

Amicom Version 2.1 documentation

Daniel Pfund

COLLABORATORS

	<i>TITLE :</i> Amicom Version 2.1 documentation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Daniel Pfund	November 28, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Amicom Version 2.1 documentation	1
1.1	Amicom Version 2.1 documentation	1
1.2	Introduction	1
1.3	What is Amicom ?	2
1.4	System requirements	2
1.5	Copyright	2
1.6	Contents of the disk	3
1.7	Packet radio use	4
1.8	What is packet radio ?	4
1.9	What do you need for packet radio ?	5
1.10	What does Amicom do ?	6
1.11	Installation	7
1.12	The 'konfig.ac' configuration file	7
1.13	The contents of the other files	12
1.14	Standard texts	13
1.15	Using Amicom with the WA8DED Hostmode	14
1.16	Using Amicom with the KISS mode	15
1.17	Using Amicom with the Baycom modem	17
1.18	Using the terminal program	18
1.19	Screen setup	18
1.20	Pull down menus	22
1.21	Commands	24
1.22	Description of the TNC commands for Baycom.driv and KISS.driv	26
1.23	Keyboard layout	30
1.24	Remote commands	31
1.25	Data transmission	33
1.26	Different kinds of data transfers	34
1.27	Sending text data	34
1.28	Sending binary data	34
1.29	Special functions	36

1.30 Autosave function	36
1.31 Boxcheck function	36
1.32 Monitor save function	37
1.33 Explanations	38
1.34 Character set converter	38
1.35 PCFont	39
1.36 Sysop mode and password	39
1.37 Explanations concerning the transmission format	40
1.38 Explanations about the monitor contents	42
1.39 For further questions	43

Chapter 1

Amicom Version 2.1 documentation

1.1 Amicom Version 2.1 documentation

```

      / / / /   ##
     / / / /   ###
    / / / /   ####
   / / / /   ## ##
  / / / /   ## ##
 / / / /   #####
\ \ X / /   ## ## ## ## ## ## ##
 \ X / /   ## ## ## ## ## ## ##
  \ X /   ##### ## ## ## ## ## ## 2.1
<----->

```

```

=====
Documentation for Amiga Amicom Version 2.1
=====

```

(C) Copyright 1994 by Gerhard Loder, DL3MGQ

This manual was translated by Daniel Pfund, HB9VBC

```

Introduction
Packet radio use
Installation
Using the terminal program
Data transmission
Special functions
Explanations
For further questions ...

```

1.2 Introduction

```

Introduction
-----

```

What is Amicom ?
System requirements
Copyright
Contents of the disk

1.3 What is Amicom ?

Amicom is a packet radio terminal program for the Commodore Amiga, which is used for the wireless data transmissions of radio amateurs. It works with almost any hardware (modem, TNC, multimode controller) and gives you all the comfortable functions that are necessary in packet radio use.

I would like to thank here all the OMs who helped me to develop this software. A special thanks to Mathias, DG2FEF for the detailed informations concerning DAMA and the beta testing (everyday a new version :-), as well as Daniel, HB9VBC [Hey that's me! :-)] for the English translation of this manual. A big apology to all the OMs who had to bear my tests, which fell, of course, right on prime time, hi.

1.4 System requirements

Amicom needs an Amiga with a minimum of 1 Mb RAM, Kickstart version 1.2, Workbench version 1.3 and one floppy drive. A hard drive is recommended for optimal convenient use. For the connection to the radio, you need one of the following:

- A TNC (TNC2, TNC3, ...) with WA8DED compatible HOST mode. The extended hostmode from DG3DBI is also automatically recognized.
- A TNC or multimode controller (PK232, MFJ 1278, KAM, ...) with a software which can accept KISS mode. This is the case with most multimode controllers.
- A BayCom modem from Johannes, DG3RBU and Florian, DL8MBT. It's a simple modem which finds place on the serial port. A faster processor is necessary to be able to use this modem, an Amiga with a 7Mhz MC68000 will not do!

1.5 Copyright

Amicom can be freely copied and given on from OM to OM, but only in the form of a 1:1 copy of the original disk or the original archives. You are only allowed to give it in the unchanged state and for free, all commercial usage, as well as selling it through PD retailers, is highly prohibited! A distribution through PD networks is allowed but only when there is no fee charged for the connection. All other rights to this software as well as all other files attached to it stay by the author.

The program can only be used for non commercial applications in the radio amateur community, all commercial usage is prohibited. It is not allowed

to give this software for free with other commercial products.
If I hear of any violations to these conditions, I would take immediate legal actions against them!

The use of this software is at one's own risk, I do not take any responsibilities for any damages caused (loss of data, broken serial chips, ...) by the use of this program. I can't give you a guarantee for a total good function of the software.

The original rights to the BayCom modem stay by Johannes Kneip, DG3RBU. The modem can only be used for non commercial purposes in the radio amateur community. Modems made are only for your personal usage. An advertisement for the software (BayCom) or the programming of a software that uses their hardware, has to get prior approval of the authors.

1.6 Contents of the disk

You can find the following files on the original disk or in the original archives:

AmiCom/AmiCom	The terminal program itself.
AmiCom/AmiCom.info	The icon that goes with it.
AmiCom/Anleitung.txt	The german documentation.
AmiCom/Anleitung.txt.info	The icon that goes with it.
AmiCom/BayCom.drv	The driver for the BayCom modem on the serial port.
AmiCom/BayCom.iff	A picture in IFF-ILBM format which shows the electronic diagram for the BayCom modem including the mods for an Amiga usage.
AmiCom/BayCom.iff.info	The icon that goes with it.
AmiCom/Documentation.doc	The English documentation that you are reading right now ;-).
AmiCom/Documentation.doc.info	The icon that goes with it.
AmiCom/greeting.ac	A text file in which a personal greeting message can be stored for each call sign.
AmiCom/help.ac	A text file with the german and English help files.
AmiCom/history.txt	A text file containing the program history for the different versions.
AmiCom/Host.drv	The driver for the WA8DED hostmode .
AmiCom/KISS.drv	The driver for KISS mode.
AmiCom/KISS.ini	A text file containing the command sequence which must be sent to the TNC or multimode controller to put it into KISS mode when the program starts.
AmiCom/konfig.ac	The main configuration file. The most important informations for the program configuration are contained in this text file.
AmiCom/logbook.ac	This text file is only made if you wish, it contains the logbook entries for all your connections.
AmiCom/names.ac	This text file is automatically used by AmiCom and contains for each call sign a name up to 20 characters long. Here are only included the call signs and location of most of the german,

	as well as some neighboring digipeaters and mailboxes.
AmiCom/password.ac	This text file contains the passwords used to activate the sysop modes.
AmiCom/text.ac	In this text file, you can define up to 26 multi line standard texts. These standard can then be sent by pressing a key. Some standard texts already used by the system are already contained in this file.
Fonts/AmiCom.font	An IBM Character font, so you can represent on your AMIGA those special IBM block graphic drawings and special characters.
Fonts/AmiCom/style	Font information.

1.7 Packet radio use

Packet radio use

What is packet radio ?
 What do you need for packet radio ?
 What does Amicom do ?

1.8 What is packet radio ?

Packet radio is a mean to transfer data (text, images, computer programs, ...) with wireless transmissions between radio amateurs. Transfer rates up to 2400 bit/s transform the digital information into a series of 2 different audible tones and transmits them via the microphone entry. It is very similar to RTTY, for example. Other methods are used for higher speeds. The transmission is established thanks to a special protocol, which is called the "AX.25", an amateur adaptation from the world famous X.25 protocol. With this protocol, you are insured to get an error free transmission because errors are detected and corrected by asking another transmission from the sender. With this protocol you can also use only one channel for multiple users.

The interesting thing about this transmission method is that both stations which are in a packet radio connection, do not need to hear themselves. They can connect through up to 8 other packet radio stations, the data is then always transmitted from one station to the other until it finally reaches destination.

In the early days of packet radio, there were only some individuals talking to each other using this mode. Over time, there came digital relays stations, called digipeaters, which were well placed and had dedicated directional transmissions with other digipeaters (links), mostly on higher amateur bands. Thanks to these digipeaters, it was possible to contact other amateurs with only very little material and power. You can even send news worldwide through mailboxes, or exchange information on all kinds of topics. The research goes on every day, and the experiments of radio amateurs have no limits.

1.9 What do you need for packet radio ?

To be qrv in packet radio, you need, apart from your radio station, some hard and software. You have a lot of possibilities:

1. A TNC

A TNC (Terminal Node Controller) is a small computer with some RAM for the data and an EPROM (Erasable Programmable Read Only Memory) containing the packet radio software, also called firmware. A TNC also contains a modem (MODulator/DEMODulator) which converts the digital signals to AF tones. With the in built serial port, you can establish a connection with a terminal or a computer, with the corresponding software. The way the data is exchanged between the computer and the TNC is determined by the firmware you use. The most common TNC is the TAPR TNC-2 and there exists a lot of software for it. The software is free and can be exchanged very easily.

2. A multimode controller

A multimode controller (MMC) or multiprotocol controller (MPC) is basically the same as a TNC, but it can also be used for other digital modes like RTTY, FAX, CW... A multimode controller contains most often more than one modem (one for each mode) and uses software that is specially written for that hardware. The internals differ from one make to the other and it is simply not possible to exchange the software as easily as with a TNC.

3. A special software solution for a computer with a simple modem. These special software solutions exist for the C64 (Digicom) and the PC (BayCom) for example. BayCom is a really simple modem, which uses the serial interface of the computer. A special software controls the modem so packet radio can be used with it. This implies that the computer does a lot of the work: since the modem is so simple and easy to make, the computer has to do a lot of calculations for it.

TNCs and multimode controllers give you a lot of different possibilities for connecting to your computer. The simplest way is to use the terminal mode which is available on most devices. This way, the transmission is simply represented in normal ASCII text. You can switch between commands and data with special control characters like CTRL-C or ESCAPE. The only job the terminal has to do is to give to the TNC the commands you type on the keyboard and to display on the monitor the text it receives from the TNC. For this job, you simply use a terminal program which was intended to be used with telephone modems. This method is very easy, but not convenient at all. The conducting of multiconnections or the transmission of binary data is not possible without some major problems. Another very common way to talk to the TNC is the WA8DED hostmode. The host here is the computer, since he's the only one having the control of everything. The transmission in WA8DED hostmode uses a special binary format. The data is collected into blocks and is transmitted with some other information, so you can distinguish very clearly between channels as well as data, commands, answers, and errors. To be able to use the WA8DED hostmode, you need a specific software on the computer; a simple terminal program is not enough. On the other hand, using such a software is much more comfortable because special amateur radio needs are incorporated into it. These softs often allow you to send data on more than one channel. On a special channel, the monitor, you can always see what is happening on the QRG, etc... Most TNCs or multimode controllers also support the KISS mode

because it is a very simple mode; so it is very easy to implement it. In the KISS mode, the only thing the TNC still does is use it's modem capabilities. The received frames are sent almost untouched to the computer, which also has to send the frames ready to transmit to the TNC. The whole AX.25 protocol is managed by the computer. That's what makes programming the KISS mode so complicating. Even though, it has the advantage that you can also use higher level protocols like TCP/IP.

1.10 What does Amicom do ?

Amicom is a special terminal program for packet radio. Amicom doesn't use one specific hardware configuration because it consists of two different parts: the terminal program itself, and a driver to communicate with the TNC, multimode controller or even specific hardware (like a BayCom modem). Because you are able to exchange these drivers, the program is able to use a lot of different hardware configurations. The drivers that exist for Amicom up to now are:

Host.drv	A driver that uses a TNC with WA8DED hostmode on the serial port. It works with all TNCs who have a WA8DED or NORD<>LINK firmware, or which are compatible with one of these. Amicom recognizes automatically if the firmware uses the extended hostmode from DG3DBI, and makes effective use of it.
KISS.drv	A driver that uses TNCs or multimode controllers which support KISS mode. KISS mode is available on almost any device, but has some disadvantages. If the TNC can also use WA8DED hostmode, it should be preferred.
BayCom.drv	A driver for the BayCom modem (developed by DG3RBU and DL8MBT) connected to the serial port. For the moment, only 1200 Baud is supported, for use with a TCM3105. In order to supply the modem with Amiga's 12V, you need to make a small modification to the original modem (1 diode and 1 resistance).

Amicom offers a really comfortable user interface with all the functions you need for a comfortable packet radio usage. You can define up to 255 separate different channels. You can have a separate and independent connection on each channel; each connection being able to have a two way data transfer. There is a separate write and reception window with numerous editing functions for each channel. You can switch comfortably between the channels with the function keys. There is a separate window for constant monitoring of the frequency. You can scroll back ANY window and the scroll back size is only limited by your RAM size. Other stations can control Amicom remotely with special remote commands (which start with a "//"). This way, you can start a file transfer remotely from another station, even if the OM isn't there. Amicom uses a data file containing 20 characters for each call sign, which are displayed when you're connected to them and can be sent to them when the station connects. This file contains in the base version most of the german and some neighboring digipeaters and mailboxes. In another text file, you can define up to 26 multi line standard texts which can simply be sent by the press of a key. These standard texts also support some simple macros like date, time, name, callsign, etc... Some standard texts are used by Amicom for special functions. For example when someone connects you, Amicom can send (if you

wish) a greeting message. 7plus binary coded files can be stored right at the reception with the correct filename to a drawer you previously defined. A boxcheck function can alphabetically order by title the messages from your mailbox and you can simply retrieve them with a click of the mouse.

1.11 Installation

Installing Amicom is real easy. The main program and all the files that go with it have to be put in the same directory, that you can name as you like. When you start the program from cli, you should type first "CD <directory>" then type Amicom. When you start the program from the Workbench, this is done automatically. But before you start the program, you must absolutely make some changes to the konfig.ac file!

```
The 'konfig.ac' configuration file
The contents of the other files
Standard texts
Using Amicom with the WA8DED Hostmode
Using Amicom with the KISS mode
Using Amicom with the Baycom modem
```

1.12 The 'konfig.ac' configuration file

The "konfig.ac" text file contains all the most important informations and settings Amicom uses. Each line contains a keyword, which has a special function, with one or more parameters separated by spaces. The parameter contains either a number (e.g., 2, 100), a text chain (e.g. "test", "df0:") or a switch (ON or OFF). Amicom ignores lines with a semi colon (;) in front of them.

The following keywords are used:

AMDATE <switch>

AMDATE ON Shows the date in the American format: MM/DD/YYYY

AMDATE OFF Shows the date in the European format: DD.MM.YYYY

AUTOSAVE <switch>

Turns the autosave function on or off. If autosave is on, all the received 7plus files will be saved with the correct name to the AUTOSAVE defined directory.

AUTOSAVEDIR "<path>"

Tells Amicom where he should automatically save the 7plus data he receives when AUTOSAVE is on. The 7plus filename will be appended to the AUTOSAVEDIR, so it has to end either with a "/" or a ":".

Example:

AUTOSAVEDIR ":AmiCom/Bin/"

BAUDRATE <value>

Tells Amicom what baud rate to use between the computer and the TNC. This value will be given over to the driver (e.g. "Host.driv") which will take over the serial connection himself. The serial connection always uses 8 Bit, 1 stop bit and no parity (8N1). You have to put your TNC or multimode controller to these settings (most are already fixed this way, so you don't have to touch anything).

BELL <switch>

Turns the bell (Big Ben bell) on or off when someone connects you or when a connected stations rings you with the remote command "//RI".

BOXCHECK <switch>

Turns the boxcheck function on or off.

BUFFER <channel number> <size>

Sets the size of the scroll back buffer for a specific channel. The bigger this size, the more you can scroll back. But the program gets slower, with the size of the RAM you use. Channel number 0 is the monitor, the size is given in kilobytes (1024 bytes).

Example:

BUFFER 0 10; the size of the monitor scroll back is 10*1024 bytes

BUFFERS <size>

Like BUFFER but for all channels. If you first set a specific channel value with BUFFER, it will be overwritten with BUFFERS, so first set BUFFERS then specific channels size with BUFFER.

CHANNELS <number>

Tells Amicom how many channels you want. CHANNELS can take any value between 1 and 255, but each channel uses memory and slows down the program. A useful value for CHANNELS is 6, because you can just show that many calls in the status line 1. When using WA8DED hostmode, you can only set CHANNELS to the maximum number of channels your firmware can handle. Amicom tells the TNC the value of CHANNELS with the command TNC (Y <number>) so it doesn't take too many connects.

CONVERTER <number>

Tells Amicom which character set converter he should use.

CONVERTER 0 There is no character conversion used apart from the conversion of End Of Lines (LF to CR). It is the equivalent of the ANSI character set of the Amiga.

CONVERTER 1 The Amiga character set is changed to the IBM character set before transmitting and the other way round for reception. This concerns mainly the special national characters like for example éèàçöäüßçø\$ μ \$ degree .

CONVERTER 2 Like 1, but the german "Umlauts" (öäüßÖÄÜ) are converted to the DIN character set ([~|{}).

CTEXT <switch>

Turns the connect text on or off. If it is on, then a connecting station will receive the standard text "C" which can be a small greeting or station information.

DEVICE "<name>"

Tells Amicom which serial communication device it should use with the TNC, normally this is "serial.device", but for multi serial cards you can change

this to another device.

DRIVER "<name>"

Tells Amicom which driver it should use for the communication with the TNC or for controlling the hardware (modem). Up to date, these drivers are:

BayCom.drv Driver for the BayCom modem from DG3RBU and DL8MBT for the serial port.
Host.drv Driver for using a TNC with WA8DED or compatible hostmode firmware.
KISS.drv Driver for all TNCs or multimode controllers with KISS mode.

ECHO <switch>

If ECHO is on, all the text you type on the type-in window will be echoed to the receiving window. The echoed text has another color to be able to differentiate the two type of texts.

EDITWINDOW <number of lines>

Sets the size of the type-in window. This size can always be changed when the program is working by clicking the left mouse button on the status line 2 and moving up or down with the mouse. The size of the scroll back buffer cannot be changed, it is fixed to 40 lines.

EDITOR "<name>"

Here you can define your preferred text editor which can be called by a command or by the pull down menu. The editor must have it's own window, since it is called with the command "RUN >NIL: <Name>".

Example:

EDITOR ":c/MicroEmacs"

END

The configuration file is finished after END.

ENGLISH <switch>

ENGLISH ON All remote texts (answers, errors) are in English.

ENGLISH OFF The texts are in german.

INFOTIME <value>

Sets the value of the infotime in 1/100 of seconds. The infotimer has the following function:

The text you entered is not sent right away to the TNC, but the program waits some time before doing it, this time is defined by INFOTIME. If you type more text before this time is up, then this other text is collected with the first and they are sent together. This function avoids that a lot of small frames are sent out and it gives you some higher data transfer rate.

INTERLACE <switch>

If INTERLACE is set on, the screen will open in interlaced mode, that gives you twice the vertical size of a normal screen.

LOGBOOK <switch>

Amicom will conduct an automatic logbook if you set this function on. It will save the information to the file "logbook.ac". It saves automatically the date, start and end time of the connections.

MAXFRAME <value>

MAXFRAME sets the maximal number of frames that Amicom can send which have not yet been acknowledged. This value can be between 1 and 7. If Amicom already sent out MAXFRAME data frames, then he will not give any more data to the TNC. The value given here is also sent to the TNC with the command (O <value>).

MHEARD <switch>

Amicom will keep a list of the 30 last stations heard if this function is set on. This list can then be seen by a pull down menu (or keyboard shortcut) and even be asked from another station with a remote command .

MONITORWINDOW <number of lines>

This sets the size 1 of the monitor window. This size can always be changed after you started the program by clicking on the status line 3 with the left mouse button and moving up or down. The size 2 of the monitor window is always set to 0 and both sizes can be switched in permanence by pressing SHIFT+HELP.

MYCALL "<Call>"

Sets your own callsign, the most important information! Amicom sends this info to the TNC (I <Call>).

Example:

MYCALL "HB9VBC"

MYNAME "<name>"

Sets your own name, important for the converse mode.

Example:

MYNAME "Daniel"

NUMPAD <switch>

NUMPAD ON The numeric keypad is used to control some of Amicom's special functions (Home/End/PgUp/PgDn/ScrlL...).

NUMPAD OFF The numeric keypad is used for numbers.

Amicom can also be used without numeric keypad, which is especially important for Amiga 600 users...

PACLEN <value>

Sets the maximum size of data blocks in an Info frame. PACLEN accepts values between 16 and 256.

PALETTE <color register> <red value> <green value> <blue value>

You can set the screen colors with PALETTE. The red, green, and blue values can be set between 0 and 15. <color register> can be one of the following:

0=background

1=text color 1 (self written text, status line, pull down menus ...)

2=text color 2 (output text in the output window, monitor heard line ...)

3=background color 2 (background of the status lines)

Example:

PALETTE 0 0 0 0 ;background=black

PALETTE 1 15 15 0 ;text color 1=yellow

PALETTE 2 0 15 0 ;text color 2=green

PALETTE 3 8 8 8 ;background color 2=grey

PCFONT <switch>

If PCFONT is on, Amicom will use the IBM character set (Amicom.font) instead of the normal Amiga character set (topaz.font). Since this is not

compatible with the character set converter, is it only used for reception, for example of representations of IBM block graphics in drawings, tables, etc... A permanent usage by setting it on in this configuration file is actually not meaningful.

REMOTE <switch>

A connecting station can issue special remote commands (starting with a "//" at the beginning of the line) if this function is set on, and thus remote control your station.

REMOTEDIR "<path>"

Sets the remote directory. The connecting station can have read/write access to this directory as well as all others underneath it if REMOTE is set on.

Example:

```
REMOTEDIR ":AmiCom/Remote/"
```

SCREENHEIGHT <value>

Sets the screen height that Amicom opens when the program starts. The width is always 640 points. "SCREENHEIGHT 0" chooses the normal workbench size: 256/512 with PAL Amigas and 200/400 with NTSC Amigas (INTERLACE OFF/ON). [For the AGA Amigas, you can use the DblPAL/NTSC modes by setting this to 0, INTERLACE ON and use a commodity that forces the OpenScreen() routine to open in DblPAL/NTSC like ForceMonitor does.]

SOUND <switch>

If this function is set on, all received data will be signaled by an acoustic click. With the different tones it is possible to hear if the data is received on the current channel (lower tone) or another channel (higher tone). This also determines if a <BELL> character (CTRL-G, ASCII code 7) will be sent to the speaker or not.

TNC <channel number> "<command>"

With this keyword, you can automatically send upon program start a command to the TNC for example to initialize it. Important commands should always be sent on channel 0, since after a disconnect, all other channels are set to the values of channel 0.

Example:

```
TNC 0 "M UISC" ;Turns the monitor on.
```

TNCCLOCK <switch>

Amicom will send the time with the command "K" to the TNC when this function is on, e.g.: "K 23.03.1994", "K 17:00:10".

If your TNC firmware doesn't have a clock (for example like the original firmware from WA8DED), then you should disable this function, since you would only get an error.

TNCSTOP <channel number> "<command>"

Like TNC, but the command will be sent to the TNC when you quit the program.

Example:

```
TNCSTOP 0 "M N" ;Turns the monitor off again.
```

UMLAUTE <switch>

UMLAUTE ON The german "Umlauts" (öäüßÖÄÜ) can be typed in the type-in window.

UMLAUTE OFF The german "Umlauts" are not allowed and are transformed the

following way:

ö->oe, ä->ae, ü->ue, ß->ss, Ö->Oe, Ä->Ae, Ü->Ue

UNIT <number>

Sets the number of the serial unit used to communicate between the Amiga and the TNC. With the built in serial port of the Amiga (serial.device) this unit will always be 0, but for multi serial cards it is possible to choose the serial port you wish with this function.

1.13 The contents of the other files

Apart from the main program (Amicom, the terminal) and the drivers (#?.drv), Amicom uses also the following files:

greeting.ac

In this file, you can set for each call a personal greeting. It will then be used in a standard text, like for example in a connect text, when the macro "%b" is used.

The file "greeting.ac" has the following structure:

```
-----
;you can put a comment with a semi colon at the beginning of the line
#DL3MGQ
This is the greeting for DL3MGQ and it goes on until the next rhombus(#)
at the beginning of a line.
#HB9VBC
This is the greeting for HB9VBC .
#
-----
```

help.ac

This file contains all the help texts for the remote commands "//H" in german and English. It doesn't need to be touched.

KISS.ini

This file is opened by KISS.drv and contains the commands that are necessary to put the TNC or multimode controller into KISS mode.

konfig.ac

This is the main configuration file for Amicom, the content has already been discussed.

logbook.ac

This file is optionally used by Amicom. It contains the history of all the connections you make.

names.ac

This file is automatically used by Amicom and contains for each call, also with SSID, a name with up to 20 characters. Unknown stations can register their name with the remote command "//N". This file also contains the type of station (user, mailbox or digipeater). The type should absolutely be correct since it is very important for some functions.

The file "names.ac" has the following structure:

<call, optional SSID> <type, 0...2> <name, up to 20 characters long>

If there is no SSID present for a call sign (ANYTHING), the entry will apply to all SSIDs. If there is a SSID present (ANYTHING-1), then the connecting station must have this SSID. For this reason, when there are multiple entries for a call, calls with SSIDs should be put before calls without them.

The type of the station can take the following values:

0 = a terminal, a user, a normal station.

1 = a bbs, a mailbox.

2 = a digipeater.

Example:

```
-----
DB0MWS-8 2 BBS München West
DB0MWS-9 1 TCP/IP München West
DB0MWS 1 Digi München West
HB9IAP-8 1 BBS Geneva
HB9IAP 2 Digi Geneva
HB9VBC 0 Daniel
-----
```

password.ac

This file contains the password needed to switch to the sysop mode by remote, it's structure will be discussed later.

text.ac

You can define up to 26 standard texts (A-Z) in this file. These standard texts can then be sent by pressing a combination of keys.

1.14 Standard texts

You can define up to 26 standard texts (A-Z) in the file "text.ac". These standard texts can then be sent by pressing a combination of keys. Some texts are used by the program for special functions:

Text C: The connect text. This text is sent when you are connected from someone. It can contain for example a greeting or a short station description.

Text I: The info text. It contains the station description and can be asked for by the connecting station with the remote command "//I".

Text N: This text can be asked for with the remote command "//N" and contains for example important news (/News).

Text P: The prompt. It is sent after some remote commands .

Text Q: The quit text. It is sent at the end of a connection when the connection station issues the "//Q" command, and can contain a goodbye message.

The following macros can be used in the standard texts, which of course will be replaced by Amicom with the actual values:

%b The file "greeting.ac" is looked up for a personal greeting of the connecting station and sent. It is used in the connect text (text "C").

%c The callsign of the connecting station.

%d The date. It will either be sent in the European (DD.MM.YYYY) or the American (MM/DD/YYYY) format, this depends on the setting in the konfig.ac file.

%m Your own callsign which was used for MYCALL.

%n The name of the connecting station. If the name is unknown, so it is not in the "names.ac" file, then the call sign is used (like %c).

%p The channel number on which the station is connected.

%v The version of Amicom, to date "2.1".

%z The time if the "HH:MM:SS" format.

%; If the name of the connecting station is known, that is, he is registered in the file "names.ac", then this macro is treated as a comment (;) and ignored.
If the name is unknown, then the line starting with a "%?" is sent.
You can for instance ask the connecting station to register his name with the remote command "//N" by using this macro in the connect text.

%% The character "%" itself.

The file "text.ac" has the following structure:

```
-----
;Comments can be put using a semicolon at the beginning of the line.
#A
This is text A which goes until the next line that starts with a rhombus(#)
#B
This is text B.
It goes on more than one line.
;And here comes the connect text:
#C
Hello %n, welcome to the station operated by %m.
%b
;%?Please enter your name with the command '//N <name>'.
#
-----
```

1.15 Using Amicom with the WA8DED Hostmode

Since Amicom is basically prepared to be used with a TNC with WA8DED Hostmode, it is not necessary to make major modifications to make it work. Just edit the file konfig.ac with your favorite text editor (ex: ED) and make the following changes:

First set your own call sign and name, for example:

```
MYCALL "HB9VBC"
MYNAME "Daniel"
```

Then set the driver you wish to use, for hostmode it's "Host.drv". The keyword DRIVER has to be changed to:

```
DRIVER "Host.drv"
```

The extended hostmode from DG3DBI is automatically recognized and used. Amicom also needs some paths to know where to put the data. If Amicom is directly in the main directory of a diskette or a Hard Disk, you can leave the following entries untouched:

```
REMOTEDIR ":AmiCom/Remote/"
AUTOSAVEDIR ":AmiCom/Bin/"
```

It's important that the paths always end with a "/" or ":", because Amicom just adds the file name to this.

If you want to use a text editor with the pull down menus or a command, then you need to tell Amicom which one and where it is:

```
EDITOR ":Utilities/MicroEmacs"
```

If your TNC firmware has no clock (command "K"), then you should change the keyword TNCLOCK to the following:

```
TNCLOCK OFF
```

If TNCLOCK is on, Amicom tries to set it with the command "K". This only gives you an error message if your TNC doesn't have a clock.

If your TNC is connected to the in built serial port of the Amiga, then you have to use:

```
DEVICE "serial.device"
```

```
UNIT 0
```

This should be changed if you use an multi serial card.

The baud rate of the serial port have to be the same for the computer and the TNC. If your TNC works at 9600 Baud (serial), less is not recommended for Hostmode, you should set this:

```
BAUDRATE 9600
```

The TNC has to work with 8 Bit data, 1 stop bit, and no parity (8N1).

If you use a firmware from WA8DED (e.g.: TNC-Firmware version 2.6), then the TNC expects to receive the frack time in seconds and not 1/100th of seconds like other version (e.g.: NORD><LINK). The parameter "F" should be changed from "F 800" to "F 8".

```
TNC 0 "F 8"
```

The other parameters should only be changed if you know exactly what they do!

You can also put some personal greetings, etc... in the files greeting.ac and text.ac , but this is not necessary for a first test.

1.16 Using Amicom with the KISS mode

Extensive changes are necessary to use Amicom with the KISS mode. First, open the file konfig.ac with your favorite text editor (ex: ED) and make the following changes:

First set your own call sign and name, for example:

```
MYCALL "HB9VBC"
```

```
MYNAME "Daniel"
```

Then set the driver you wish to use, for KISS mode it's "KISS.drv". The keyword DRIVER has to be changed to:

```
DRIVER "KISS.drv"
```

Amicom also needs some paths to know where to put the data. If Amicom is directly in the main directory of a diskette or a Hard Disk, you can leave the following entries untouched:

```
REMOTEDIR ":AmiCom/Remote/"
```

```
AUTOSAVEDIR ":AmiCom/Bin/"
```

It's important that the paths always end with a "/" or ":", because Amicom just adds the file name to this.

If you want to use a text editor with the pull down menus or a command, then you need to tell Amicom which one and where it is:

```
EDITOR ":Utilities/MicroEmacs"
```

If your TNC firmware has no clock (command "K"), then you should change the keyword TNCLOCK to the following:

```
TNCLOCK OFF
```

If TNCLOCK is on, Amicom tries to set it with the command "K". This only gives you an error message if your TNC doesn't have a clock.

If your TNC is connected to the in built serial port of the Amiga, then you have to use:

```
DEVICE "serial.device"
UNIT 0
```

This should be changed if you use an multi serial card.

The baud rate of the serial port have to be the same for the computer and the TNC. If your TNC works at 9600 Baud (serial), you should set this:

```
BAUDRATE 9600
```

The TNC has to work with 8 Bit data, 1 stop bit, and no parity (8N1), otherwise Amicom can't put the TNC into KISS mode. KISS mode always uses 8N1.

KISS mode has some disadvantages: the software which makes the protocol (AX.25 L2) is not informed of the status of the DCD (if the frequency is occupied or not). It also cannot check if the TNC has already sent out the frames or not (because the frequency is occupied for example). For this reason, it is very important to set the timer 2 (parameter @T2) correctly. This timer sets the time that the computer should wait before sending out an acknowledgement (RR0v, REJ0v, RNR0v) after reception of a packet (I00^ data). If it is set too small, then each received packet will be acknowledged alone. If it is set too big, then the received packets are acknowledged too late and the corresponding station sends them again for nothing. For 1200 Baud packet, the optimal time was found to be 2 seconds, so the following has to be changed:

```
TNC 0 "@T2 200"
```

This value has to be set smaller, for higher baud rates on the radio side (e.g.: 9600) and be checked during a data transfer with the monitor.

The other parameters should only be changed if you know exactly what they do!

Amicom, respectively KISS.driv sets the TNC or multimode controller to KISS mode on program start, and sets KISS off when it's ended. You have to tell Amicom how your TNC behaves in order to be able to do this, that's why there is the file "KISS.ini". The structure of this file is simple: Like always, a line starting with a semicolon (;) is ignored, so you can leave comments or lines you don't need any more. All other lines contain the ASCII codes of the characters that have to be sent to the TNC.

Example:

To switch to KISS mode with a TNC with TAPR software, you would use:
<press CTRL-C> KISS ON <press Return> RESTART <press Return>

That would correspond to the following line in KISS.ini:

```
3 75 73 83 83 32 79 78 13 82 69 83 84 65 82 84 13
^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^
| | | | | | | | | | | | | | | |
| K I S S | O N | R E S T A R T +-RETURN
+-CTRL-C      +-space +-RETURN
```

In the file are already included some entries for different TNCs, some which weren't tested by myself. So all you need to do is find the right line, take the semicolon away and leave all the others with semicolons.

You can also put some personal greetings, etc... in the files greeting.ac and text.ac , but this is not necessary for a first test.

1.17 Using Amicom with the Baycom modem

Amicom can be used with the Baycom modem from Johannes, DG3RBU and Florian, DL8MBT since the version 2.0. For the time, only modems with 1200 Baud and which use the serial port can be used, for example the plug in modem with TCM3105. The current is normally taken from the TXD, DTR and RTS pins, but this won't work on an Amiga. So you have to use an external current supply or modify the modem with 1 diode and 1 resistor so it can take the current from pin 9 of the Amiga's serial port (+12V). Even if you make this change, the modem stays completely compatible to the P.C. version and you can still use it on a P.C. You can find the file "Baycom.iff" on the original Amicom disk or in the original archive. It contains the schematics of the modem, as well as the modifications for the Amiga. But those who think they can finally sell their TNC should read on. The Baycom modem is a simple modem which transforms the received AF signal (1200 and 2200 Hz) into respective digital HI/LO levels. When sending, the modem makes the correct tones respectively to the level on the serial port and sets the PTT of the transceiver. That means that the software has to control bit by bit the modem and that the modem has no intelligence like a TNC. This is achieved by special interrupt functions which have to be called quite extensively, which is of course very straining for the multitasking system of the Amiga. You also need a very exact timing since the interrupts need to be exactly at the right time and very regular. This is exactly the problem. I have found out that some graphic routines disable the interrupts a short time. This is especially noticeable with the ScrollRaster() function which is used to scroll the screen contents. I have no idea why it is like this. The time the interrupts are disabled depends of the processor you use and not of the use of the Blitter like one could think. That means that you can only use a Baycom modem with a fast processor. It works perfectly on my Amiga with a 25 MHz MC68030 and even multitasking is still possible. With a 7 MHz MC68000, you will always get receive errors (Reject) when the text is output to the screen, because there is no CPU time left. An Amiga 1200 with a 14 MHz MC68020 will work problem free. But if you use Amicom with other software, you could have some problems. Like stated, the interrupts have to stay free at all times. That's why all programs which disable the interrupts even a little while, will give transmission errors. An example for such programs are old hard disk drivers which disable the interrupts when they access the hard drive. To see if the interrupts are disabled, you can use the function "@DEBUG". When you click on the speaker symbol, the "Baycom.driv" will output a click on audio channel 0 at each interrupt. You should get a constant whistle. Important is the tone when you don't receive a packet (DCD off). Is this tone distorted or even discontinued, then a problem free use with a Baycom modem is not possible. Baycom.driv also uses the timer A in CIA B which is only used for data transfer of the serial port, so it is normally free. Sometimes it is still in use from another application or a program that used the serial port before, this can be fixed by doing a reset. But there are also some programs which just simply take this interrupt without checking if it's free, that's the case of some sound module players. Using such programs with Baycom.driv gives you a system crash. Recognition of the DCD, so if the QRG is free or not, is achieved by checking the modem's data throughput. Because the TCM3105 in the modem also reacts to noise, you have to use the squelch on your transceiver. It is recommended to build a

"digital squelch" with an XR2211 because the radio squelch is usually so slow and is the main culprit of reception errors and collisions. The modem will then only react to the AF tones, so your squelch can stay open. The schematics for this are also contained in the file "Baycom.iff".

To use Amicom with a Baycom modem, some changes are necessary. First, open the file konfig.ac with your favorite text editor (ex: ED) and make the following changes:

First set your own call sign and name, for example:

```
MYCALL "HB9VBC"
```

```
MYNAME "Daniel"
```

Then set the driver you wish to use, for the Baycom modem it's "Baycom.drv". The keyword DRIVER has to be changed to:

```
DRIVER "Baycom.drv"
```

Amicom also needs some paths to know where to put the data. If Amicom is directly in the main directory of a diskette or a Hard Disk, you can leave the following entries untouched:

```
REMOTEDIR ":AmiCom/Remote/"
```

```
AUTOSAVEDIR ":AmiCom/Bin/"
```

It's important that the paths always end with a "/" or ":", because Amicom just adds the file name to this.

If you want to use a text editor with the pull down menus or a command, then you need to tell Amicom which one and where it is:

```
EDITOR ":Utilities/MicroEmacs"
```

The entries for the serial port (device, baudrate) are not important when using a Baycom modem and are just ignored.

The other parameters should only be changed if you know exactly what they do!

You can also put some personal greetings, etc... in the files greeting.ac and text.ac , but this is not necessary for a first test.

1.18 Using the terminal program

Using the terminal program

Screen setup

Pull down menus

Commands

Description of the TNC commands for Baycom.drv and KISS.drv

Keyboard layout

Remote commands

1.19 Screen setup

Once the program is started, Amicom opens a screen that looks like this:

```
AmiCom 2.1 [01.01.1999 23:59] [ ][ ]
```

```
-----
1:discon      2:discon      3:discon      4:discon      5:discon      6:discon
-----
```

```
-----
1 T:discon                                NO  UO  TO  discon.                1AB
-----
```

```
-----
MH:
-----
```

On the top line, you can see the screen title line (Amicom 2.1...) in which there is the actual date and time. Then follows the status line 1 (1:discon...), the type in window, the status line 2 (1 T:discon...), the output window, the status line 3 (MH:...) and finally the monitor window.

Status line 1

This line contains the calls of the first 6 connected stations. The active channel has a frame around it and is displayed beneath the status line 1. If data is received on any other channel, then you will see instead of the double point (:) a bigger than symbol (>). You can switch between channels with the function keys or just click on it with the mouse.

Type in window

The type in window consists of a 40 line buffer of which only a portion is displayed here. All the text you enter on the keyboard is first stored in this buffer so you can edit it freely with the cursor keys. You can just type freely your text, any words that will not fit on the line are just reported to the next line (wordwrap feature). When you have completed 3 lines, they are automatically sent off. If you want to send them off before that, just press the Return key. When you move around with the cursor keys, the lines that belonged together are split and when you press Return, they will be sent one at a time. Pressing Return will then send out the line on which the cursor is currently positioned.

"Del" erases the character underneath the cursor "<-" the character before the cursor and "CTRL+Y" erases the whole line.

If you are at the beginning of a line, you can go into command mode by

pressing "ESCAPE" which will be represented by a "» " at the beginning of the line. Enter your command for the TNC of Amicom and send it off with Return. Amicom can differentiate automatically TNC commands from Amicom commands. The size of the type in window can always be changed by clicking on the status line 2 with the left mouse button (leave the button pushed) and moving up and down until you like the size, then stop pushing the button.

Status line 2

The status line 2 consists of a line like:

```
-----
1 D:DB0ZKA      Digi Augsburg          NO U0 T0  Info Transf.      1AB
-----
<-----1----->|<-----2----->|<-----3----->|<-----4----->
```

1: This is the number of the channel which is currently displayed, the type of station connected and the call (DB0ZKA).

The type can be one of the following:

T: Terminal, a normal user, a normal station.

B: BBS, a mailbox.

D: Digipeater

Not all functions work with each type of station. News with "//1..." can only be sent to terminal stations, the autosave function only works with terminals and mailboxes and finally mailboxes cannot send out remote commands (for security).

2: The name of the station to which you're connected to on this channel. It can be made up of maximum 20 characters and is contained in the file "names.ac". Unknown stations can register their names with the remote command "//N...".

3: The link status of a connection. Is one the following:

N (Not sent): Number of frames not sent but in the TNC buffer.

U (Unack): Number of frames that were sent out but not yet acknowledged.

T (Tries): Number of tries to transmit a frame. If T is constantly bigger than 1, then you have a poor radio connection.

Then follows one of the following texts:

discon. Disconnected, no connection.

Link setup Amicom is trying to establish a connection.

Frame Reject A reject frame was sent (FRMR). This is a big protocol error and terminates most of the time the connection.

Disc. req. Amicom is trying to disconnect.

Info transf. Normal information transfer, the most common thing in a connection.

Rej. sent A reject was sent in order to get the missing data again.

Waiting ack. Waiting on an answer from the other station. If no data is transmitted over a period of time, the connection is tested to see if it's still ok.

Device busy Our station cannot accept more data because the receive buffer is full.

Rem. dev. b. The other station cannot accept more data.

Both dev. b. Both stations are busy.

W.a. & d.b. Waiting ack. & Device busy.

W.a. & r.b. Waiting ack. & Remote device busy.

W.a. & b.b. Waiting ack. & Both devices busy.

R.s. & d.b. Reject sent & Device busy.

R.s. & r.b. Reject sent & Remote device busy.

R.s. & bdb. Reject sent & Both devices busy.

4: Some important status information is shown here:

0-2: The number of the selected character converter chosen.

A: The autosave function is on. If a 7Plus coded binary file is received, Amicom will automatically save it with the right name.

B: The boxcheck function is on. If Amicom can identify if the received text is a list from a DieBox or FBB bbs, it will save this text and sort it to a special buffer.

b: Data has arrived into the boxcheck buffer.

C: This channel is in converse mode. All data received on this channel are automatically resent to the other channels in converse mode.

S: This channel is in sysop mode. The connected station can use the special sysop remote commands .

T: A small connect text was defined and will be sent when a station connects.

After a space, you can also have this:

R: The received data is saved to disk (Read). More info is on the status line 3.

S: Data is sent from disk (Send). More info is on the status line 3.

The character set changer display should absolutely be checked when you transfer for example 7Plus or 7Bit coded binary data. If such data is sent out as text data or saved to disk, then the character converter has (!) to be set off, otherwise the binary data is useless. Because people always forget to set the character converter off, there is a special submenu in the menu "File Send" called "7Plus". If data is sent out this way, only the end of line character is changed (LF->CR). When receiving, the character converter is automatically set off.

Output window

All received data and informations from the program are shown in the output window. Following colors and style are used for easier comprehension:

Text color 1: This data was received or output from the program himself and are for the user.

Text color 2: This data came from the user or from the program and were sent off.

Beware: this text is only visible if you set ECHO ON in the file konfig.ac.

Normal: This is normal received or typed in text.

Bold: This text came from the program, for example error messages or output to a remote command.

Cursive: This text came from the TNC or the driver, for example status display.

You can scroll back/fwd in the output window with the keys "PgUp", "PgDn", respectively "shift+cursor up" and "shift+cursor down". With "Home" and "End", respectively "ctrl+cursor up" and "ctrl+cursor down" you can go directly to the buffer beginning/end. How far you can scroll depends on your BUFFERS setting. While you scroll back, the TNC is not emptied, so always go back to the buffer end with "End".

If you click separate words in the output window, they'll appear in the type in window.

Status line 3

In normal use, Amicom will display in the status line 3 up to 11 calls which were heard on the frequency. If 11 calls have already been heard and a 12th is heard, then the one which hasn't been heard for the longest time is replaced by the new one. A call is taken off the list if it wasn't heard in the last 5 minutes. With this function, you can always see which stations are actually QRV on the frequency.

During a data transfer, another representation appears. If the total length of the data is known (send, receive with autobin or autosave of a 7Plus data), you will get a line like:

```
-----
R7 TEST.7PL 3450/9876 (34%), t=5 min.
-----
```

```
<----1---->|<---2--->|<-3->|<---4--->
```

- 1: R: Amicom is receiving and writing on (hard)disk (read).
 S: Amicom is sending data from (hard)disk.
 T: Transfer of a text file with character and EOL conversion (LF<->CR).
 B: Transfer of a binary file without character conversion.
 A: Transfer in automode (autobin).
 7: Reception of a 7Plus file with autosave or transmitting without character conversion; only EOL conversion .
 The file name follows after that without path (TEST.7PL).
 2: Number of bytes already transferred and total number (3450 from 9876)
 3: Number of bytes already transferred in percent of total (0-100)
 4: Time still needed if the transfer stays at this rate. This time is only an approximation and cannot be held for true because the data transfer rate normally always fall.

Points 3,4 and the total size will be missing if the total size of the data is unknown.

Monitor window

On the bottom of the screen you can see the monitor window. You can constantly monitor the activity of the QRG in this window. You can switch to whole screen with the HELP key. Once in the whole monitor screen, you can also send out unnumbered info on the frequency (CQ calls, etc...), or see the scroll back monitor buffer similarly to the output window scroll back buffer. If you press HELP again, you will switch back to the last active QSO channel. You can choose between two monitor window sizes with shift+HELP. You can also set the monitor window size by pulling on the status line 3 with the mouse.

1.20 Pull down menus

Most program functions are called by pull down menus. Here is a list of all the functions. The shortcuts which can be used with the right Amiga key are listed in parentheses.

Program menu:

- Info Shows a dialog box with a copyright info and the size of your free RAM.
 Editor (E) Calls the specified editor in konfig.ac with

"RUN >NIL: <name>".

Shell (S) Opens a new shell with "RUN >NIL: NewShell".

Quit (Q) Quits the program.

File menu:

Read file The received data can be saved to (hard)disk this way. You can only read one file at a time. While reading, the autosave function is off.

Send file Sends a file from a (hard)disk.

Stop reading The file which Amicom was saving to, is closed.

Stop sending Stops the transmission of a file.

Delete file Deletes a file name, like the command "Delete" in shell.

Function menu:

Connect (C) A call sign is asked for. Amicom attempts to connect this call.

Disconnect (D) The connection is ended. Please wait until the connection is really finished!

C. Message (T) You can enter a short message that will be sent upon connection. (max 80 chars, e.g.: "Moment pse, just have a 600 ohm call, "be right back"...)

Undo (U) Amicom shows all the received messages of the type "Connected to..." or "Reconnected to..." and actualizes the logbook. The status line will also contain the actual call. If for example you should receive a "false message" from a BBS, you can undo it with this command. The logbook will be changed too.

Boxcheck/Show (B) Shows the contents of the boxcheck buffer. You can leave this mode by pressing ESCAPE.

Boxcheck/Clear The boxcheck buffer is cleared and the memory is freed. This is done automatically when you leave the mailbox.

Boxcheck/Next (N) Sends a read command ("R") for the next chosen message.

Rec. Buffer/Save Saves the contents of the receive buffer (output window) to (hard)disk.

Rec. Buffer/Clear Clears the receive buffer and output window.

Printer/On Everything that is received will be sent to the program's internal 64K printer buffer and printed. This buffer is used because you'll receive the data faster than your printer can print it.

Printer/Off Turns the print function off. The printer buffer is cleared.

Printer/Break Printing is immediately stopped and the printer buffer is cleared.

Standardtext (-) After having chosen this function, or by pressing rightAMIGA+"-", you have to type a key (A-Z) to send out that key's specific standard text.

Monitor menu:

Save/On Starts the monitor save function.

Save/Off Stops this function.

Save/Statistics Shows how many frames have already been saved by the monitor save function.

Mon. H. /Show (M) Shows the big monitor heard list. This list contains up to 30 heard stations.

Mon. Heard/Clear Deletes the big and small monitor heard list.

Switch menu:

Some functions can be turned on/off with this menu. The bold menu points

are only valid for the active channel, the normal style menu points are valid for all channels.

Converter	Choose the character converter (off, IBM or IBM/DIN).
Umlaute	"Umlauts" on/off.
Sysop	Sysop mode on/off.
Convers	Converse mode on/off.
PCfont	Uses the IBM character set (Amicom.font)
Echo	Turns echo on/off. If it is set on, all typed text will also appear in the output window.
Bell	Turns bell on/off when connected or "//RI".
Sound	Turns noise on/off when receiving data or CTRL-G.
Ctext	Sets connect text on/off.
Remote	Sets remote on/off. ("//XXXX" commands on/off)
MHeard	Sets monitor heard list on/off.
AutoSave	Turns the autosave function on/off.
Boxcheck	Turns the boxcheck function on/off.

1.21 Commands

If the cursor is in the type in window at the beginning of a line, you can switch to command mode by typing ESCAPE. The command mode is represented by a "" at the beginning of the line. You can then enter your command for the TNC or the program and send it by pressing RETURN. The differentiation of program or TNC commands is made by Amicom. Here is a list of all the program commands. Which TNC commands can be used depends on the driver you use or of your TNC firmware. A list of commands for KISS.drv or Baycom.drv follows in the next chapter.

Program commands

Each command is a keyword followed by one or more parameters which can be a number, a character chain, or a switch. A switch is made up of the words "EIN", "AUS", "ON" or "OFF". If the character chain contains spaces, it should be set between quotation marks, e.g.: 'df0:data', '"This is a test."'. If the command is sent without a parameter, the actual value of this parameter will be displayed.

List of program commands:

AUTOSAVE <switch>

Turns the autosave function on/off.

BELL <switch>

Turns the bell on/off when someone connects or uses //RI.

BOXCHECK <switch>

Turns the boxcheck function on/off.

C <call, call and path>

Connect, a connection is established to the callsign, eventually via the specified digipeaters. If a connection is already established to that station, Amicom will send out the error message "STATION ALREADY CONNECTED", increments the SSID by MYCALL and tries to connect again.

Example:

```
c hb9iap
c hb9iap-8 hb9iap
c hb9iap-8 via hb9iap
```

CHANNEL <channel number>

CH <channel number>

Switches to the specified channel number. Because you can "only" (?) go the first 30 channels with the function keys, this command is necessary to be able to access the channels 31-255 (for those who need that much...).

CONVERS <switch>

CONV <switch>

Turns converse mode on/off on active channel.

CONVERTER <number>

Chooses the character set converter for the specified channel, <number> can take a value between 0 and 2.

CTEXT <switch>

Turns the connect text on/off.

ECHO <switch>

Turns the echo function on/off.

END

Stops or breaks a data transfer.

FLUSHMHEARD

FMH

Erases the big and small monitor heard lists.

D

Disconnect, the connection on this channel is ended.

DELETE <file name>

DEL <file name>

Erases the specified file name on (hard)disk like the shell command "Delete".

EDITOR

ED

Starts the configured editor.

MAXFRAME <value>

Sets the maximum number of frames that can be sent out without being acknowledged. MAXFRAME can take values between 1 and 7, the set value will be sent to the TNC with the command "O <value>".

MHEARD <switch>

Turns the updating of the big monitor heard list on/off.

NEWSHELL

Starts a new shell.

PACLEN <value>

PAC <value>

Sets the maximum length of data frames. PACLEN accepts values between 16

and 256.

PCFONT <switch>

Turns the font "AmiCom.font" on/off.

PRINTER <switch>

Turns the printer on/off.

RA <file name>

Starts the reception of the file name <file name> in automode (autobin).

RB <file name>

Starts the reception of a text file to disk, the character set converter will be used.

REMOTE <switch>

Turns the remote commands with //XXXX on/off.

SA <file name>

Starts the transmission of the file <file name> in automode.

SB <file name>

Starts the transmission of the file <file name> without the character set converter.

ST <file name>

Starts the transmission of the file <file name> with the character set converter.

SOUND <switch>

Turns the sound by received packets or CTRL-G on/off.

SYSOP <switch>

Turns the sysop mode on/off on this channel.

UMLAUTE <switch>

Turns the umlauts on/off.

1.22 Description of the TNC commands for Baycom.drv and KISS.drv

When using "Host.drv", the TNC commands are only dependant on your TNC firmware since "Host.drv" directly sends the commands unchanged to the TNC. By "KISS.drv" and "BayCom.drv", the TNC commands are treated at the driver's level, just like the whole AX.25 protocol, a part from the status and error reports and the monitor output. The usage is so very similar as to using WA8DED hostmode firmware. If a command is used without parameters, it's actual value will be displayed.

Here is a list of TNC commands for the "BayCom.drv" and "KISS.drv". Those marked with a star (*) are valid for each channel separately. When you disconnect, the values of the channel 0 will be taken again. This is why the most important settings have to be made on channel 0!

List of TNC commands:

B <0...65535> (DAMA time-out in seconds)

This parameter is only used for DAMA activity, it will be automatically detected and used by Amicom. With the parameter "B", you can set the DAMA time-out in seconds, the default setting is 120 (2 minutes). If during this time, Amicom hasn't received a poll from the DAMA master, Amicom will switch back to CSMA usage (normal usage).

C <call> [via] [digipeater digipeater...] (Connect) *

Sets a connection to the defined station. You can also specify a route with up to 8 digipeaters. If a connection is already established to this station on this channel, then the new route will be used instead and the link rebuilt.

On channel 0 (monitor) you can define the unproto (UI frames) destination call with this function (e.g. "C CQ").

Example:

```
c hb9iap
c hb9iap-8 hb9iap
c hb9iap-8 v hb9iap
c hb9iap-8 via hb9iap
```

D (Disconnect) *

Ends the connection on this channel, but before that, Amicom waits until all the sent frames have been acknowledged by the other station. If the disconnect command is sent twice, all the frames are dropped immediately and only the disconnect request is sent. If the disconnect command is sent three times, the connection is immediately terminated without even waiting for the disconnect acknowledgment (UA-) from the other station. This solution should only be used in helpless cases, as the other station maybe hasn't received the end of connection status.

F <1...65535> (timer 1, frack in 1/100 of seconds) *

Sets the value of the timer 1 (frack) in 1/100 of seconds. If a frame has been sent out to the other station and if he hasn't acknowledged during this laps of time, then the same frame will be resent.

I <call> (mycall) *

Sets your own callsign. This setting should absolutely be made on channel 0 because the other channels will take these values after a disconnect.

K <0...2>

Tells Amicom if you wish to stamp (date and time) the received packets:

0 = no stamp.

1 = stamp by status information (CONNECTED TO...).

2 = stamp by status information and monitor heard lines (fm HB9VBC to...).

L (get status) *

Gives the status of the connection on this channel. The status is composed of 6 numbers which have the following meaning:

1. number: The number of not yet displayed status informations.

2. number: The number of not yet shown data frames.

3. number: The number of frames that have not been sent but are in the TNC buffer. This value is also displayed on the status line 2 (N).

4. number: The number of sent data frames which haven't been acknowledged yet. This value is also displayed on the status line 2 (U).

5. number: The number of retries of the last frames. This value is also shown on the status line 2 (T).

6. number: Link status of the connection. It is displayed as text on the status line 2.

M <U|I|S|C|N>

With M you can define what's being displayed in the monitor:

U: unproto frames (UI) are displayed.

I: info frames (I) are displayed.

S: control frames (RR, REJ, RNR, FRMR, SABM, DISC, UA, DM) are displayed.

C: show the monitor even if you're connected.

When using Amicom, you should set "M" to "UISC"!

N <0...65535> (N2, tries) *

Sets the maximum number of retries. If you still don't get an answer from the other station, then the connection is terminated (LINK FAILURE). The value 0 sets this to infinity, which is not very useful...

O <1...7> (maxframe) *

Sets the maximum number of unacknowledged frames. Once so many frames have been sent out, Amicom waits for an acknowledgment from the other station.

P <0...255> (persistence)

Sets the value for the built in random generator. The TNC will never transmit with a value of 0, but will always transmit with a value of 255. This parameter is related with the slot time (W), useful values are between 16 and 128.

Warning: The parameters P and W should only be changed if you know exactly how they work! When I speak with OMs who make using the DIGI impossible, I find out that it's always because they changed these parameters for a reason or another and didn't know what they were doing!

Q [+|-] <call> [search route] (QRV? search function)

With Q it is possible to search for a station, just like the F command by a Flexnet.

Q <call> Searches for the station <call> directly and on the configured search route.

Q <call> <route> Searches for the station <call> on the route <route>. The <route> can contain up to 8 digipeaters. If <route> is specified then the configured one is not used.

Q + <route> Sets the route <route> to the search routes. You can have up to 8 search routes, each containing up to 8 digipeaters.

Q - <route> Takes the <route> out of the search route.

R <0|1> (digipeat) *

Turns the L2 digipeating on/off.

T <0...255> (TXDelay in 1/100 of seconds)

After having pressed the PTT switch of your trx, you have to wait some time before you can really send data at the specified power, on the specified frequency which has to be set by the PLL, until the transmission relays switches, etc... If you didn't wait, some data would get lost. To remedy this, use the parameter "T" which tells the TNC how much time he has to wait after pressing the PTT switch before really sending data. A TXDelay time of 250 ms (T 25) should suffice for most modern transceivers, a time

too long only blocks the frequency for nothing.

U (connect text)

With U you can normally set the connect text on/off. This is done by Amicom and is useless in this case. So you don't always get an error message if the command "U" is present in the config file, it is also implemented here but has no function.

V (Version)

Shows the protocol version used.

W <0...255> (slot time in 1/100 of seconds)

Sets the slot time for the persistence function in 1/100 of seconds. This setting goes together with the "P" parameter, see there for more infos.

X <0|1> (lock transmit)

0: the trx is locked.

1: the trx is free to send.

Y <0...255> (channels)

Sets the maximum number of connects the TNC can take from the outside. If there is already "Y" number of connections, then if another station wants to connect you (SABM+), Amicom will send back a disconnected mode (DM-). This value is automatically set to the number of CHANNELS set in the config file.

"Y" without parameters shows the number of channels set and the number of connections in parentheses.

Shows how much of the buffer is still free. Behind is the maximum value of the buffer in parentheses.

Turns the full duplex on/off. This is only possible if you separated the reception from the transmission and isn't normally used by normal stations but only by dedicated links between digipeaters.

Opens the debug window in which you can see the status of the DCD and PTT as well as if you're in DAMA mode or not. When you click on the speaker symbol, BayCom.driv will send out a click on audio channel 0 with each interrupt. You should hear a constant tone. If the audio is not made up of a constant frequency when you don't receive anything (DCD off) or even if it's broken, then you can't use the BayCom modem without major problems. During reception (DCD on), it's normal to hear a little distortion in the sound, because the phase length always changes so the DPLL (Digital Phase Locked Loop) can synchronize to the receiving signal.

Sets the maximal length of the Info Poll frames (I+).

Before the received frames are acknowledged, Amicom will wait the "@T2" set time to see if another info frame is being received. If the timer has timed out and no other info frame is received, then the acknowledgment (RRv, RNRv, REJv) is sent out. This timer is especially useful in KISS mode, so not each and single frame is acknowledged. By Amicom, the following modification is used: After reception of an acknowledgment (RRv, REJv, RR-...), Amicom will also wait the "@T2" time before sending out another info frame (I^). This avoids unnecessary transmission with the KISS mode.

If you're transferring data (you don't have much to talk about, hi), then Amicom will check after @T3 time of receiving nothing if the connection is still ok.

After sending the last frame, Amicom will wait @TA time before dropping the PTT line. This is sometimes needed by slow modems.

1.23 Keyboard layout

Keyboard layout in normal mode:

F1...F10	Switches between the channels 1-10.
SHIFT+F1...F10	Switches between the channels 11-20.
CTRL+F1...F10	Switches between the channels 21-30.
HELP	Switches to the monitor channel (channel 0). When you press HELP again, you go back to the channel you left (QSO channel).
SHIFT+HELP	Switches between the two alternative monitor sizes at the bottom of the screen.
ESCAPE	If the cursor is at the beginning of a line, then you can enter TNC or program commands by pressing ESCAPE.
DEL	Erases in the type in window the character that is under the cursor.
<-	Erases the character just before the cursor.
CTRL-Y	Erases the complete line in the type in window.
RETURN	Stops the entry on a line and sends the command or the text out to the TNC. Amicom then jumps to the next line.
CURSOR UP	Moves the cursor in the type in window up one line.
CURSOR DOWN	Moves the cursor in the type in window down one line.
CURSOR LEFT	Moves the cursor in the type in window back one character.
CURSOR RIGHT	Moves the cursor in the type in window forward one character.
SHIFT+UP or PgUp	Scrolls back one page in the output window. The output is stopped until you go to the end of the buffer again.
SHIFT+DOWN or PgDn	Scrolls forward one page in the output window.
CTRL+UP or Home	Jumps to the beginning of the buffer.
CTRL+DOWN or End	Jumps to the end of the buffer, incoming text will be displayed again.
ScrL on numeric pad	The output will be stopped until SclL is pressed again, like this you can read the received text.

Keyboard layout in the boxcheck buffer:

F1...F10, SHIFT+F1..F10, CTRL+F1...F10	Will jump to the normal QSO channels and close the boxcheck buffer.
HELP	Switches to the monitor channel (channel 0). When you press HELP again, you go back to the channel you left with the boxcheck buffer.
SHIFT+HELP	Switches between the two alternative monitor sizes

at the bottom of the screen.

ESCAPE Goes back to normal mode. Boxcheck display will be ended.

CURSOR UP Scrolls back one line in the buffer.

CURSOR DOWN Scrolls forward one line in the buffer.

SHIFT+UP or PgUp Scrolls back one page in the buffer.

SHIFT+DOWN or PgDn Scrolls forward one page in the buffer.

CTRL+UP or Home Jumps to buffer start.

CTRL+DOWN or End Jumps to buffer end.

F You can define a search string for which Amicom will search through the boxcheck buffer.

R Goes back to normal operation and sends a read command for all (!) selected messages. It is better to close the boxcheck buffer and read the messages one by one with rightAMIGA+N.

SPACE The top most message is selected. Pressing SPACE again will un mark it. All marked messages can then be read automatically.

1.24 Remote commands

Amicom can be remotely controlled if you activated this feature (REMOTE ON). The remote commands start with a "//" at the beginning of a line, this is to differentiate them from the normal QSO text. Remote commands will be ignored from mailboxes, that's why you have to set the entries in the file "names.ac" correctly!

The following remote commands are supported by Amicom:

//CONV

Starts converse mode on this channel. All what is entered here will be sent to the other channels which are also in converse mode. You can a QSO with more than one station like this. Please note that this function should only be used if your local digipeater doesn't have a converse mode function itself.

//CS

Gives the connect status. This is a list containing all the connected stations with their channel numbers. A line of the connect status could look like this:

Connect Status de HB9VBC, 02.05.1994 14:10:30:

1 > B:HB9IAP-8 via HB9IAC 02.05.1994 14:00:30 BBS Geneva

1|2|3|4-----|5-----|6-----|7-----|8-----|

1: The number of the channel on which the connection is taking place.

2: The direction of the connection:

>: You connected the station (HB9IAP-8).

<: You were connected from the other station.

3: The type of the station. It can have the following values:

T: A terminal, a normal user.

B: A BBS, a mailbox.

- D: A digipeater.
- 4: The call of the station to whom you are connected. If the connection goes through one or more digipeaters, the last digipeater will also be displayed.
- 5: The date of the connection.
- 6: The time of the connection. You can see how long you have been connected like this.
- 7: The name of the station if it's known.
- 8: If the channel is in converse mode, you will also see here "conv".

//DIR [<sub directory>]

//D [<sub directory>]

Shows the directory of the remote directory or a sub directory of it. The other stations have read and write privileges on this remote directory!

//ECHO <text>

//E <text>

Sends the <text> back.

//END

Closes a file you opened with //W, //WT or //WB or brakes the data transfer (//WA, //R, //RT, //RB, //RA).

//HILFE [<command>]

//HELP [<command>]

//H [<command>]

"//H" gives you a list of all the remote commands possible. You can get specific help for a command with "//H <command>".

//INFO

//I

Sends the standard text "I". It normally contains a small station information or presentation.

//MH

Shows the monitor heard list. This list can contain up to 30 stations which were heard in the monitor. It is sorted from the last heard to the first heard.

//NAME [<name>]

//N [<NAME>]

"//N" gives you the name which of the station which is connected on this channel while "// <name>" will set the name corresponding to your call into the file "names.ac".

//NEWS

//NE

Sends the standard text "N".

//PRIV <password>

Changes to sysop mode.

//RT <file name>

//R <file name>

Reads the file <file name> from the remote directory, the character set converter is used. The transmission can be stopped at all times with "//END".

```
//RA <your file> [<my file>]
//RPRG <your file> [<my file>]
Reads the file <your file> from the remote directory in automode (autobin).
If the optional parameter <my file> is specified, Amicom will first send
out a "//WPRG <my file>" so as to start the automode reception at the other
station's computer.
```

```
//RB <file name>
Reads the file <file name> from the remote directory without using the
character set converter and EOL conversion.
```

```
//Q
//BYE
Ends a connection. Before ending, Amicom will send the standard text "Q"
which can contain for example a goodbye message.
```

```
//VER
Sends the version number of the program and a copyright message.
```

```
//WT <file name>
//W <file name>
Opens the file <file name> in the remote directory. You can then enter
text that will be saved to this file, the character converter will be used.
To close the file, you must enter "//END".
```

```
//WA <your file> [<my file>]
//WPRG <your file> [<my file>]
Opens the file <your file> in the remote directory for automode
transmission. If the optional parameter <my file> is specified, Amicom
will first send out a "//RPRG <my file>" so as to start the automode
transmission at the other station's computer.
```

```
//WB <file name>
Opens the file <your file> in the remote directory. You can then enter
text that will be saved to this file, the character converter will not be
used. To close the file, you must enter "//END".
```

In the case you would like to experiment with these commands before going off on the air, you can try these on a non connected channel with yourself. Just go to a non connected channel and issue the remote commands you wish. If the ECHO function is on, you will see the results in the reception window.

1.25 Data transmission

Data transmission

Different kinds of data transfers
Sending text data

Sending binary data

1.26 Different kinds of data transfers

When transferring data over packet radio, you should make a difference between two types: text files and binary data. Text files consist of normal ASCII text which can be directly read and sent. Binary files can consist of workable programs, images or special file formats; they contain all the characters between 0-255. The differentiation has two causes:

- In packet radio, the ASCII code 13 (return character) is used to separate lines. Various computer systems use different line separators or combinations, e.g. linefeed (ASCII code 10) by the Amiga or return+linefeed by MS-DOS. If text files have to be sent, you have to change the line separators to return first. This would make binary files useless because those need ALL the characters from 0-255. Also for non ASCII code normed characters like the German umlauts, it is necessary to transform the characters first.
- A binary file has to be transmitted without any missing bytes, transmission errors have to be detected. A special transmission protocol has been created to avoid that the other station who wants to save the file has to know when it starts and finishes. This protocol is the autobin protocol. If you want to send binary files through your BBS, there exists a lot of programs which will transfer your files to normal ASCII text. The most know in the packet radio network is 7plus. With this program, you can also find and correct transmission errors.

1.27 Sending text data

Text data transmission is the easiest of all data transmissions. You can, with Amicom, write a simple message with an editor and send it directly from (hard)disk. The other station receives the text just like if it had been written normally. All that is received on a channel can also be save directly to (hard)disk, you can then read or print the messages later. If you call the function "File/Read File/Text", you can choose a file name to which all the data will be saved immediately.

Because the Packet Radio network uses the Return character for end of lines (ASCII code 13), Amicom will change all the linefeeds from the Amiga (ASCII code 10) to returns before sending out text, when receiving the contrary applies. If a character set converter is used, e.g. IBM or DIN/IBM, then most special characters or national characters "...") are also changed.

1.28 Sending binary data

For a binary transfer, all the 256 characters are needed. That means that special control characters also have to be used. There is also another problem with binary transfers:

All characters have to be transferred correctly, no character can be missing

and no character should be added. So that the other station knows when the file starts, how long it is, and when it's finished, there is the Autobin protocol. An automode transmission functions somewhat like this:

```
OM A tells OM B that he wants to transmit binary data:
> OM A -> OM B: "I'm going to send you binary data, start saving."
OM B starts the automode reception with the function "File/Read File/Auto".
> OM B -> OM A: "Ok, I'm ready now."
OM A starts the transmission with the function "File/Send File/Auto".
> OM A -> OM B: "#BIN#2400"
(2400 is the length of the file)
> OM B -> OM A: "#OK#"
The software from OM A will start sending after the reception of "#OK#",
and the software from OM B will save everything that he's receiving to
(hard)disk. After having received 2400 bytes, the file will be closed.
```

Remote commands can also be used to transfer files from one station to the other. Example:

```
OM A would like to transfer a binary file to the hard disk of OM B. He
sends the following command:
> OM A -> OM B: "//WA Test"
The software from OM B will probably say that he is ready. This could be
something of the like:
> OM B -> OM A: "<AmiCom>: Ready for automatic file transfer."
OM A starts the file transfer:
> OM A -> OM B: "#BIN#2400"
> OM B -> OM A: "#OK#"
The file transfer goes on. Once the software from OM B has received 2400
bytes, it will close the file and acknowledge the reception.
This could be something of the like:
> OM B -> OM A: "<AmiCom>: File transfer ended."
```

Most stations and mailboxes can only transfer files as text. To bypass this problem, there are several programs which can transfer binary files to text. The most common in the Amateur radio world is 7Plus. A 7Plus file transfer goes like this:

Let's say you want to transfer a file that is called "test". You first have to transform it to a text file with 7Plus. You write the following command in the shell:

```
"7Plus Test"
7Plus will then make a file called "TEST.7PL". If the file is longer,
7Plus will split it up into several parts, e.g. "TEST.P01" and "TEST.P02".
These parts can then simply be transmitted as text files. Once all the
parts are received by the other station, they'll have to enter the
following command in the shell: "7Plus test.7pl" or "7Plus test.p01" and
find the original binary file again. If there is a transmission error, an
error report will be made which can be sent to the originator of the
program so he can send you only the missing part. More information is
given in the 7Plus documentation.
```

When transferring 7Plus coded files as text files, you must (!) disable the character set converter (CONVERTER OFF), otherwise the file will be useless. Because most people simply forget to turn off and then on again the character set converter, Amicom has the function "File/Send File/7Plus". This function sends text without the character set converter, so only the

linefeeds are changed to returns. If the autosave function is on, received 7Plus files will be automatically saved with their right names to the autosave directory.

1.29 Special functions

Special functions

Autosave function
Boxcheck function
Monitor save function

1.30 Autosave function

Most mailboxes are only capable of transferring text files. If you want to transfer binary files, you have to transform them first to text. The most common packet radio binary coder is "7Plus" from Axel Bauda, DG1BBQ. 7Plus will code automatically binary files to small text files. If there was a transmission error somewhere, 7Plus allows you to transfer only the missing/destroyed part. It is actually a very practical way to transfer files, but the saving of the multiple parts is a cumbersome chore. The autosave function of Amicom will do this job for you. With the "AUTOSAVEDIR" keyword in the konfig.ac file, you can define to which directory the files have to be saved to. If Amicom receives a line which starts with " go_7+", " go_text." or " go_info.", it will automatically stop using the character set converter and save the file with the right name (*.7pl, *.pxx, *.err, *.cor or *.inf). It will close the file and start using the character set converter again after having received a " stop_text." or " stop_info.". If the file already exists, Amicom will append to the name a number (e.g. "test.7pl_2") to avoid erasing the old file.

1.31 Boxcheck function

Most mailboxes have a check function which will show you the message titles since your last login. Like this you can check in by regular intervals to see if there is anything new of interest. But the mailboxes usually receive hundreds of messages a day which are listed in their order of reception. Having to look through this list is already a cumbersome task but then you have to send several read commands if you found some messages of interest. This is especially the case with FBB style mailboxes, because the numbers are usually made up of 5 or 6 digits. The boxcheck function of Amicom will make this task much easier for you. If Amicom receives a check or list line, it will automatically put it into a buffer and sort the line by the TO field and by alphabetical order. The contents of this buffer can be seen with the "Function/Boxcheck/Show" function or by pressing Right-Amiga+B. Then you can select the interesting

If you selected the point "Only Info-Frames" and if you selected a fm- and a to- call, you can also select the point "Keep Order". Amicom will then save the frames in the same order as the other station sees them himself without the monitor headers (fm... to... ctrl I00...). You can then save for a specific station his entire QSO, e.g. for passive box checking! This cannot be ensured however since you don't take part in the QSO, you could miss some frames and have some "holes".

If you want to save a text file, you can choose which character set converter to use, [OK] will start the monitor save function and [CANCEL] will stop it.

1.33 Explanations

Explanations

Character set converter

PCFont

Sysop mode and password

Explanations concerning the transmission format

Explanations about the monitor contents

1.34 Character set converter

A computer is only a machine which can handle numbers. So the computer can also display text characters, they are also saved as numbers: each character has a specific number for it, A is 65, B is 66, etc... This serie was defined in the ASCII code (American Standard Code for Information Interchange), but only for the characters from 0-127. These first 128 codes contain all the control codes and characters for English. For the characters from 128-255, no standard has been set and so more than one is used; each computer maker having it's own. This range contains mostly less used graphic characters and special national characters like the German umlauts~.

In the amateur radio community, all kinds of computers are used and so all kinds of different character sets. In order to avoid confusion and problems, Amicom has two conversion tables which can be used before transmitting and when receiving:

CONVERTER 0: The character set converter is set off, only the normal ANSI character set of the Amiga is used. Before sending out data, only the linefeeds are changed to returns for PR usage.

CONVERTER 1: The German umlauts (") and most other national characters "...") as well as control characters (...) are converted to the IBM character set, as long as they are present in both (IBM and Amiga) character sets.

CONVERTER 2: Like 1, but the German umlauts (") are converted to the DIN character set (|{}~\[]). These characters can then not be used any more!

The DIN character set is an old usage from the time when data transmission

were still using only 7 bits. It is still used very much in the packet radio transmissions but it's become absolutely useless since packet radio uses 8 bit transmissions! When using this character set, you have the disadvantage of not being able to use the other characters (|{}~\[]) which are still in use a lot. This way, you get umlauts when there shouldn't be any...

Warning:

When you are transferring ASCII coded binary data (e.g. with 7Plus, 7Bit, UUE...) you absolutely have to turn the character set converter off! A such file which has gone through the character set converter is useless.

1.35 PCFont

Sometimes you may find in the mailboxes special drawings (maps, diagrams, tables) which were drawn using the IBM block graphics characters. To see these correctly on the Amiga, there is the font "AmiCom.font" which consist of an original IBM character set.

You can turn it on/off with the menu "Switch/PCFont" of the command "PCFONT ON/OFF". It's useless to leave it permanently since it doesn't work together with the character set converter and only when receiving.

1.36 Sysop mode and password

If the remote command is on (REMOTE ON), every station can issue remote commands , but not all commands are allowed by all stations, some are reserved to the sysop of the station. If you want to issue them, you have to enter the sysop mode first. This is done by using a function from a pull down menu or the remote command "//PRIV". This command has to be followed by a 4 character password which is calculated the following way: In the file "password.ac" there is a password of 64 characters for each call sign. On this password, 4 letters are taken with an offset. The offset is calculated like this:

You take the login time which can be shown by example with the remote command "//CS". You add the hours, minutes and seconds and from this sum you take the modulo 60 value.

Example:

The login time is 17:55:48. You calculate:

$17+55+48 = 120.$

The modulo 60 value is obtained by subtraction of 60 a number of times until you get a value that is ≤ 60 , so:

$120-60 = 60.$ The offset is 60.

Let's say that in the field "password.ac" there is the following password:

HB9VBC This_is_a_very_long_password,_which_has_exactly_64_characters_!!

^
|
60th char.

HB9VBC can then switch to sysop mode by entering the remote command `"//PRIV s_!!"`.

1.37 Explanations concerning the transmission format

I still always receive questions concerning PR transmissions, modem drivers, etc... For this reason, I'll explain here how the packet radio transmissions work.

The transmission protocol used by packet radio has the name "AX.25". This is a radio amateur adaptation of the world famous X.25 protocol (AX.25 = Amateur X.25). The data to be transmitted is collected into blocks and sent with some protocol information. These blocks are also called frames or packets, that's where the name "packet radio" comes from. The communication is controlled by special frames, e.g. connect request, disconnect request, acknowledgment of the correctly received frames, repeat demand by transmission errors, etc...

You can define 3 types of frames:

I-frames: Info frames, they contain the actual information.
 S-frames: Supervisory frames, they control if the transmission is ok or not.
 U-frames: Unnumbered frames, they take control of special functions like connection, disconnection, etc... A special form of U frame is the UI (unnumbered information) which was implemented especially for amateur radio. A UI frame contains information which doesn't correspond to a particular connection and so doesn't take part in the protocol. You can send CQ calls or beacons for example with a UI frame.

U- and S- frames have the following structure:

```
+-----+-----+-----+-----+-----+
| Flag | Address field | Control block | FCS | Flag |
+-----+-----+-----+-----+-----+
```

I- and UI- frames have the following structure:

```
+-----+-----+-----+-----+-----+-----+-----+
| Flag | Address field | Control block | PID | Info | FCS | Flag |
+-----+-----+-----+-----+-----+-----+-----+
```

Description of the particular fields:

Flag	This special character is used to differentiate each frame. This character never shows up in a frame.
Address field	The frames can be sorted to the different connections with this field. The address field contains the source and destination call signs as well as up to 8 digipeaters. Some unused bits in the address field are used for other functions like DAMA master recognition. The address field can be made up of between 112 and 560 bits.
Control block	The control block defines what type of frame it is (S-, U-

or I- frame). It also contains the frame number so you can see if you lost a frame in the transmission. The control block is made up of 8 bits.

PID The Protocol ID is only used in I- and UI- frames. It defines which layer 3 protocol is used. By normal AX.25 transmission (no layer 3) it is \$F0. The size of the PID field is 8 bits.

Info The info field is only available by I- and UI- frames. Here are the actual data, it can be made up of up to 256 bytes (PACLEN).

FCS The Frame Check Sequence is a 16 bit error proof sum which is calculated over the whole frame. With this FCS, you can check if there was a transmission error or not. Frames with errors have to be repeated by the other station until they arrive correctly.

How is each frame sent out?

The transmission uses the NRZI coding. The difference with the asynchronous serial transmission which is used by the RS232 serial port of the Amiga is the following:

With the asynchronous transmission, the data is transmitted byte by byte, before the byte, there is a start and after the byte there is 1 or 2 stop bits. If you use 8 data bits, one stop bit and no parity (8N1), then the transmission of the character %11010010 would look like this:

```

Low (+12V)      #####          #####          #####          #####
                # #          # #          # #          # #          # #
                # #          # #          # #          # #          # #
(0V)            # 0 # 1 1 # 0 # 1 # 0 0 # 1 # 0 # 1
                # #          # #          # #          # #          # #
                # #          # #          # #          # #          # #
High (-12V) ...## #####          #####          #####          #####...
                aaaaabbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbccccc

```

a = start bit
b = data bits
c = stop bit

Because each byte starts with a start bit (level change from high to low), the receiver can synchronize to the data and can always check for the polarity of the signal in the middle of a bit. Other than that, you can be sure to get the beginning of each byte thanks to this level change and leave a big time between each bytes if you have to.

With the synchronized NRZI coding, the actual state of the line is not important since a 0 bit will always consist of a level change as opposed to a 1 bit which stays at the same level. By packet radio at 1200 baud, the signals are transformed into AF tones of 1200 and 2200 Hz, the character %11010010 would look like this:

```

2200/1200 Hz      #####          #####          #####          #####
                   # #          # #          # #          # #
                   # #          # #          # #          # #
                   1 1 # 0 1 # 0 # 0 1 # 0
                   # #          # #          # #          # #

```

```

# # # #
1200/2200 Hz ...##### ##### #####...
```

Start and stop bits don't exist any more, the data bits have to be transmitted without any pause. In order to be able to differentiate each frame, they start and end with a special character, the flag (%0111110). "Bitstuffing" is used so that this character cannot be found in a frame: the sender adds a 0-bit after a serie of 5 1-bits and the receiver takes away this 0-bit again after 5 1-bits. This way, you also avoid having with a long serie of 1-bits synchronization problems because there has been no level change. The receiver should check for the level at exactly the middle of the bit.

As you can see, this form of coding is not very well adapted for the serial port chip (UART) of the Amiga. When using a simple modem on the serial port (BayCom modem), we use a simple trick: The hand shake lines are used as data lines and controlled bit wise by the software. For this reason, it is very demanding on the computer for timing and only applicable for low baud rates. For higher baud rates, you would need a special serial port chip like one from the Z85XX series, which is used for example in a TNC-2.

1.38 Explanations about the monitor contents

If you set the monitor on (M UISC), you'll see all the received and sent frames displayed in the monitor window. With the hostmode (Host.drv), the display will depend on your TNC firmware. It is usually the same as with the "BayCom.drv" and the "KISS.drv", whose display is as following:

```
fm <source call> to <destination call> [via <digi calls>] ctl...
... <control block> [pid <PID>] [[DAMA]]
```

[...]	This display is optional.
<source call>	The call sign of the sender of the frame with eventually a SSID (e.g. HB9VBC-1).
<destination call>	The call sign of the receiving station of the frame with eventually a SSID.
<digi calls>	Here you can have up to 8 digipeater call signs. If a frame was sent from a digipeater, it will have a start (HB9IAP*). If there is no star, the frame was sent from the <source call>.
<control block>	The control block of a frame is decoded and shown here as a keyword, the explanation follows.
<PID>	The Protocol ID is shown here by I- and UI- frames in hexadecimal, e.g. "F0".
[DAMA]	If the transmission came from a DAMA master, "[DAMA]" will be shown at the end of the line.

Keywords used for the representation of the control block:

Iab	An info frame. The contents of the information is shown after the monitor headline (fm...).
UI	An unnumbered info (UI- frame, Unproto transmission). The contents of the information is shown after the monitor headline (fm...).

RRa	Reception acknowledgment. The station is ready to receive more data.
RNRa	Reception acknowledgment. The station is not ready to receive more data.
REJa	Reject frame. A frame was lost in the transmission and has to be sent again.
SABM	"Set asynchronous balanced mode", a connection request.
DISC	"Disconnect", a disconnection request.
DM	"Disconnected mode", this connection does not exist or is interrupted.
UA	"Unnumbered acknowledge", acknowledges SABM or DISC.
FRMRc	"Frame reject", a serious protocol error. The 6 digits following the FRMR describes the error more thoroughly.
?cch	An undefined control block.
a	The number of the next awaited frame (reception frame counter).
b	The number of this frame (transmission frame counter).
c	Hexadecimal number.

After the control block, you can see one of the following characters:

<nothing>	A frame in protocol version 1 (not used anymore).
!	A frame in protocol version 1 with P/F bit set.
v	An answer frame in protocol version 2.
-	An answer frame in protocol version 2 with P/F bit set (Final).
^	A command frame in protocol version 2.
+	A command frame in protocol version 2 with P/F bit set (Poll).

1.39 For further questions

If you have any other questions or problems, please contact one of the following radio amateurs:

Germany:	DL3MGQ @ DB0KCP.#BAY.DEU.EU	(the author, in German if possible)
Switzerland:	HB9VBC @ HB9IAP.SROM.CHE.EU	(the translator, in English, French or German)
France:	F6HTW @ F1EBV.FMLR.FRA.EU	(French)
England:	G6RIL @ GB7SYP.#19.GBR.EU	(English)
Australia:	VK5STR @ VK5PO.#BVL.#SA.AUS.OC	(English)

That's it for the documentation to Amicom Version 2.1. The author, Gerhard, DL3MGQ @ DB0KCP.#BAY.DEU.EU, wishes you a lot of fun with it.