from the data rate. The channel needs to be free for at least this time (about 4 secs on 1200Bd and 10 users) to cause FlexNet to think about transmitting. When this time is over, the channel is checked in variable intervals and - if necessary - the node goes "on air". The throughput is drastically increased by this algorithm, especially on duplex nodes. Now aggressive stations do not have the advantage of being serviced faster; there is always enough free time for weak stations. If there are only one or two stations, the channel is almost fully accessed, thus making fast downloads possible on low traffic times. On simplex nodes, of course, many collisions still happen.

But even here, the new algorithm improves things significantly. It would be good if the user software could follow this trend to adapt the parameters to the actual conditions. A computer should be able to do this job better and faster than a human could do, especially since it seems to be common that most users are unable to set their parameters correctly.

3.2. The phases of a connection via the Digipeater

3.2.1. Link Setup

During the connection setup the received SABM-frames are routed normally, the UA is not sent immediately. This means that the connection is established only if the called station is reachable. When the called station answers with an UA, this will be routed to the calling station. At this moment, 2 QSOs are in the user list, one from the calling station to the called station, and one back. A special feature of FlexNet is, that the connection can be established in one single step without disadvantages, the common L2 digipeating is simulated. At home, you can directly "Connect <user> via <entrynode> ,<exit node>" or "Connect <destination node> via <entry node>". You do not need several "Connect" commands to make your way to the destination. As hop-to-hop-acknowledgement is still active, the quality of your QSO is not affected.

3.2.2. Information Transfer

During the information transfer, hop-to-hop-acknowledgement gets activated. Every I-Frame is immediately acknowledged by the digipeater. Now the digipeater tries to route the packet further on to its destination. REJects due to interference or collision only happen at one section, not in the whole route.

3.2.3. Link Failure

If the connection breaks at one side during information transfer, the digipeater interrupts the whole connection and displays the message:
"*** OKONE: link failure". Thus, the partner gets informed that something does not work.

3.2.4 Disconnect

When station "A" disconnects, the digipeater responds immediately with an UA. The digi now tries to abort the connection to "B". When there is not any data for station "B" anymore, this happens at once. If there are some I-frames left, they are sent to "B" before the connection is terminated. When "B" tries to send frames to "A", which is already disconnected, these frames are discarded.

3.3. Methods of Routing

As mentioned above, 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

3.3.1. 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 neighbor.

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 neighbor of DB0ODW on port 2. So the frame is sent on port 2 to DB0AAC which will then route it to DB0ZDF.

3.3.2. 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 right port.

Example: DB00DW has the following interlinks:

```
1:DB0KT
2:DB0AAC
3:DB0IE
4:DB0AIS #
```

The frame <fm DG3FBL to DK7WJ via DB00DW,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.

3.3.3. 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 SSID's without any problems; you will always receive your frames on the right port. Incoming connection wishes and UNPROTOS are routed according to this list.

3.3.4. SSID routing

The last chance to get the frame delivered is the SSID specified for the digipeater. The sysop can assign SSID's to certain ports using the PARMS command. 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:

DB00DW has the following mappings of the SSID's 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 DB00DW-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 DB00DW-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"`

4. Infobox-commands

4.1. User Commands

User commands are all the commands normal users can access. The sysop has a set of additional commands or may specify additional parameters to normal user commands. In this documentation, <CR> means entering of a Carriage Return, \$0D. The "=>" is the system prompt of FlexNet; input is expected now. All input can be made either upper or lower case. Is another command entered than those listed below, the node answers with: "invalid command".

4.1.1. L<A>test News

Syntax: A <CR>

The A-Command shows the text for latest news as set by the sysop. After a cold reboot this text is empty.

4.1.2. eacon

Syntax: B <CR>

The B-Command shows the current beacon file. In this file you can see which beacon is sent on which port in which interval. After a cold reboot the default beacon is sent on port 0 or 1.

4.1.3. <C>onvers mode

Syntax: C <CR>

If no callsign is given, the CONNECT command puts you in convers mode. By this mode, a great number of stations can have a round table conversation. There are 255 different convers channels available. After entering the C-Command, you get a list of all stations connected to the node and, if they are in convers mode, too, the channel on which they are. Now the node prompts for a number, which selects the channel you want to join.

Example:

```
=>C <CR>
users:
0: DL1AA 0:DL1ZZ ---: DL2XY      73: DG3FBL      73: DK7WJ
```

```
channel ? 73 <CR>
*** starting convers, exit: /q
```

In this example, DL1AA and DL1ZZ are on channel no. 0 and DG3FBL and DK7WJ on channel 73. DL2XY is connected to the node without being in convers mode. Having given the desired number 73, the conversation starts. All stations logged in onto the chosen channel get the message:

```
" <DL9ABC>: *** Logon"
```

While being in convers mode you have the following commands at your disposal:

```
"/w"          shows all stations connected to the node (with convers
              channel number if available)
"/c"          shows the actual channel number
"/c n"        switches to channel n
"/s <call> <msg>" sends private msg to <call> only
"/m <call> <msg>" sends private msg to <call> only
"/q"          quits convers mode
```

If a station disconnects while being in convers mode or quits convers mode, all other users of the channel get the message:

```
"<DL9ABC>: *** Logoff".
```

If a user changes to another channel, the users of the left channel get the message:

```
"<DL9ABC>: *** switched to channel n"
```

If there is no channel number entered on convers start-up, convers mode is ended immediately. You are then prompted for a new command.

4.1.4. <C>onnect

```
-----
Syntax: C Call [via] [digi1 digi2 ... digi8] <CR>
```

The CONNECT command is used to connect further onwards. The node will try to connect you to the station via the path you specified. To confirm your command, you get the message "link setup...". As soon as the connection is made, you will get "*** connected to <call>" from the node. When the called station did not respond, you get "*** failure with <call>". If the called station sends a Busy (DM), the message "*** busy from <call>" is sent to you.

The link setup can be interrupted by sending a single <CR> to the node.

If you see the message "*** can't connect twice", you have tried to establish a QSO which already exists with the same call sign fields.

With the C-Command it is also possible to change the user port, if the node has more than one. By typing "C -7" you change to the port with the SSID 7. This is acknowledged by the message "*** <call>: SSID OK".

If you connect to another station from the node onwards, and that station disconnects you, you will get reconnected to the node. To show you what happened, you get a "*** reconnected to <call>" then.

A connect request will be denied, if it causes a loop in the network. If, for example, you are connected to DB0KT via DB0ODW, you cannot connect back to DB0ODW nor to other nodes behind DB0ODW. You should quit the QSO with DB0KT then and retry after the reconnect.

Example: (user is connected to DB0HP)

```
=> C DB0ODW <CR>
link setup...
*** connected to DB0ODW
RMNC/FlexNet V3.3d - DB0ODW - JN49 IQ - Help mit H

=> C DB0HP <CR>
*** DB0ODW: loop detected
=> Q <CR>
73!
*** reconnected to DB0HP
=>
```

4.1.5. <D>estinations

Syntax: D [call] <CR>

The DESTINATIONS command prints out the destination table maintained by the node. In this table all nodes, where the autorouter knows a way to, are shown. For every callsign there are the SSID range of the callsign and the average round trip time in 100 ms steps are shown. As an optional parameter a destination callsign may be given. The node will now try to work out the way to this node and will show it (after some seconds, depending on the (round-trip time). Uppercase callsigns mean that the node knows the FlexNet protocol, lower case callsigns are inserted by the autorouter to reach the next FlexNet node. The characters "???" mean, that the previous digi does not know the way to the destination. This may happen, when the route to the destination is reorganized at the moment or when the destination is not reachable anymore. The "D-Table" is usually the same on all nodes. Only when round trip times get too high, a node is not shown anymore. Only nodes that you can reach without link loops are shown by default. This reduces link load and has the advantage that you will see only the nodes that are not in your direction. By using the option "*", you will get the complete list. Another possibility is the selective display of a part of the list. By entering "D HB9" for example, you get all destinations starting with "HB9", i.e. the whole Swiss network. Both parameters may be used together. If you type "D * HB9" you will get all Swiss destinations, including these you cannot reach without loops.

4.1.6. <F>ind

Syntax: F call <CR>

With the FIND command it is possible to look for a station which is standby on this or another node. When the F-Command and the callsign are entered, the digi sends UI-frames with the POLL-bit set to this station via some neighbor nodes. Source callsign is the callsign of the OM who issued the FIND command. If the called station hears the frame, it will answer with a DM- Frame. The node analyses all frames coming back and is able to determine if this was an answer of the FIND command. If this is the case, you will get a message via which node the station was found. If the called station is already connected to the node, no special frame is sent and the user will get the message that the user is QRV on the digi.

Example:

```
=>F DK7WJ <CR>
*** DK7WJ found via DB0ODW
=>
```

Only the node via which the called station was found is put out. It will be known to the autorouter. If the station was not found, a system prompt "=>" appears again. Since the used UI and DM frames may get lost, it is advisable to use the FIND command more than only once to be sure the user is not QRV. Due to the protocol, the SSID of the called station must be known.

4.1.7. <H>elp

Syntax: H <CR>

The H-Command prints out the text-file HELP. The text can be entered by the sysop only and should give short help text about using the node. After a cold reboot the text is empty.

4.1.8. <I>nfo

Syntax: I <CR>

The I-Command prints out the text-file INFO. This text can be entered by the sysop only and should provide information about the node (QTH, equipment, antennas and so on). After a cold reboot the text is empty.

4.1.9. <IO> (In/Out)

Syntax: IO <CR>

The IO-Command shows the state of the I/O-ports of the RMNC reset card. There are 16 lines in and 16 lines out. The latter may be set only by the sysop. Using this ability it is possible to remote control the node by hardware. There are no limits to the fantasy of the sysop. The data is shown binary.

Example:

```
=>IO<CR>
I: 0000 0000 0000 0000      O: 0000 0000 0000 0000
=>
```

The input lines are shown first and then those of the output. 0 means "low", 1 means "high". The meaning of the single bits needs to be documented by the sysop.

4.1.10. <L>inks

Syntax: L <CR>

The LINKS-Command displays the link table set up by the sysop.
Example:

```
=>L <CR>
DB0KT      0-7      60/68      P1
DB0AAC     0-15     (---)      P2
DB0IE      0-1      583        P3 @
DB0EQ      0-8      (355/399)  via DB0IE
DK7WJ      8-11     44/67      P0 -
DB0ABA
DB0BBS     0-15     ---        P5
```

In the first column the callsigns of the neighbor nodes are shown. Second column shows the SSID ranges of these stations (default: 0-15). In the third column you read the round trip time to the neighbor in 100 ms-steps. No number here means that the round trip time is not calculated.

Three hyphens mean that the link is not available at the moment. Three hyphens within brackets mean that the link is not available but the autorouter is aware of another way to the station. If there is only one number in the column, the link partner does not know about the FlexNet protocol, or the internode QSO could not be established. When the sysop knows that the neighbor does not know the FlexNet protocol, he may set the attribute "@" to the link. Then only the link is tested, not if the partner knows the protocol. If the round trip time is surrounded by brackets, the link is so bad that it is not made known to the network. If there are two numbers, separated by a diagonal stroke, the neighbor is a FlexNet node. In this case the round trip times of both directions are shown. If these values are within brackets, the autorouter knows a better way to the destination, i.e. the direct link is not used. The 4th column shows either the port number of the link to the neighbor (on direct links) or the stations via which the neighbor is reachable. A hyphen behind the port number means that the link is not made known to the network. This may be used for temporary links or software tests for example.

4.1.11. <LO>cal

Syntax: LO <CR>

The LO-Command shows the text-file LOCAL. This text is appended to the CTEXT for local users, but it can be displayed by the LO command separately. The text may only be entered by the sysop. After a cold reboot this text is empty.

4.1.12. <M>ail

Syntax: M <CR>

The MAIL-Command connects you to the nearest BBS as defined by the sysop. This command therefore works like a "Connect" command with predefined destination. The BBS callsign can be shown with "M ?" (notice the space!)

4.1.13. <MH>eard

Syntax: MH [options] <CR>

The MHeard-Command by default displays the last 30 direct heard callsigns. Optionally, a port number, a callsign (with or without SSID) or a number (16 ... 200) of entries to be listed, may be given.

4.1.14. <MY>call

Syntax: MY <CR>

The mycall command gives the callsign and the SSID range of the node.

Example:

```
=> MY <CR>
mycall: DB0ODW, SSID's: 0-7
=>
```

4.1.15. <P>arameters

Syntax: P <CR>

The PARAMETERS command puts out a list of the current parameters and some channel statistics. Additionally, the links as shown with the <L> command are displayed.

Example:

```
=> P <CR>
po id td  qso usr  tifr rifr tkby  rkby  qty   mode      links
1  -- 10   30  1   365  287  50    33  100  9600d*+   DB0KT 0-10   6/6
2  --  1    36  1   271  908  30    163  99  19200d*+  DB0GV 0-0    4
3  --  1     1  1     0    0    0     0  100  9600d*+   DB0GV 6-6   10
4  -- 40    3  1    27    3    2     0  82  1200*+    DB0TCP 0-15 580/647
5  --  1   50  1   835  377  102   55  100  19200dtr*+ DB0SHI 0-15 11/39
6  --  1   39  1   582  546  78    42  100  38400d*+  DB0GV 10-12 1/1
7  -- 40    4  1    31    3    2     0  70  1200*+    DB0ASF 0-15 229/243
8   7 40    8  8   184   36   34    1  92  1200*+
```

The single columns mean:

po: Port number
id: Port SSID, on interlink-only ports "--"
td: TxDelay in 10 ms units
qso: number of QSOs on this port, internode QSOs are also counted
usr: number of stations heard on this port (since 3 mins)

tifr: transmitted I-frames within the last 10 mins
rifr: received I-frames within the last 10 mins
tkby: transmitted kilobytes within the last 10 mins
rkby: received kilobytes within the last 10 mins
qty: quality of the channel; this is updated every 10 mins, but not if
there was nothing to send.
mode: Baudrate on this port, additionally:

"c" KISS: CRC-Mode, HDLC: Software-DCD (depends on hardware)
"d" fullduplex
"t" external TX-Clock
"r" external RX-Clock
"z" NRZ mode
"m" DAMA master
"s" port is synchronized
"u" port is user port
"y" autosysop
"+" 8 Mhz CPU-Clock (RMNC)
"! " 12 Mhz CPU-Clock (RMNC)
"# " 16 Mhz CPU-Clock (RMNC)

links: see <L>-Command

When counting the I-frames, re-iterated frames and frames which got lost due to DISC are not counted. The kilobyte statements are only the contents of the acknowledged I-frames, re-iterations are not counted, too. Thus, this is the genuine net data rate.

4.1.16. <Q>uit

Syntax: Q <CR>

The QUIT-Command ends the QSO with the node. After a "73!" you get disconnected. If you are connected from another FlexNet node, you will be reconnected to that node.

4.1.17. <S>etsearch

Syntax: S <CR>

The SETSEARCH-Command displays all digipeaters via which the FIND-Command searches for someone.

Example:

```
=>S<CR>
search digi's:
DB0ODW
DB0KT via DB0ODW
DB0AAI via DB0ODW
DB0DA via DB0ODW
DB0IE via DB0ODW
=>
```

The frame generated by the FIND-Command would be sent via DB0ODW, DB0KT, DB0DA, DB0AAI and DB0IE.

4.1.18. <T>alk

Syntax: T <call> [<text>] <CR>

With this command you can talk to other users connected to the node. There are two modes: If there is a text given behind the callsign, then this line is sent and you get back to the prompt. Thus, you have to issue a new Talk-Command for each line. By "T <call> <CR>" you get into the permanent talk mode which can be left later by using "/q". This is similar to convers mode, with the difference that it does not happen on a convers channel. All Convers-Commands are active and the current status can be displayed with "/c".

4.1.19. <U>sers

Syntax: U [n] <CR>

The USERS-command displays all users which have a QSO with or via the node. Additional information is provided:

Example:

=> U <CR>

```
1: S5      P0: DB0ODW>DG3FBL
6: S7 U1   P0: DB0ODW>DK7WJ
```

```
35: S5      P0: DL1AA>DB0GV v DB0ODW DB0KT
2014: S5     P8: DB0GV>DL1AA v DB0KT DB0ODW
```

i.e.

1. column: QSO number. The node uses this number for internal management of the QSO.
2. column: QSO state. This number shows the state of the QSO. (see appendix for explanation)
3. column: shows the number of unacknowledged frames of the QSO, if there are any.
4. column: port
5. column: calls and digipeater field

The QSOs with the node are shown first, then the ones which run via the node. Additional parameters may be specified on the "U" command line. If you enter an "i", only QSOs with the node are shown. If you enter a port number, you get all QSOs via that port. Using "U *" you get additional information about the QSOs. The parameters may be combined. For example, "U * 4" shows all QSOs on port 4 with detailed information.

Example:

=> U * <CR>

```
1: S5      F100 M3   P0   : DB0ODW > DG3FBL
6: S7 U1   F87  M7   P0   : DB0ODW > DK7WJ
```

```

35: S5      ! F50  M4  P0  : DL1AA > DB0GV v DB0ODW DB0KT
2014: S5    ! F66  M7  P8  : DB0GV > DL1AA v DB0KT DB0ODW

```

Additionally, the actual FRACK time "Fxxx" and the MAXFRAME "Mx" are shown for each QSO. On DAMA masters the DAMA priority is shown instead of FRACK. A "!" in front of the F-value says, that the QSO is using header-compression (see section 9.8).

4.2. Sysop-Commands

In addition to the commands a "normal" user may use, the sysop has some extra privileges. This includes additional commands, or additional parameters to the user commands. Commands which have additional parameters are the as followed:

```

<L>inks          setup of link partners
<M>ail           setup of nearest BBS
<MY>call         setup of node callsign and SSID range
<P>arms         setup of sundry parameters
<IO>            read inputs/set outputs

```

Additional commands are:

```

<CAL>ibrate     transmit calibration signal
<K>ill          kill a single QSO
<MO>de         setup HDLC parameters/reset L1-devices (SCCs)
<SY>sop        sysop authorization
<W>rite        write text-files (beacon- and search-files, too)
<TR>ace        monitor a port
<RESET>        cold reboot of the node
<RESTART>      warm reboot of the node

```

4.2.1. <CAL>librate

Syntax: CAL <ch> <mins> <CR>

The CALIBRATE-Command turns on the TX of a specified port (parameter <ch>) for one minute. While this time the carrier is modulated by a continuous sequence of 0 and 1, producing a "diddle" tone with a 50:50 ratio. The command is useful in 2 cases:

- It makes it possible for the link partner to set his antenna to the right direction.
- The symmetry of the modulation may be checked and the modem maybe adjusted for best results.

If there are frames in the buffer, they are sent before the calibration starts. To trigger the PTT watchdog the CAL signal is interrupted every 15 seconds for a short time.

Optionally, the calibration time in minutes may be given. Default is one minute. With the command "CAL <ch> 0" the CAL signal is cancelled.

4.2.2. <IO>

Syntax: IO <bit_no> 0|1 <CR>

With the IO-Command the output-lines of the reset-card may be set or cleared. <bit_nr> specifies the number of the bit to be changed, and 0 or 1 says whether the bit should be set to "high" or "low".

4.2.3. <K>ill

Syntax: K <QSO-No> <CR>

The KILL-Command terminates an existing QSO. The QSO number must be specified (U-Command, 1. column). Why this command? It is not made to let the sysop feel like the "big boss", but sometimes it is necessary to kill a QSO.

Example:

Station A is connected to station B, and A transmits a longer text to B. After a certain time, the receiver of the text, station B, gets busy, i.e. his TNC sends RNR. When this state is not fixed by B itself, the QSO will last for ever, at least up to the next blackout.

4.2.4. <L>inks

Syntax: L <ch|CALL|-> <CALL> [#|\$|-|@|>|)|!] <CR>

The LINK-Command is used to set up interlinks. Two parameters are needed, a 3rd one is optional. Callsigns may be written either as upper- or lowercase.

1st parameter:

port: set up direct interlink on this port
callsign: set up interlink via that callsign, i.e. the destination is reachable via a neighbor already specified.
- : the minus sign deletes the link entry

2nd parameter:

Here the callsign of the link partner is given. If no SSID's are specified, 0-15 are assumed. When only the SSID 0 is to be linked, -0 has to be added to the callsign.

3rd parameter: options

"#" link is not shown to the users, thus hidden links, for example for service reasons, are possible .
"\$" link is not checked for availability and not made known to the network.
"@ " no internode communication on this link , but may be used, if the partner is not aware of the FlexNet protocol (i.e. mailboxes)
"- " partner is not made known to the network. This makes emergency- or testlinks possible. Internode communication takes place, thus destinations are routed, only the partner stays hidden.

">" Subnet-Link. This is used to set up subnets, which will receive all information from the network, but are not made known to the network. The partner callsign and the destinations from it are saved for routing reasons, but not sent to the other network nodes.
")" Works like ">", but the link is hidden (">" + "#")
"! " no forwarding of Subnet destinations: Same like ">", but the difference is that the "gateway" node is made known to the network.

It is possible to have more than one link to the partner on different ports. The router will always use the best link available. You should remember this if changes are made. The old entry may still be valid under circumstances. It is also possible to link partners with the same callsign, or a callsign covered by the SSID range of the node. This is interesting for mailboxes, service computers, DX clusters and similar systems. Using this feature, only the node is forwarded to the network and not every single SSID of the different computers. This helps keeping the D-list in the network smaller.

Example:

```
MYCALL DB0AIS 0-10
L 1 DB0AIS-8 @ (Mailbox)
L 2 DB0AIS-9 @ (Cluster)
L 3 DB0AIS-10 @ (TCP/IP)
```

Only DB0AIS 0-10 is known to the network. If there is a connect request to DB0AIS-8, it is sent out on port 1 to the BBS. The links are not tested as usually. If the link is not available, the user gets connected to the node itself. Here, the user should get to know what is wrong, perhaps by the C-text. This routing method works on user ports, too. In our example, if DB0AIS would have a user port, the node may be connected as DB0AIS or as DB0AIS-3, and the BBS DB0AIS may be directly connected without digipeaters.

More examples:

```
L 3 DB0KT
All frames to DB0KT will be sent on port 3
```

```
L 1 DB0KT
L 1 DB0FUL
```

On port 1, 2 link partners are reachable, so the setup has both calls on port 1.

There is a principle which says that if no SSID's are specified behind the link callsign, all SSID's are routed via this port. But when a SSID is specified, only the SSID is routed.

Example:

```
L 1 DB0KT
All frames route to DB0KT, i.e. also the frames to DB0KT-1, DB0KT-2 are routed to port 1.
```

```
L 1 DB0KT-7
```

Only the frames to DB0KT-7 are sent to port 1. Other SSID's are routed by the D-List, when no other links to DB0KT are specified. On FlexNet partners the SSID-Range is automatically adapted.

To remove an entry from the link list, a "-" is given instead of the port number as the first parameter.

Example:

DB0ODW has the links

```
1: DB0KT
2: DB0EAD
3: DB0IE
```

"L - DB0KT" deletes the routing entry for DB0KT. If there is more than one link to the partner, the command has to be given several times to delete every entry. Always only the first entry is removed from the table. Links to NET/ROM partners should be set up as shown in the appendix.

4.2.5. <MO>de

Syntax: MO <port> <mode> <CR>

This command sets the operating parameters of a specified port. The mode parameters are:

```
<num>  baudrate (on internal clock)
"d"    full duplex (half duplex is default)
"t"    external TX Clock (depends on hardware)
"r"    external RX Clock (depends on hardware)
"z"    NRZ mode (depends on hardware, NRZI mode is default)
"c"    KISS: CRC mode; HDLC: Soft-DCD (depends on hardware)
"m"    DAMA master
"s"    synchronize channel with other "s"-channels
"u"    user port, activates for example TxDelay measurement
"y"    auto sysop: Stations which connect on this port without digipeaters
        in their path are automatically sysops. For security reasons, on
        RMNC this may only be defined in the EPROM.
"--"   deactivate port
"."    dummy, if there are no arguments needed on special hardware
```

The parameters baudrate, "d", "t", "r", "z" and "c" depend on hardware. Check out the hardware or driver documentation.

Examples:

```
MODE 3 19200d      ;port 3 19200 baud full duplex
MODE 3 38400trz   ;port 3 38400 baud ext. clock, NRZ
MODE 3 -         ;special case: turn off port 3
```

When a MODE command is entered, all Layer1 modules of all channels get initialized. This is not problematic, only frames currently being transmitted or received are involved. QSOs are not affected. A Mode-Command may re-initialize "hanging" SCCs. Therefore you should always try a Mode-Command first before a RESET or RESTART since all QSOs get lost in this case. The port number is not relevant, a "MO 11" will do the job.

4.2.6. <M>ail

Syntax: M <call> <CR>

With this command, a BBS callsign is assigned to the node, which can be reached by the users issuing the "M"-Command. It must be reachable in a single step and known to the autorouter.

4.2.7. <MY>call

Syntax: MY <call> [<ssid1> <ssid2>] <CR>

The MYCALL-Command is used to set up the callsign of the node. With the optional SSID's a range may be defined by which the node can be connected. The SSID range must include the SSID's of every port, no port SSID must be outside the SSID range defined by MYCALL.

Example:

```
M DB00DW 0 7
```

The node callsign is set to DB00DW. The node may be connected as DB00DW-0 to DB00DW-7. When the MYCALL is changed, this will affect only new QSO's. Existing connections stay valid under the old callsign. The internode communication is re-initialized completely, since the change of the callsign needs to be forwarded to the network immediately.

4.2.8. <P>arameters

Syntax: P I <value>
 or P S <value> <port>
 or P T <value> <port>

The PARAMETER command is used to set up the TxDelay, SSID and the node timeout of a specified port.

"P I <n>" sets the node timeout to <n> minutes where a range from 60 to 255 is valid. n=0 (recommended) disables the timeout.
"P T <n> <port>" sets the TxDelay of port <port> to <n> in 10ms units.
"P S <n> <port>" sets the SSID of port <port> to <n>. When an already-set SSID has to be deleted, <port> has to be 16.

Why do we need SSID's ? They do two jobs: Only on ports which do have a SSID, everyone is allowed to connect. Exclusive interlink ports therefore should not have SSID's (exception: links to NET/ROM partners, see appendix). The SSID is also needed for routing purposes, if a user who is not in the MHeard-list shall be connected on a specified port. The connect then needs to go via the according SSID, i.e. via <nodecall>-<port>-<SSID>.

4.2.9. <RESET>

Syntax: RESET <CR>

This command cold-reboots the node. All QSOs, parameters in the RAM and the text files get lost. The default parameters as stored in the RMNC master EPROM are

set. You should use this command only when your parameters got disordered and you want to reset to EPROM defaults. This command is available only on RMNC systems.

4.2.10. <RESTART>

Syntax: RESTART <CR>

The RESTART-Command basically does the same as the RESET command. However, parameters and text files remain in memory, so usually you should use this command instead of RESET. You should use both commands only in emergency, since all QSO's and the routing information get lost. This command is also available only on RMNC systems.

4.2.11. <SY>sop

Syntax: SY <CR>

The SYSOP-Command is used to do the sysop authorization. When a remote request is sent to the node from the sysop to enter the sysop mode, the node answers with a random number. This number has to be answered by the sysop with the exact combination.

The algorithm used is easy enough to do the calculations without a calculator, therefore security is limited. Perhaps the algorithm will be changed in future. How does it work ?

Example:

```
=>SY<CR>          <- sysop command
12345>             <- random number from the node
```

Assumed that the sysop code programmed in the node is 54321. Now the calculation:

* multiply the coinciding random & sysop numbers :

```
1 2 3 4 5          <- random number
5 4 3 2 1          <- sysop secret number
```

1*5=5; 2*4=8; 3*3=9; 4*2=8; 5*1=5

* Now sum up the products:

5+8+9+8+5=35

Ready. 35 is the number the node expects to receive. Now you are sysop (provided, your calculation was OK). To make it more difficult for spies, you may send the SY-Command more than once. The calculation has to be right only once. If the other answers are wrong, it is more difficult for a spy to catch the secret code. After successful login as sysop no message is returned. You may now try out whether it went all right by a harmless command like TIMEOUT. If you are logged in as sysop, the node timeout is not valid for you anymore, i.e. you may stay connected to the node as long as you wish. The sysop authorization is removed by disconnect, reconnect (link reset) or a Connect-Command. It is possible that more sysops are logged in at the same time.

4.2.12. <TR>ace

Syntax: TRACE <ch> [<call>] [<|>] [>] [#] [\$] <CR>

By using the TRACE-Command, you can monitor the traffic on a given port. This of course only works as long as the buffers do not overflow. When your own QSO is fast enough, you may monitor for a longer time. Compressed QSO's are shown only if your node is the source or the final destination of the QSO. The command is cancelled if the buffers overflow or you type another command. Only one sysop may monitor only one port at one time. You should note that this command needs a large amount of memory and bus capacity, which will slow down specially the monitored channel. Therefore, you should not use it too often and under no circumstances permanently. This command is only available on Solomaster systems.

Options:

do not display RR/RNR/REJ-Frames
\$ suppress the I and UI texts, i.e. display only frame headers
<call> trace only this callsign. Only source and destination callsigns of a frame, i.e. no digipeater calls are checked. SSID is taken into account if specified.
> only frames sent
< only frames received

4.2.13. <W>rite

Syntax: WRITE <A|B|C|H|I|L|S> <CR>

By using the WRITE-Command, the texts for L(A)test news, (B)eacon, (C)-Text, (H)elp, (I)nfo, (L)ocal and (S)etsearch may be entered. All texts except (B)eacon and (S)etsearch may have any desired format. The C-text is sent after the standard system prompt at the beginning of a connect. Standard prompt is "xxxx/FlexNet Vx.x". The C-Text is shown after this.

After any Write-Command, the amount of available memory on the RMNC is shown.

We recommend the following usage of the texts:

LATEST NEWS: recent information about the node and things every user should know

INFO: general information about the digipeater. QTH, hardware, antennas, use of IO-ports etc. You should not forget to mention the user QRG's.

LOCAL: this text is appended to the C-Text, when the user comes direct, i.e. without digipeaters in the path. This is the right place for information about the channel access method and other information which is only interesting for local users.

HELP: A short user's guide to the system with the most important commands

The text files cannot be saved in the EPROM due to space limitations. The end of the text is marked by either /EX or Ctrl-Z. The text is saved up to the last line before the /EX. Since many PR-stations use "split screen" programs, it is recommended to begin every text except the C-Text with a single <CR>. This looks much better.

The Beacon-file has a special format. You may set up any beacon on any port. The file format is as follows:

```
# <t> <p> <tocall> [via [via ...] ] :Beacon
text.....#

<#>          delimits the different beacon information
<t>          time between two beacon transmissions, in minutes.
(1..255 minutes)
<p>          port number where the beacon is to be sent
<tocall>     Destination call of the beacon, for example "BEACON", "RMNC",
             "FLXNET", "TEST" or similar , there may be up to 8 digipeaters
             specified, via which the beacon will be sent.
```

Example:

```
10 0 RMNC:Digi Odenwald * JN49IQ * Krehberg/Odw. *#
30 1 RMNC DB0KT DB0ODW:DB0KT QRV#
5 0
FLXNET:Testbeacon DB0ODW
```

Our example consists of 3 beacons, each delimited by a "#". (beacon1...#beacon2...#beacon3...) Between the single statements there may be <CR>s to improve readability. It is not important whether the callsigns are upper- or lowercase. Source callsign of the beacon is always the mycall of the node. When no beacon-text has been entered since the last cold-reboot, the default-beacon is sent every 3 minutes:

```
#3 0 FLXNET:RMNC/FlexNet V3.3d
```

All beacons are sent as UI (Unproto-Information) with the command bit set.

The SETSEARCH-file has a special format, too. There may be as many search paths as you like. The limit is dependent upon the available memory. The number of the digipeaters is limited to 7. The format is:

```
<call1>
<call1> [ <call2> [ <call 3> [ <call 4> [ <call 5> ]]]]
```

The SETSEARCH-file contains all paths via which the FIND-Command sends its UI-Frames. The first callsign in the line is the digipeater which shall send the UI to the user, the following digipeaters specify the path to the destination digi. These paths should always be identical to the path a user will use. This means, digipeaters on the route to the destination should be left out, the autorouter will know the way.

Example:

```
DB0ODW
```

DB0DA via DB0ODW
DB0KT via DB0ODW
DB0AAI via DB0ODW

First line says that the UI-frame is sent via the local user port. Second line demonstrates how a path to DB0DA is set up. DB0DA is reachable from DB0ODW via autorouter.

5.Command Overview

Statements inside [] are optional.

5.1. User Commands

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]

5.2. Sysop Commands

CAL <ch> [min]	- port <ch> sends calibration signal
IO <bit_nr> 0 1	- set Output bit nr. <bit_nr> to 0 1
K <QSO_nr>	- kill QSO nr. <QSO_nr>
L <ch> <call>	- route <call> to <ch>
L <viacall> <call>	- route <call> via <viacall>
L - <call>	- delete link <call>

```

M <call> - assign local BBS
MY <call> [ssid1 ssid2] - set mycall and SSID range
MO <ch> <x> - set port <ch> to mode <x>
P I <x> - set node timeout to <x> minutes
P S <ssid> <ch> - set SSID <ssid> on port <ch>
RESET - cold reboot
RESTART - warm reboot
SY - sysop authorization
TR <ch> [...] - monitor port <ch> (solomaster only)
W <A/B/C/H/I/L/S> - write text-files (end: /ex)
  A - text LATEST NEWS
  B - text BEACON
  C - text C-Text
  H - text HELP
  I - text INFO
  L - text LOCAL
  S - text SETSEARCH

```

=====

6. Legal Terms - License Agreement

=====

The following restrictions have only the purpose to guarantee the quality and the development of the software and, of course, to avoid commercial use of it, except in special cases agreed on with the author of each module. By experience, on uncontrolled distribution it happens that only parts of the software or obsolete versions of it are spread. This results into problems at the end-user's and into unnecessary and annoying check backs. Bad experiences cause the list to grow constantly...

- RMNC/FlexNet, PC/FlexNet and the accompanying utilities and the documentation are products of Gunter Jost, DK7WJ. Exceptions are marked as such. All rights reserved by the author. The user is granted the right to use the software under the following conditions:
- The software is used only for non-commercial purposes within amateur (HAM-) radio. All other usages, especially in other radio services, need a written consent by the author in each separate case.
- The legal rules of amateur radio operation are strictly followed.
- Commercial copying and sale of the software is not allowed without prior written consent of the author.
- No changes must be made at the software which are not explicitly agreed with the author. This does NOT apply for setting specific parameters.
- Copyright marks and copyright messages of the software modules must not be changed or removed.
- The software must not be distributed via BBS systems or public accessible bulletin boards, neither in part nor as a whole.
- Neither the author nor the distributors of the software may be liable for any damages, no matter of what kind, which may occur when using the software. THE RISK OF USING THE SOFTWARE IS ENTIRELY YOUR OWN!

Under this conditions you may make as many copies of the distribution disk as you wish and give them away, where you always have to state the original source (FlexNet-Gruppe Darmstadt, G. Jost, DK7WJ).
The sourcecode of the software is not available.

6.1. Disclaimer

Neither the author nor the distributor of the software may be liable for any damages that may eventually occur when using the software RMNC/FlexNet or PC/FlexNet. The software is provided "AS IS", without warranty of any kind, including but not limited to the warranties of merchantability and fitness for a particular purpose. No features should be taken for granted. The documentation may contain errors or mis-translations.

By using the software you agree to the conditions above.

7. Installing RMNC/FlexNet

7.1. The RMNC

The RMNC consists of an port controller card for each port on a euro-sized board (100 * 160 mm). Additionally a so called reset card is needed which contains the system watchdog and the In/Out-ports for supervisor and control reasons. The cards are run in master/slave mode, i.e. one card is the master, the other cards are the slaves. The whole communication on the system bus is controlled by the master, that means the master polls every slave for information and mediates them where they belong. The only hardware difference between the cards is that the master contains a different EPROM than that of the slaves. This has the advantage that only the master needs to be configured, all slaves receive their configuration information from the master. After a reset, the master determines the number and the addresses of the attached cards. Then these cards get initialized with their parameters. If there is a malfunction at one of the port controller cards, the master gets to know this after the watchdog-reset which will occur. The defective card will not be polled anymore, and the node will still work - provided that the damage does not block the bus.

After the reset, all controller cards are ready to accept QSO's. A connection via the digipeater counts as a QSO, too. The software on the controller cards will manage the complete L2 connection on its own and mediates all data directly to the according controller card.

7.2. The RMNC-Solomaster

Since V3.1 it is possible to use one RMNC-card without HDLC/RF-port. This card was named Solomaster. The Solomaster has several advantages. At first, it can service the slaves faster, because it has not to worry about the RF port. This brings about a notable increase of the data throughput on huge and busy nodes.

Secondly, since we could delete the L2- and L1-modules, we gained space in the EPROM to implement new features. When generating the EPROM for the Solomaster you have to pay attention to the following: The first command line in the parameter file needs to be "SOLO". For the Solomaster, a clock frequency of 4, 8 or 12Mhz is possible. When assembling the Solomaster, you do not need to install the SCC and the parts for the modems. When using a Solomaster, it gets port number 0; this can be seen in the <P>-list. Only the Solomaster contains the full features of FlexNet. The normal master only contains a minimum of needed functions and will probably not be supported anymore in further versions. Even now the fullmaster channel should not be used for very busy channels, since it cannot maintain many QSO's and is relatively slow.

7.3. Card Addresses

When installing the RMNC/FlexNet software, the card addresses need to be configured. Attention, the master card has no address, i.e. all DIP-switches need to be open! For addressing the other cards, please refer to the hardware manual. The cards may be addressed in any desired order, gaps do not matter.

In the parameter table (P-command) the master always is port 0. When using a Solomaster, then no port 0 is shown. Now the EPROMS can be installed on their boards. Only ONE master EPROM has to be installed, all other cards must be equipped with (identical) slave EPROMS.

The RMNC should be ready for use now. Immediately after the reset, the first beacon is transmitted on the master port or on solomaster systems on port 1.

7.4. Generating a MASTER EPROM

The software is distributed unconfigured. To configure it, you need at least an EPROM programmer capable of programming 27C512 EPROMS, and an IBM compatible PC.

7.4.1. Parameter Compiler

The master EPROM is configured with a PC program. With the aid of this compiler and a parameter file, a configured binary file will be created which is ready to be programmed into the EPROM. The slave EPROMS do not need to be configured as they are the same on every RMNC. To get the compiler to make the binary file, you have to create the parameter file first. The parameter file is a plain ASCII file which can be made with any editor. This file contains all necessary parameters. The syntax of the file is similar to the syntax of the sysop commands. When you have finished making the file, you can start the compiler. If you are lucky you now have a file with the same name as that of the text file, but with the extension ".BIN" in the directory. This file also exists when warnings occur there. If the compiler detects syntax errors, no .BIN-file will be created. For control reasons a second file with the extension ".LST" is produced. You should have a look at it to ensure that the compiler understood everything. The .BIN file can be burned into a 27C512 EPROM now.

Syntax of the compiler:

```
MRMNC <parmfile> [<binfile>] <CR>
```

The first command line parameter is the file name of the parameter file with extension. Specification of the <binfile>-name is optional and will only be necessary if the name desired is different from the name of the text file. If a file with the same name exists, the compiler will ask if you want to overwrite it. The compiler creates a list file (".LST") which contains all default parameters.

7.4.2. Structure of a parameter file

 Everywhere in the file, except inside of commands, comments are allowed. Everything which stands behind a "*" or a ";" is ignored. Empty lines are ignored, too. The syntax of the commands is the same like entering the commands into the digipeater. The file is not case sensitive. It is recommended that you note down everything carefully. This makes changes easier later.

The SOLO-command has to (if desired) be the first command in the file, the order of the other commands does not matter. You should pay attention to the commands SPEED and END, which can only be given in the parameter file, and not as commands online. The WRITE-command does not work either, texts can be entered only online.

Differences to the normal syntax occur at the SYSOP-command. Here the secret number for the node is entered.

Example:

```
* Configuration of DB00DW
* date: 1st Apr. 1993
```

```
SOLO                * We are using a SOLOMASTER!
SPEED 8             * The SOLOMASTER runs at 8 Mhz clock
                   * speed
                   * (4/8/12/16 Mhz are possible here)
Mycall DB00DW 0 7   * Mycall of the node, SSID range 0-7

Sysop 12345         * This is the secret number for sysop
                   * authorization

P SSID 0 1          * Userport is port 1 with SSID -0
P SSID 2 5          * 2nd Userport (-2) on port 5

L 2 DB0KT           * Interlink to DB0KT
L 3 DB0AAC          * Interlink to DB0AAC
L 4 DB0IE           * Interlink to DB0IE
L 5 DK7WJ-8        * Test link to DK7WJ-8 with no forward
                   * to the network

IO 7 1             * set IO-bit 7 (turn on PA to DB0IE)

mode 1 96t          * Port 1 with 9600bd and external
                   * TX-Clock
mode 2 96d          * Port 2 with 9600bd fullduplex
mode 3 96d          * Port 3 with 9600bd fullduplex
mode 4 24           * Port 4 uses 2400bd
```

```

P I 0          * No timeout for the node (Infobox)

* Layer1-parameters

parm TxDelay 25 1      * User ports need a higher TxDelay
p t 6 2              * Interlinks only need 60 ms
p t 6 3
p t 6 4
parm TxDelay 25 1      * this is another Userport

END                * The END command must be the last one!

```

7.5. Slave-EPROMS

The slaves need not to be configured. The EPROM file RM_SLAVE.BIN is intended for programming into an 27C128 EPROM. When a 27C256 EPROM is being used, the file must be placed at the upper half, beginning at \$4000.

7.5.1. EPROM Patch Slaves 9-15

The RMNC2 controller cards contain only a 3bit address switch, thus you can only address a maximum of 8 cards (1 master and 7 slaves). For the slaves 9 - 15, the byte \$3F00 (\$7F00 on 27C256) must be patched to a value different to \$FF. This adds "8" to the address setting of the DIP switches. On RMNC3 cards this is not necessary anymore, here the address can be set directly.

7.5.2. KISS-Slave

It is possible to use the KISS protocol directly with the node. There is no separate KISS EPROM anymore, on the RMNC2 you have to add a single wire between pins 18 and 39 of the 6522. On RMNC3 you have to short the jumper JP1. Since the KISS protocol is not secure, a CRC mode was added. It can be activated by a MODE-command. Specifications of this new mode are available from the author.

8. Installing PC/FlexNet

=====

PC/FlexNet runs completely in the background as a TSR, that means that other applications can run simultaneously if there is enough memory left. However, the FlexNet infobox and the beacon generator may not be serviced under some circumstances, so that a dialogue with the node is impossible. This only happens when using badly programmed applications.

QSO's via the digi and the internode communication are not affected and should always work, whatever the PC has to do. Probably things get slowed down a little bit.

8.1. Hard- and Software Requirements

- PC/XT, better AT with at least 512kB RAM
- PC/FlexNet needs 200kB RAM, plus space for the L1 drivers and applications
- Operating system MS-DOS 3.1, better 5.0 or 6.2. Tests with MS-DOS 6.0 caused problems, we have no experiences with DR-DOS or other DOS versions. We recommend the use of MS-DOS 5.0 or 6.2, here most modules can be loaded into the UMB's, provided there is enough memory available.
- IO-ports as necessary, according to the L1 drivers available

Principally, a PC/XT will work. The gained performance mostly depends on the speed and the throughput of the L1 hardware drivers.

PC/FlexNet supports several loadable L1 drivers. They are installed in the memory by simply calling them. This makes it easy to support any hardware. A "driver development kit" for interested developers is available from the author. The port numbers derive from the order of the driver installation. A single driver can support more than one port depending on the hardware. FlexNet, however, is limited to a maximum of 16 ports, the last port (15) is reserved for internal purposes.

The port drivers are included on the distribution disk, depending on which drivers are available. For every L1 driver there is a appropriate *.DOC-file which explains the installation. By starting the drivers with the option /?, you will get a short help as well.

Many people on different places are working on PC/FlexNet at the same time. Thus, there always new versions of kernels, drivers and applications. It is always a good idea to ask for new versions if there occur any problems. Changes, even in the installation procedure, may happen. Please read the *.DOC-files carefully!

=====

8.2. Installation and Configuration

=====

At the beginning, all files must be copied into a directory which should be in the DOS search-path. The start of PC/FlexNet should be done via a batch file because most of the L1 drivers need additional command line arguments. Occurring errors should abort the batch file. A sample batch file is on the distribution disk and can be easily changed to fit your needs.

FLEXNET.EXE must be loaded first, then - if a node is to be installed - FLEXDIGI.EXE. Pure endpoints (Terminal, Cluster, BBS and so on) should not use it. Then the L1 drivers follow in the order you require. At last, the activation of the modules is made by the utility "FLEX". After doing this, no more port drivers can be installed.

FLEXNET.EXE has an optional parameter, which specifies how many RAM may be used by FlexNet. Default is 15kb, but this lasts only for few QSO's. The minimum for nodes with several ports is about 80kb. Depending on how many ports you use, you should experiment with this value. FlexNet loves memory more than everything else and runs best when it has about 30kb per port and additional 20kb for administration.

To load the modules, generally (from DOS 5.0 onwards) you should use the "LOADHIGH" or "LH" command. It does not do any harm if there is not enough memory in the UMB; the file is loaded into conventional memory then. You still gain a little memory, since the environment blocks do not fragment the memory. You may check this by using the "MEM /D" command.

Calling FLEX.EXE with the argument "/U" uninstalls all L1 drivers and removes FLEXNET.EXE from memory. As usually on DOS, no other TSRs should be loaded after FlexNet, otherwise your machine might crash.

The first start of FLEXNET.EXE creates an empty parameter file. Port 15 is generally the interface for applications. The parameter AUTOSYSOP ("y") is set on this port, you should not change it. Now you should set the sysop secret code using "SYSNUM.EXE". The secret code becomes valid at the next start of PC/FlexNet. With "TNC.EXE" you can connect the node now and continue in setting the parameters. If you made a mistake, you could simply delete the file "FLEXNET.FPR" and begin again. "TNC.EXE" is a simple TNC emulation. With "<ESC> H <CR>" you get a short help. The node can be connected with "<ESC> C <CR>".

The parameter setting of the software can be done now either by the TNC emulation or via remote control. Please check the documentation of the L1 port drivers. Like always on FlexNet, the rest of the parameter settings is very easy and can be finished in a short time.

Before you decide to build a digipeater using PC/FlexNet now, you should think about the following: The RMNC is still the preferred platform for FlexNet, and something that does not work there will not work on the PC either, except from some bagatelles. The user shall find a uniform and well known (from the RMNCs) user interface. Who prefers the optimum of reliability and performance for minimal costs and maintenance should use the RMNC.

=====

9. Appendix

=====

9.1. Protocol version AX25 V2

FlexNet understands QSO's with AX25 version 1 and AX25 version 2. A QSO with the node itself can only be done using version 2. When the node encounters a SABM to it with V1, it replies with DM. When station A is trying to connect a station B via the digi using V2, and the other station responds with V1, the connection happens in V1. However, hop-to-hop acknowledgement is disabled for this QSO, it is not even mentioned in the user list.

9.2. Unproto-Frames

All UI- (unproto) frames are transmitted and routed, even frames using protocol version 1. This makes it possible to run TCP/IP without any problems, since most TCP/IP programs send their UI-frames in AX25 V1. The length of the I-field of the UI-frames must not exceed 256 bytes.

9.3. RNR-Behavior/Recovery

Because the memory of the node is limited, it may happen that a user is set to RemoteBusy (RNR) from the digipeater. FlexNet works very defensively in this matter to ensure that there is not too much memory used for one single QSO. It makes no sense anyway to buffer more than about 10 frames for one QSO. RNRs therefore do not mean that the memory of the node is short, but only that there are enough frames buffered to service the QSO fluently. This ensures a fair partition of the channel resources to the QSOs. RNRs often happen in convers mode when one of the partners has a slow link. When the RNR condition has gone, the braked station gets a RR frame with the poll bit set. This causes a RR-final from the station, but only this ensures that the status change is detected by that station. On many AX25-implementations only a RR without poll is sent, which causes hangs of several minutes when it gets lost.

9.4. Reconnects

A special problem on digipeaters with hop-to-hop acknowledgement are reconnects. A reconnect does mean that a station tries to (re-) establish another connect to its partner using the same calls (and SSID's) and the same or a different path. When this connection happens on a different path, problems may arise. The digi holds the running QSO in his table. When there are no unacknowledged frames, nothing happens. The "zombie-QSO" dies after 15 minutes. However, if there is still data to be sent, the digi tries to service the other station. But since the QSO was reestablished using the other path, the sequence numbers of the original QSO do not match with the new ones. This causes a FRMR (frame reject) and the connection is aborted. The best way to avoid this trouble is to end an existing connection properly before establishing a new one.

9.5. I-Polling Method

During a connection, it often happens that an I-frame is not acknowledged correctly. In this case, AX25 V2 says that a poll frame (RR) has to be sent in order to query whether the I-frame was received correctly. If not, the I-frame is retransmitted, otherwise the next frame is sent. However, it is provable that this method causes a useless load to the channel, when the I-frame in question is a very short one. With the I-polling method, the short I-frame is repeated immediately with the poll bit set. This is a mixture of the old V1 and the new V2-protocol version. For statistical reasons, the threshold of this method is set to 40 bytes. Any I-frame with fewer than 40 bytes length is repeated immediately when it was not acknowledged within the FRACK-time (T1). The I-poll method is only used at the first 3 retries, then normal polls are applied.

9.6. Channel Access Method

Up to FlexNet V3.1 the p-persistence algorithm was used. This had the disadvantage that the send-willingness of the station could only be adjusted manually. Since V3.2 there has existed a fully adaptive channel access method, which adapts itself by determining the number of users and the channel allocation to calculate an optimal compromise between throughput and the collision-probability. Variables used:

B Baudrate (1/s)
 TxD Tx-Delay (s)
 R Random number (1 .. 10)
 U Total of the users on the channel
 K channel allocation
 t1 waiting time after transmission (s)
 t2 waiting time between two TX-tries

$$K = \left(\frac{19200/s}{B} + 1 \right) * (U + 1)$$

If $K > 255$, then K is set to 255. This prevents an increase of $t1$ on more than twenty users on 1200Bd.

$$t1 = K * (R+1) * 8 * TxD$$

If $t1 > 10s$, then $t1$ is set to 10s, this can only happen on baudrates below 1200Bd.

$t1$ starts when its own transmission has ended. $t1$ is stopped as long as the channel is busy. As long as $t1$ is active, the TX is blocked. When $t1$ has expired, and the node wants to transmit, the DCD is checked in $t2$ intervals. If there is only one user, $t2$ becomes 0.

$$t2 = 2 * TxD * (U-1)$$

When the channel is busy after the expiration of $t2$, $t2$ starts again, otherwise the transmission is started. $t2$ is not interrupted by the DCD.

9.7. Layer 2 States

 The second column of the user list shows the actual L2 states. They shall be briefly explained here. For more information, please refer to the AX25-V2 protocol specification. Attention, we are using other numbers for the single states. Only the numbers 1-7 are the same as in the specification. The other states, which are only relevant for BUSY (NR) states, have been done by using bit masks, i.e. there are offsets added.

State	Description
1	disconnected
2	link setup
3	frame reject
4	disconnect request
5	information transfer
6	REJ frame sent
7	waiting acknowledgement

Disconnected

 This state is the default state. No QSO exists at the moment, therefore it is not shown in the user list.

Link Setup

A connection is being set up, i.e. SABM-frames are being transmitted, the acknowledge (UA) has not been received yet.

Frame Reject

Due to a sync error a FRMR-frame was sent, the connection is restarted. This may happen due to a software error or due to two stations with the same call sign.

Disconnect-Request

A connection is being disconnected, i.e. DISC-frames are being sent. The acknowledge (UA) has not been received yet.

Information Transfer

Hopefully, the most seen state. The connection is established, info-frames are being exchanged.

REJ Frame sent

A REJ-frame was sent, because a received frame was not in the correct order. A retransmission is requested.

Waiting Acknowledge

The link is being polled, either because I-Frames have not been acknowledged or because the link has to be checked.

Busy-States

When the station is busy, i.e. cannot receive any more packets, 8 is added to the state number. When the opposite station is busy, 16 is added to the state. When both stations are busy, 24 is added. Thus, numbers up to 31 may occur.

9.8. Header Compression

----- --> Only available on Solomaster systems!

In the address field of the L2 protocol, up to eight digipeaters can be specified, which identify the path of a frame. In contrast to other node concepts (Net/ROM), FlexNet uses these fields for the routing of the frame to avoid a separate L3 header. All routing information is put into this address field. A disadvantage is, that this address field may grow very long, since every call sign allocates 7 bytes. Due to the VirtualCircuit-concept, however, this becomes redundant when a L2 connection is established. So with FlexNet V3.2 a compression of these headers was introduced. Instead of the 14 - 70 Byte, the header is now only 7bytes long, which causes a significant increase in performance. Especially short frames (RR, short I-Frames) shrink to 10 - 25% of the original length, which speeds up some interlinks to the double speed! Of course, this method can only be used if both partners of the interlink are capable of the protocol used. Address fields are only transmitted completely during link setup, during the QSO itself only the shortened addresses are used. Another shrinking of the header would be possible but could be dangerous if used on non exclusive channels. An obvious call sign needs to be transmitted to avoid mistakings. During the link setup, the own QSO number is sent behind the control field as a 14bit integer. As opposite to the address field, this number is

is discarded. This may happen if there was a reset and the QSO number is now given to an other QSO.

- Compressed SABM's or UI's are not generated but ignored: SABM's should always contain the complete path. UI's can only very seldom be treated as a QSO, so compressing the callsigns is of no use.
- Header compression is only available on Solomaster systems. If you do not like it, you may deactivate it by using fullmaster software.

9.9 Clock options

9.9.1. Operating modes of the SCC

The ZILOG 8530 SCC contains two separate serial ports, every channel has its own clock generator. Since a HDLC channel needs two different clocks (x and 32times x), the clock generator of the second port (port B) is used to generate the single clock(x). Thus, port B is not usable for data transmission. If you had used both ports for data transmissions, the RX-Clock would have needed to be generated externally and additional parts would have needed on the card. When using RMNC2 cards with their own AFSK modem, the clock is derived from the PCLK supplied from the reset card (RMNC3 has a local oscillator). Port A of the SCC is used for data transmission. The clock generator A supplies the clock for the DPLL of the receiver(32*x), the clock generator B supplies the TX-Clock(x). On RMNC3, the A-port supplies the 32*x-Clock needed for the FSK modem. The TX-Clock is generated by the modem. Thus, "external clock" needs to be set here.

9.9.1.Clock Options

During the development of RMNC3 it proved necessary to change the clock options. Especially affected is the combination of external TX and internal RX clock. However, who uses some of the clocks on the RMNC for attached modems should have a close look at the following table. The new options and pin functions are as follows:

```
Mode:  --: internal RX and TX      I: Input
        t: external TX            O: Output
        r: external RX
        tr: external RX and TX    SCC-Pins: TRxCA=14, RTxCA=12, TRxCB=26
```

Mode	TRxCA	RTxCA	TRxCB	comment
--	O F32	I TxC	O TxC	short-circuit TRxCB - RTxCA needed!
t	O F32	I TxC	O F32	ATTENTION NEW: Put clock to RTxCA !
r	O TxC	I RxC	O F32	as usual, but swapped the clocks
tr	I TxC	I RxC	O F32	no changes

On RMNC2, almost everything stays as usual. On RMNC3, internal clock is set, when the AFSK modem is used. When using the FSK modem or the AFSK option with echo, external clock must be used.

When using an external modem, either Tx and/or Rx-Clocks can be supplied from outside. Between these options you can switch by using the MODE-command. Since the external modem must supply the single Rx-Clock, the internal DPLL of the SCC is not used. The different clocks are injected via the modem-disconnect connector. Take care, between RMNC2 and RMNC3 there are different mappings ! On RMNC2 TRXcA is the input for the Tx-Clock (pin 16 of the modem connector), RTXcA is the input for Rx-Clock (pin 18). On the RMNC3, there are only TxC and RxC. Be careful about the following, too: Under normal usage (internal modem), TRXcA/TxC is an output line, when using an external modem, this line becomes an input line! If you have connected an external modem, and you switch to normal mode, probably two outputs work against each other. On the RMNC2, RTXcA is short circuit with TRXcB. Via this circuit the A DPLL gets its 32x-clock. When using an external modem, you have to cut this connection.

9.10. Links to Net/ROM partners

FlexNet V3 makes it possible to link Netrom neighbors and to forward these interlinks, provided they are working. For the following explanations, we assume that our FlexNet node DB0FLX has an interlink to the NETROM node DB0NR. The Netrom node consists of several TNC's with different SSID's. The user port usually has the SSID -0. DB0FLX is on DB0NR-4 for example. Obviously, we should add DB0NR-4 to the link table of DB0FLX as it was done in V2. However, this has disadvantages:

- The Netrom node perhaps has other FlexNet neighbors. If they forward their DB0NR-x information, too, DB0NR appears more than once in the FlexNet D-tables. This makes the D-tables very uncomprehensive.
- When a user wishes to connect DB0NR, he does not know which path to DB0NR the best is, since FlexNet may route the different DB0NR-x a different way. However, it is possible to solve the problem with proper entries in the link list. The idea is, that for the network it is enough to know that DB0NR exists. How the single TNC's of DB0NR are reached is only of interest for the neighbors of DB0NR. The link entry of DB0FLX has to be as follows (provided that DB0NR is on port 5)

```
L 5 DB0NR-4 -      * the "-" means that the link is checked,  
                  * but not forwarded to the network
```

```
L DB0NR-4 DB0NR   * this is the trick: The link to the user  
                  * port DB0NR-0 is checked and forwarded to  
                  * the network as ssid-range 0-15
```

FlexNet now knows a way to DB0NR, but only DB0FLX knows that it has to go to DB0NR-4 directly, and to the other SSID's of DB0NR via DB0NR. In the D-lists only a DB0NR (0-15) appears. The user port DB0NR-0 is directly reachable. If there would exist a Convers-TNC DB0NR-8, it could be directly reached, too. The thing becomes even more Wizzardly when DB0NR has a second FlexNet neighbor. We assume DB0ZZZ is reachable via DB0NR-2. We add the following two link entries:

```
L DB0NR-4 DB0NR-2 $      * indirect link. Not tested, not forwarded,  
                          * but displayed in the link list
```

```
L DB0NR-2 DB0ZZZ      * Link to FlexNet via DB0NR. Will be tested  
                      * and used by FlexNet.
```

That has been the good news, and now the bad news:

- The D-command shows a route to DB0NR-14 now, even if it does not exist. The Link-command is only capable of linking single (-0 -1 -2) or all (0-15) SSID's...
- The Netrom linkports have to turn on L2 digipeating. When accessing DB0NR-0, this should cause no trouble, since the diode matrix works without collisions.

Furthermore, the link as set up via this trick suffers from the not available hop-to-hop-acknowledgement. This compromise is acceptable when looking at the advantages of the network: When the links are working 100%, it has no disadvantages, hop-to-hop-acknowledgement is not necessary if there are no RETRIES. If the link is bad, FlexNet uses a better route, if available. When the Netrom sysop is stupid and still wants L2 digipeat turned off on the interlinks, the link still should be set up as described. Then the NETROM node is not forwarded and the users have to find their way themselves. As soon as the digipeating is turned on, everything will work OK again.

9.11. Subscription Service

For the FlexNet software, a subscription service is available to sysops. All sysops get always the new versions without looking out for it everywhere. This service is provided free of charge, but cannot be guaranteed. It depends on whether there come in enough donations for funding this service, also there is not always enough time to do it. If you are interested in subscribing, ask the author.

9.12. Correspondence

The FlexNet-Group of Darmstadt, and therefore the authors and the maintainers of the software are reachable by mail :

FlexNet-Gruppe Darmstadt
Gunter Jost, DK7WJ
Lichtenbergstr. 77
D-64289 Darmstadt
Germany

Or via AX25 Email: DK7WJ@DB0GV.HES.DEU

The manual was translated by Mario Lorenz, DG0JAB @ DB0JES.#SAA.DEU.EU, Internet-Email: ml@donald.bsz.szb.sn.schule.de. As I am not a native English, this translation surely contains mistakes. Please send bug reports or suggestions to the my Email-address. Thanks.

1. News on Version 3.3.....	1
1.1. Channel Synchronization	1
1.2. DAMA-Master	2
1.3. DAMA-Slave	2
1.4. Local C-Text	2
1.5. New Link Options.....	2
1.6. New Trace Options	2
1.7. Talk Command.....	2
1.8. TxDelay Measurement.....	3
1.9. New MODE-command Parameters	3
1.10. New U-Options	3
1.11. New L-Option	3
1.12 Heardlist / MH-Command	4
1.13. Better command line interpreter	4
1.14. Other changes to Version 3.2.....	4
2. INTRODUCTION	4
2.1. The history of FlexNet.....	4
2.2. Important features of FlexNet.....	5
2.2.1. Hop-to-Hop-Acknowledgement	5
2.2.2. Connectability	6
2.2.3 Remote Control.....	6
2.2.4. Installation.....	6
3. Using FlexNet	6
3.1. Adaptive Parameter Settings.....	6
3.2. The phases of a connection via the Digipeater	7
3.2.1. Link Setup.....	7
3.2.2. Information Transfer.....	7
3.2.3. Link Failure.....	8
3.2.4 Disconnect.....	8
3.3. Methods of Routing	8
3.3.1. Destination Table Routing	8
3.3.2. Link Table Routing	8
3.3.3. Heardlist routing	9
3.3.4. SSID routing	9
4. Infobox-commands	10
4.1. User Commands.....	10
4.1.1. L<A>test News	10
4.1.2. eacon	10
4.1.3. <C>onvers mode.....	10
4.1.4. <C>onnect.....	11
4.1.5. <D>estinations	12
4.1.6. <F>ind.....	12
4.1.7. <H>elp	13
4.1.8. <I>nfo	13
4.1.9. <IO> (In/Out).....	13
4.1.10. <L>inks.....	14

4.1.11. <LO>cal	14
4.1.12. <M>ail	14
4.1.13. <MH>eard.....	15
4.1.14. <MY>call.....	15
4.1.15. <P>arameters	15
4.1.16. <Q>uit.....	16
4.1.17. <S>etsearch.....	16
4.1.18. <T>alk.....	17
4.1.19. <U>sers.....	17
4.2. Sysop-Commands	18
4.2.1. <CAL>librate.....	18
4.2.2. <IO>.....	19
4.2.3. <K>ill.....	19
4.2.4. <L>inks.....	19
4.2.5. <MO>de.....	21
4.2.6. <M>ail	21
4.2.7. <MY>call.....	22
4.2.8. <P>arameters	22
4.2.9. <RESET>.....	22
4.2.10. <RESTART>	23
4.2.11. <SY>sop	23
4.2.12. <TR>ace.....	24
4.2.13. <W>rite.....	24
5.Command Overview	26
5.1. User Commands.....	26
5.2. Sysop Commands.....	26
6. Legal Terms - License Agreement.....	27
6.1. Disclaimer	28
7. Installing RMNC/FlexNet.....	28
7.2. The RMNC-Solomaster	28
7.3. Card Addresses	29
7.4. Generating a MASTER EPROM.....	29
7.4.1. Parameter Compiler	29
7.4.2. Structure of a parameter file	30
7.5. Slave-EPROMS	31
7.5.1. EPROM Patch Slaves 9-15	31
7.5.2. KISS-Slave.....	31
8. Installing PC/FlexNet.....	31
8.1. Hard- and Software Requirements.....	32
8.2. Installation and Configuration	32
9. Appendix.....	33
9.1. Protocol version AX25 V2.....	33
9.2. Unproto-Frames	33
9.3. RNR-Behavior/Recovery.....	34
9.4. Reconnects	34
9.5. I-Polling Method.....	34

9.6. Channel Access Method 34
9.7. Layer 2 States..... 35
9.9 Clock options 38
 9.9.1. Operating modes of the SCC 38
 9.9.1.Clock Options 38
9.10. Links to Net/ROM partners 39
9.11. Subscription Service 40
9.12. Correspondence..... 40