SANA2Meter

Marius Gröger

COLLABORATORS

	TITLE : SANA2Meter		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Marius Gröger	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

SANA2Meter

1.1 SANA2Meter

```
SANA2Meter -- A statistics monitor for SANA-II compliant network devices

(C) Copyright 1996 Marius Gröger

TABLE OF CONTENTS
```

1.2 SANA2Meter - Description

```
Description
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

The SANA2Meter is a simple tool that displays the network device statistics of any SANA-II network device. The set of displayable values include static parameters such as hardware type, MTU, speed in bits per second as well as dynamic values such as number of sent and received packets and error counters. In addition, SANA2Meter is able to display up to 20 device special statistics using generic code for the inquiry. For more experienced users and network administrators the packet type tracking function may come in handy. By means of this feature the user can keep track of one or more particular packet types. Last but not least, on highly networked systems such as gateways or bridges or servers the administrator might like SANA2Meter’s ability to display statistics of any number of network interfaces simultaneously. All dynamic statistics are updated periodically.

SANA2Meter can be called from both CLI and Workbench. A great lot of values, including the set of displayed statistics, refresh time, window position, public screen, and font can easily be customized by CLI arguments, tooltypes, and menu selection. The current setting may be saved to the tooltype list from the GUI.

SANA2Meter can be used with any SANA-II compliant network device including popular SLIP or PPP devices used to connect to an internet provider service.

1.3 SANA2Meter - Features

```
Features
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- ~Monitors all standard SANA-II network device statistics

- Monitors up to 20 special statistics

- Monitors any number of particular packet types using SANA-II packet type tracking

- Monitors any number of network devices simultaneously

- A great lot of settings may be customized

- GUI uses Olaf Barthel’s great gtlayout.library

- Localisation

- ~Completely pure code

- ~Distributed as Freeware, complete source code included

1.4 SANA2Meter - Requirements

```
Requirements
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  · AmigaOS Release 2.04 (V37) or higher, 3.1 (V40) is recommended for best
    rendering quality

  ·~gtlayout.library V24 or higher (tested with V32.3)

  ·~Some SANA-II compliant network device

  ·~Workbench 2.1 (V38) if localisation is desired
```

1.5 SANA2Meter - Installation

```
Installation
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  The installation is done best by using the supplied Installer-Script. If
  you don’t want to use the installer, just put one of the executables
  suitable for your system located in the bin directory to a place where
  you want to. The Help directory contains this AmigaGuide document.

  The Icons directory contains Icons for a standard AmigaOS Workbench as well
  as some that are suitable for the MagicWB environment and color palette.
  SANA2Meter offers some  options . If you want to launch SANA2Meter from
  Workbench edit the tooltype list accordingly. If you plan to run in from
  CLI, pass in the right commandline arguments.

  In the Catalogs directory are  localisation  catalogs stored. Currently
  the only supported languages include:

  · enlish (built-in)
  · german

  SANA2Meter has been designed completely pure and reentrant. So you might
  want to put it into the AmigaDOS resident list by issuing the command:

    Resident SANA2Meter

  Note, however, that due to the poor design of Workbench the resident list
  is not searched if programs are started from Workbench, but only
  if started from CLI.
```

1.6 SANA2Meter - Startup Options and Tooltypes

```
Invoking SANA2Meter
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  SANA2Meter accepts the following options either on the commandline or given
  as tooltypes:
```

DEVICE	Network device specification
UNIT	Network device specification
SHOW	Display settings
LIST	Display settings keyword table
TRACKTYPE	Packet-type tracking settings
FORCE	Special device opening method
OFF	Initial display status
FROM	Configuration file
WINDOW	Output console for Workbench
REFRESH	Refresh rate for display
FONTNAME	Name for GUI-Window font
FONTSIZE	Size of GUI-Window font
LEFT	Initial X-position of GUI-Window
TOP	Initial Y-position of GUI-Window
NOACTIVATE	Initial activation state of GUI-Window
WINDOWTOBACK	Initial depth-ordering of GUI-Window
PUBSCREEN	Home-screen of GUI-Window

1.7 SANA2Meter - Options - DEVICE

DEVICE

Template: DEV=DEVICE/A
Tooltype: DEVICE
See also: FROM , UNIT

This keyword specifies the name of a SANA-II compliant network device. Depending on your system setup it might be neccessary to specify the full path, e.g. devs:networks/a2065.device .

See the FROM option to learn how to specify more than one device.

1.8 SANA2Meter - Options - UNIT

UNIT

Template: U=UNIT/K/N
Tooltype: UNIT
Default: 0
See also: DEVICE

This lets the user specify the network device unit number to use for opening the device.

1.9 SANA2Meter - Options - SHOW

SHOW

Template: S=SHOW/K
Tooltype: SHOW
Default: ipackets,opackets,baddata,overruns,unknown,mtu,bps,hwtype
See also: LIST , TRACKTYPE

This parameter lets the user specify the statistical counters that should be displayed. The argument is a list of keywords, each two keywords must be separated by either a comma (,), a semi-colon (;), a bar (|), or a blank ( ). The following keywords are recognized (abbreviated forms of the keywords are denoted in parenthesis):

```
   ipackets (ip)   Number of packets that the unit has received
   opackets (op)   Number of packets that the unit has sent
   baddata (bad)   Number of bad packets received (i.e., hardware CRC
                   failed)
   overruns (ovr)  Number of packets dropped due to insufficient resources
                   available in the network interface.
   unknown (uwn)   Number of packets received that had no pending read
                   command with the appropriate packet type.
   reconfigs (rcf) Number of network reconfigurations since this unit was
                   last configured.
   mtu             Maximum Transmission Unit, the size, in bytes, of the
                   maximum packet size, not including header and trailer
                   information.
   BPS             Best guess at the raw line rate for this network in bits
                   per second.
   hwtype (hw)     Specifies the type of network hardware the driver
                   controls.
   special1 (s1) trough special20 (s20)
                   Special device statistics. This kind of statistics is
                   optional to a network device driver writer, and therefore
                   not all devices support it. The special statistics are
                   used to maintain statistic counters for values that are
                   special to the device. For instance, a SLIP device might
                   provide a counter for the number of compressed packets
                   that were received so far. SANA2Meter sort of "probes"
                   the device to find out if, how much and what special
                   statistic counters it provides. The  menu  of the
                   interactive GUI will be adapted accordingly.
```

The LIST option displays a list of all keywords that are allowed to be specified.

1.10 SANA2Meter - Options - LIST

LIST

Template: L=LIST/S
Tooltype: LIST
See also: SHOW

Instead of opening the GUI a list of displayable statistics will be printed to the console. A console window will be opened if started from Workbench. The keywords that are listed are suitable as arguments for the SHOW option.

1.11 SANA2Meter - Options - TRACKTYPE

TRACKTYPE

Template: TT=TRACKTYPE/K
Tooltype: TRACKTYPE
Default: <n/a>
See also: SHOW

This parameter allows to keep track of the statistics of particular packet types. Since type tracking is done completely generic, the type number to be monitored must be given as a decimal or a sedecimal value. Multiple packet types may be kept track of. The types in question are determined by list of packet type specifiers. Similar to the keyword list of the SHOW argument, each packet type specifiers must be separated by either a comma (,), a semi-colon (;), a bar (|), or a blank ( ).

A packet type specifier primarily determines a certain packet type to be monitored. The following table lists some common packet type numbers:

packet type number	Meaning
$800	IP packets on Ethernet. This type number is also used for some other network interfaces, e.g. (C)SLIP or PLIP.
$806	ARP packets on Ethernet
$21	IP packets on PPP (RFC 1334 IP protocol).
$F0	ARCNET(?) (RFC 1051 framing)
$F1	ARCNET(?) (RFC 1051 framing)
$D4	ARCNET(?) (RFC 1201 framing)
$D5	ARCNET(?) (RFC 1201 framing)
$D6	ARCNET(?) (RFC 1201 framing)

In addition, the packet type specifier may determine which particular statistics about the packets of this type should be actually displayed. The following values are updated for each type and can be displayed:

```
PacketsSent     Number of packets of this particular type sent.
PacketsReceived Number of packets of this particular type that
                satisfied a read command.
BytesSent       Number of bytes of data sent in packets of this
                particular type.
BytesReceived   Number of bytes of data of this particular packet
                type that satisfied a read command.
PacketsDropped  Number of packets of this particular type that were
                received while there were no pending reads of that
                packet type.
```

In the packet type specifiers, a 5-character-mask determines which value should be displayed. Each displayable value is associated with a certain character position in the mask. As studying examples is always the best way to learn, the next table lists some valid packet type specifiers:

packet type specifier	Meaning
$800	Display full statistics of $800 type packets.
$800:--##-	Display only the number of transfered bytes of $800-type packets.
$800:##--#	Display only the number of transfered or dropped packets of $800-type packets.
$800:xxoox	dito.
$800:XXOOX	dito.
$800:++..+	dito.
$800:++--+	dito.
$800/++--+	dito.
$800:--##-,$806:--##-	Display only the number of transfered bytes of $800- and $806-type packets.
$800:--##-;$806:--##-	dito.
$800:--##-|$806:--##-	dito.
$800:--##- $806:--##-	dito.

Notice that on CLI startup the argument of a keyword may only contain blanks if it is enclosed in quotation characters (").

1.12 SANA2Meter - Options - FORCE

FORCE

Template: FORCE/S
Tooltype: FORCE

Some SANA-II devices might not love the fact that SANA2Meter is a not a real device user program in the way an actual protocol stack is. This switch makes SANA2Meter to appear more like such.

1.13 SANA2Meter - Options - OFF

OFF

Template: OFF/S
Tooltype: OFF

Specifying this parameter will initially turn off the display of the corresponding network interface.

1.14 SANA2Meter - Options - FROM

FROM

Template: FROM/K
Tooltype: FROM

Default: n/a

The argument to this keyword specifies an ASCII file containing more definitions. Use of such a file is necessary particularly if you want to monitor more than one device. Each line of the configuration file will be parse exactly as the commandline. Thus, a sample configuration might look like this:

```
  ; A comment
  DEVICE slip.device UNIT 0 SHOW=ipackets,opackets TT $800
  DEVICE magplip.device UNIT 0 TT $800 OFF
  LEFT 20 TOP 30 PUBSCREEN MOSAIC.1
```

Note:

* a semicolon introduces a comment, the whole line will be ignored
* each device must be specified on a new line
* since the DEVICE specification is optional, the 3rd line is correct,
  although it only specifies the window's position

The keywords REFRESH , LEFT , TOP , WINDOWTOBACK , FONTNAME , NOACTIVATE , PUBSCREEN may be overwritten by specifying them on the commandline.

If a device is specified directly on the commandline, it will of course be displayed regardless if you use a configuration file or not.

SANA2Meter remembers whether a device specification was given by CLI arguments (or tooltypes), or whether it came from a configuration file. So the "Save configuration" menu will write the current settings to the icon's tooltype list and the configuration file respectively.

1.15 SANA2Meter - Options - WINDOW

WINDOW

Template: WINDOW/K
Tooltype: WINDOW
CLI-Default: CLI-Window;
WB-Default: CON:0/0/400/200/SANA-II Meter Tool Output/AUTO/WAIT/CLOSE

Console window specifier to define an output channel when started from Workbench. On CLI startup this keyword is parsed but it's value will be ignored.

1.16 SANA2Meter - Options - REFRESH

REFRESH

Template: RF=REFRESH/K/N
Tooltype: REFRESH
Default: 2

```
This lets the user specify the number of seconds between each two
refresh cycles.
```

1.17 SANA2Meter - Options - FONTNAME

```
FONTNAME

Template: FONT=FONTNAME/K
Tooltype: FONT, FONTNAME
Default:  System default font
See also:  FONTSIZE

By this keyword a different font for the display window may be specified.
```

1.18 SANA2Meter - Options - FONTSIZE

```
FONTSIZE

Template: FONTSIZE/K/N
Tooltype: FONTSIZE
Default:  System font’s height or an 8 point custom font, respectively.
See also:  FONTNAME

By this keyword a different fontsize for the display window may be specified.
```

1.19 SANA2Meter - Options - LEFT

```
LEFT

Template: LEFT/K/N
Tooltype: LEFT
Default:  30
See also:  TOP

The X-Coordinate where the window should open.
```

1.20 SANA2Meter - Options - TOP

```
TOP

Template: TOP/K/N
Tooltype: TOP
Default:  20
See also:  LEFT ,  PUBSCREEN

The Y-Coordinate where the window should open.
```

1.21 SANA2Meter - Options - NOACTIVATE

```
  NOACTIVATE

  Template: NA=NOACTIVATE/S
  Tooltype: NOACTIVATE

  Setting this switch makes the window open in inactive state
```

1.22 SANA2Meter - Options - WINDOWTOBACK

```
  WINDOWTOBACK

  Template: W2B=WINDOWTOBACK/S
  Tooltype: WINDOWTOBACK

  Setting this switch moves the window behind all other non-backdrop
  windows right after opening it.
```

1.23 SANA2Meter - Options - PUBSCREEN

```
  PUBSCREEN

  Template: SCREEN=PUBSCREEN/K
  Tooltype: PUBSCREEN
  Default:  System default public screen (usually the workbench screen).
  See also:  LEFT ,  TOP

  The name of the public screen where the window should open on.
```

1.24 SANA2Meter - The Menu

```
The Menu
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  A menu strip is attached to the display window. Following a description of
  the various items. The menu texts are referred to using the built-in
  english texts. Users running SANA2Meter in a different language will have
  to do some translation work anyway when reading this document :-)

  Project/Save configuration

     Saves the current window position, and settings of the "Settings" and
     "Show" menus to the program's list of tooltypes. If there are devices
     displayed which have been specified in a separate configuration file,
     than that file will be updated.

  Project/About...
```

Displays a requester with copyright and author information.

Project/Quit

Terminates SANA2Meter.

Settings/Update (seconds)

The attached sub-menu may be used to set the refresh rate for the dynamic statistics in seconds.

Settings/xxx.device

Each network device is represented here by a checkable menu item. By checking and unchecking of the item you may turn the complete display of the corresponding device's statistics on and off. However, this doesn't affect the settings of the next menu.

xxx.device/#?

The set of displayable statistics of each network device are displayed in these menus. The items can be checked or unchecked to modify the set of displayed statistics. Note that if you have turned of the device's display by means of the Settings/xxx.device menu, changes done in this menu won't be visible.

The special device statistics are near the end of the menu. SANA2Meter uses automatically the statistic counter names as provided by the network driver. Note that it is the duty of the network driver to localize these names. Being a generic monitor tool without any knowledge of a particular network driver SANA2Meter can only reproduce the special statistic counter's names.

If packet type tracking is used, each type being tracked is represented by a separate submenu at the end of the menu, the subitems of which may be checked or unchecked to modify the set of displayed statistics.

Note that SANA2Meter will only update values that are actually displayed. So don't assume that values will be tracked "in background".

1.25 SANA2Meter - Program Termination

Program Termination
~~~~~~

The programm may be stopped by clicking the window's closebox, by sending an AmigaDOS break-signal to the process or by pressing ESC when the window is active.

1.26 SANA2Meter - Caveats
~~~~~~

```
Caveats
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  This section tries to address some known problems.

  I will always try to give the user the necessary technical background to
  understand the problem and, if possible, to include a work-around.

  If you have found a similar problem which is not mentioned in this
  document, you should contact  me  in order to append it to this
  section.

   gtlayout.library
   Kickstart releases
   Packet type tracking
```

1.27 SANA2Meter - Caveats - gtlayout.library

```
  gtlayout.library

  A lot  of email has reached me, complainig about gtlayout.library
  V24 or higher, which isn't available separately on Aminet.

  I don't know why. I don't even know if SANA2Meter's requirement for a V24
  release is chosen correctly, since I don't want to dig through the complete
  autodocs. Unfortunately the gtlayout.library only comes with a standard
  AutoDoc, there is no separate development and feature history. Get a recent
  term release, which will always include the latest release of
  gtlayout.library. Ask Olaf Barthel <olsen@sourcery.han.de> why the library
  distribution and support is that bad --- and hope that he'll answer your
  questions.

  I will never include gtlayout.library my distribution, so please don't ask
  me to do so!
```

1.28 SANA2Meter - Caveats - Kickstart releases

```
  SANA2Meter and different Kickstart releases

  SANA2Meter's GUI is built using the powerful gtlayout.library by Olaf
  Barthel. The aim of this library is to provide a more abstract and
  intelligent inferface than Kickstart's built-in gadtools.library. However,
  gtlayout.library only defines API functions, data structures and tag values
  that are additional to gadtools.library. Most internal actions of
  gtlayout.library end up in a some calls to gadtools.library. This saves
  memory and retains the Amiga look and feel, but also inherits the various
  levels of functionality found in the different versions of
  gadtools.library, not to mention bugs.

  Due to some bug in gadtools.library V39 the displayed numbers are not
  nicely right justified as done under gadtools.library V40.
```

When running AmigaOS 2.04 (V37), the numbers won't even be displayed in highlighted text.

There might by other limitations that I am not aware of by now , especally with respect to AmigaOS 2.04.

1.29 SANA2Meter - Caveats - Packet type tracking

SANA2Meter and type tracking

Type tracking as defined by the SANA-II standard seems to assume only one device user tracks a certain type. At least, trying to track a type that is alreay tracked by some other user leads to a SANA-II error condition. It is not told whether this means to the refused user that he is now disallowed to get the desired statistics, or that he may nevertheless retrieve them.

SANA2Meter acts like this:

- If the type could be tracked successfully, it will be displayed and untracked on exit.

- If it could not be tracked successfully, it will nevertheless attempted to be displayed; if this bothers the device SANA-II tells it to refuse the inquiry request as currently not applicable. SANA2Meter will stop displaying the counter in that case. The type won't be untracked.

By the way, the parallel port SANA-II network device "magplip.device" (AMINET:/comm/net/magplip37.7) will nest multiple type tracking requests satisfying all users as well as the SANA-II standard.

Generally SANA2Meter will stop displaying a certain value when the corresponding i/o operations needed to retrieve the statistics fail.

1.30 SANA2Meter - Troubleshooting

```
Troubleshooting
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Q: The device can't be opened. A requester appears telling that the
   device couldn't be opened.

A: Check whether you have specified the device name and unit correctly. The
   device name may need to be specified along with a full file specification:
   According to the SANA-II 2.0 standard all network interface drivers should
   be located in devs:networks. The path specification may be provided
   case insensitive, the device name, however, must be given using the
   original letter cases.

Q: The device can't be opened. A requester appears telling that the
```

device couldn't be opened. The device name and unit have been entered correctly, though!

A: Some SANA-II devices need request certain additional data when opening the device that are needed for real protocol stacks, but that are actually neccessary for the retrieval of statistic information. The FORCE option makes SANA2Meter to appear a bit more like a protocol stack.

Q: Sometimes a certain statistical information vanishes without command. It can't be display again.

A: This indicates that for some reason the network device refused to provide SANA2Meter with the desired information. SANA2Meter can't do anything about this, except ceasing the display of the value in question. Most often this will happen to the display of packet type tracking information in the case that the operation of the protocol stack will be terminated. See also the Caveats section.

If anything else has failed, you may contact me .

However, since I get a lot of email and want to satisify my employer as well, please don't waste the time of the both of us with undetailed, vague error reports. Please read the manual carefully, completely and not only once. Try to understand by thinking instead of stopping to think just because emailing me is so easy. If you can't avoid to ask me, enclose a table of your system configuration with your mail.

1.31 SANA2Meter - Known Bugs

Known Bugs in older Releases
~~~~~~

(1)

SANA2Meter 37.1 and 37.2 failed to work under Kickstart 3.0 releases. This was due to a bug in older versions of gadtools.library, that wouldn't render number fields correctly right justified! Version 40.1 of gadtools.library fixes that bug. Work-around for to get those versions to work is to get SANA2Meter 37.3 or higher (which simply suppresses right justification when a gadtools version lower than 40.1 is installed), or/and to run Kickstart 3.1 or higher. (For more technical detail refer to the AutoDoc gadtools.library/CreateGadget()).

Thanks all users that pointed that bug out. Please don't blame me but the makers of the (anyway brain-damaged) gadtools.library.

(2)

Some ancient code prevented SANA2Meter 37.2 and 37.3 to display more than 10 special statistics.

(3)

rh(c)slip.device 38.1 seems to have a little bug in it's SANA-II I/O
~~~~~~

routines which which made previous releases of SANA2Meter processing any user input only at display update time (depending on configuration). The current release works around this bug. (For the technical interested ones: after a completed DoIO() the signal bit associated with the MsgPort used for device I/O remained set. This caused a direct fall-through at the next Wait() in the main loop. Since the same MsgPort was used for SANA-II I/O and timer I/O, the program assumed a completed timer I/O request and issued a WaitIO() on the timer request. In fact, the timer was NOT done and, thus letting the program hang inside WaitIO(). Now there are different ports used for SANA-II I/O and timer I/O, and a SetSignal(0, 1<<mp_SigBit) is issued after SANA-II stats update I/O to prevent a spurious activation of the main loop.)

If you have found bugs, please contact me .

1.32 SANA2Meter - Future plans

Future plans
~~~~~~

Perhaps I will implement some more features:

·~Screen jumping

·~CPS calculation

If you have any suggestions, contact me .

1.33 SANA2Meter - Localisation

Localisation
~~~~~~

As of the current version the following languages are supported by built-in texts and transient catalog files:

- enlish (built-in)
- german

Further localisation depends on the Amiga user's community. There has been provided a blank catalog source file in the source tree (see Source/blank.ct). If you want to translate it to a new language, say, a klingonic version, you should:

1. Contact me to get an exclusive permission to perform the translation.
2. Perform it.
3. Send to me the klingon.ct file
4. Watch out for the next release's documentation, section "Credits".

1.34 SANA2Meter - History

Development history of SANA2Meter
~~~~~~

Release 37.6

·~Multiple network devices may be monitored simultaneously

· An optional configuration file keeps permanent settings also for CLI usage

·~Bugfix: Some ancient code would only display 10 special statistics

·~Workaround: rh(c)slip.device causes no longer delay of user input processing (see Known Bugs of older Releases)

Release 37.3

· SANA2Meter now also renders it's display under Kickstart 3.0 (V39). Earlier releases of SANA2Meter failed to operate accurately due to a bug in gadtools lower than 40.1

·~Loosing track of some values (most notably packet type tracking information) will terminate their display

Release 37.2

·~Packet type tracking

Release 37.1

·~Initial release

Technical history of "SANA2Meter"

For more technical details, please refer to the source code. In the header section of each source file, you find the pseudo-keyword $HISTORY: after which follows the complete revision history of this file.

1.35 SANA2Meter - Credits

Credits
~~~~~~

The following people have --- willingly or unwillingly --- contributed to SANA2Meter's existence. I wish to thank all of them.

The GUI has been created using Olaf Barthel's gtlayout library.

The tooltype modification uses parts of the supra.lib code written by Jure Vrhovnik.

The integrated CLI/Workbench argument parsing is done using the ExtReadArgs package written by Stefan Ruppert.

Some stylistic decisions concerning the look and feel of the program and the documentation have been inspired by the Executive package from Petri Nordlund, a very recommendable piece of software, by the way.

Special thanks go to Jeroen T. Vermeule and Greg Hope for hints and suggestions.

Last but not least I wish to thank all those many people who contacted me for asking questions and suggestions.

1.36 SANA2Meter - Copyright Notes

Copyright Notes
~~~~~~

This software is freeware. It is provided as-is and is subject to change; no warranties are made. All use is at your own risk. No liability or responsibility is assumed.

It's strictly forbidden to include this archive in any kind of software collection except Fred Fish's AmigaLibrary, Meeting Pearls CD's, Aminet, Aminet CD's and BBS fileareas.

SANA2Meter is Freeware and comes with the full source code, because I have learned much from studying other people's sources, even if I only learned how one should better not code :-)

Regarding the source code of SANA2Meter, you may make any changes to the source for your own use. However, you may not distribute any (re-)compilations of SANA2Meter. If you consider your changes useful for everybody, please get in touch with me. This way I can try to include them in the next public release.

Generally you may use the source for your own purposes like you want, whole or in parts. If you release a program (regardless of the policy that is applied for it's distribution) which uses vital parts of SANA2Meter, please drop me a note.
~~~~~~

1.37 SANA2Meter - Author Information

```
Author Information
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  If you like this software, or have any suggestion how to improve it, or
  just want to complain about it, feel free to contact me.

  Home address:
     Marius Gröger,
     Bärstadter Str. 4
     65307 Bad Schwalbach (GERMANY)

  Internet email addresses:
     mag@sysgo.de
     groeger@goofy.zdv.uni-mainz.de
```