hepc3

Version 1.1.0
User's Manual

Dr. Albrecht Dreß*
Max-Planck-Institut für Radioastronomie
Abteilung für Infrarot-Interferometrie
Auf dem Hügel 69
D-53121 Bonn
GERMANY

30th January 2001

Abstract

The Linux kernel module hepc3 provides simple access to the comports A and B of Hunt Engineering's HEPC3, HEPC4 and HECPCI1 TIM40 DSP motherboards. PCI Bus Master DMA accesses are supported, but the JTAG port is not available.

Contents

*E-Mail `ad@mpifr-bonn.mpg.de`

1 Introduction

The Linux kernel module hepc3 is a character device driver which enables the kernel to support up to eight of Hunt Engineering's HEPC3, HEPC4 (standard PCI) or HECPCI1 (CompactPCI) TIM40 DSP motherboards. The hardware specifications for these boards can be obtaind from the following web addresses:

HEPC3: http://www.hunteng.co.uk/products/c4x/hepc3.htm

HEPC4: http://www.hunteng.co.uk/products/c4x/hepc4.htm

HECPCI1: http://www.hunteng.co.uk/products/c4x/hecpci1.htm

The driver supports both comports A and B. For comport A, the boards can act as PCI Bus Masters, which increases the data transfer rate and reduces the cpu load. Comport B uses an interrupt-driven mode, which is also selectable for port A. The JTAG port is not supported, and as I do not have the necessary hard- and software, it is not likely that I will add it in the future.

The driver and all accompanying documentation is

Please report bugs or suggestions to ad@mpifr-bonn.mpg.de or send mail to:

Albrecht Dreß
Max-Planck-Institut für Radioastronomie
Box 2024
D-53010 Bonn
Germany.

2 Installation of the driver

2.1 Prerequisites

To install the driver you will need

- the file `hepc3-1.1.0.tar.gz` which is available from UNC's metalab archive (formerly known as sunsite) in the directory

  `http://metalab.unc.edu/pub/Linux/kernel/misc-cards/`

  and its mirrors. New versions of the driver will follow this naming scheme;

- a Linux system with kernel version 2.0.x or 2.2.x. Late 2.1.x and early 2.3.x development kernels *may* also work, but this is not tested, as well as using it with 2.4.x kernels. Please note that the driver has only been tested with Linux/Intel and LinuxPPC. Other Linux/PCI platforms (e.g. Alpha) should also work. Please mail me your experiences;
- module support in the kernel, the kernel header files, the gcc, and the module utilities (`insmod` and friends);
- proc filesystem support is highly recommended but not necessary.

2.2 A step-by-step guide

For easy installation use the following steps:

- unpack the archive file:

  ```
  tar -xzvf hepc3-1.1.0.tar.gz
  ```

- enter the directory `hepc3-1.1.0` and run the configuration script. It will check your system's setup and create a makefile. The script accepts several options, to get a description type

  ```
  ./configure --help
  ```

  There are only two options which are important for this module:

  – `--with-verbose-errors=`{`yes|no`}
    Setting this option will enforce the driver to complain about several internal errors. This is needed for debugging purposes and there is no reason why you should enable it. The default is "no";
  – `--with-pci-busmaster=`{`yes|no`}
    Port A of the HEPC3/HEPC4/HECPCI1 supports a PCI Bus Master DMA mode which will transfer data fast with much less cpu intervention. The default is yes, but there are several reasons why you do not want this feature:
    * At least one board will be plugged into a slot which does not support PCI Bus Mastering or PCI Bus Mastering is broken on your machine. Consult your motherboard's manual for details. As cpu overhead would remove the advantages, it is *not* possible to mix several boards in one system with different settings for port A.
    * PCI Bus Master DMA does not work on ix86 systems with Intel 440 chipsets (See section 5.1, page 8). `configure` tries to detect these chipsets and disables PCI Bus Master Mode, but if you know your motherboard uses an Intel 440 chipset, you should say no here.

To configure hepc3, type

```
./configure <options>
```

The configure script prints a lot of status information on the screen. If everything was successful, just type

- `make`

  to compile the module. If make was successful, type

  ```
  make install
  ```

  to install the compiled module and the header file to standard directories and

  ```
  make install_man
  ```

  to copy the man page.

Before you can use the module, you have to insert it into the running kernel and create the corresponding device files. As you can use different approaches for this, this information is contained in the next section.

3 Using the driver

3.1 Insertion into the kernel

3.1.1 Common options

The module recognises several options for both manual and automatic insertion:

`hepc3_major=`*m* sets the device's major number to m. Devices are identified by the kernel through their major numbers. A lot of them are pre-defined, and those currently supported can be found in `/proc/devices` in the section "Character devices". If you omit this option, the kernel will choose an appropriate number (usually at the upper end of the 0..255 range) for you, but you might end up with a different number each time you insert the module. With this option, you can avoid this, but you should choose the number carefully. The range from 60 to 63 should be safe;

`hepc3_buf_size=`*s* sets the sizes of the internal input and output buffers to s 4-byte words each. As a rule of thumb, a larger buffer increases transfer speed if big chunks of data have to be transferred, but as kernel memory can not be swapped, the space for user programs decreases[1]. For optimal memory usage, s should be a power of 2 for Linux 2.2 and slightly less than a power of two for Linux 2.0. The default is 1000 for Linux 2.0 and 1024 for Linux 2.2;

`hepc3_secure=`*v* defines whether all users have access to the board reset `ioctl` call ($v = 0$, see section 4.4, page 7 for details) or only root may perfom this operation ($v \neq 0$). The default is $v = 1$;

`hepc3_latency=`*l* sets the value of the PCI Latency Timer Register to l. It influences the maximum time a board can request the bus during PCI Bus Master accesses from or to comport A. As a rule of thumb, a larger value gives more time to the hepc3 DMA, decreasing the chances of the cpu (and other pci devices) for data transfer. The default value (which should be a good choice in most cases) is 32. Note: this option is available only if the driver was compiled with PCI Bus Master DMA support.

[1] The memory is allocated upon an `open` call, therefore unused ports need space only for some smaller internal structures.

3.1.2 Manual insertion and removal

To insert the module by hand, just type as root

```
insmod hepc3 <options>
```

with zero or more of the options listed above. To check the result of `insmod` see the last few lines reported by `dmesg`. Typically, they will look like this:

```
hepc30: Hunt Engineering HEPC3/HEPC4/HECPCI1 [C]PCI TIM40
                                                motherboard Rev. 0
hepc30: IRQ=11, IO at 0x1280, 0x12c0, 0x1300, 0x1340, 0x1400
hepc3: character device has major number 254
```

The only information which is useful not only for debugging purposes is in the last line: the device major number which the kernel has assigned automatically if you did not pass a `hepc3_major` argument. You will need this number for the creation of the device files (see section 3.2, page 6). Additional information can be obtained from the proc file entry (see section 3.3, page 6).

To remove the module from the kernel as usual type (as root)

```
rmmod hepc3
```

The advantage of this method is the finer control over the module's options. The disadvantage, however, is that you must be root to do this. Therefore, most people will prefer the automatic insertion, as described in the next section.

3.1.3 Automagic insertion and removal

The kernel runs a special process, the kernel daemon which can perform the actions described in the last section "automagically". Each time a process wants to access a device file, the kernel checks, whether it can support a device with the corresponding major number. If not, it asks the kernel daemon to find and insert the driver. A short time after the last process which used the driver terminated, the kernel daemon will remove the module. Everything is absolutely transparent for the user.

The kernel daemon needs the following setup to do its job:

- the module file `hepc3.o` must be in the directory `/lib/modules/<version>/misc`. This is done automatically, if you used `make install` during the installation;
- the file `/etc/conf.modules` or `/etc/modules.conf` must contain information which links the device's major number to the module file, and you must specify any options which should be passed to the module. Assuming you use the device major number 62 (see section 3.2, page 6 below), the entries in the file should look like this:

  ```
  alias char-major-62 hepc3
  options hepc3 hepc3_major=62
  ```

To check the success of the kernel daemon, you may use `dmesg` or look into the proc file (see section 3.3, page 6).

3.2 Creating device files

The linux kernel identifies drivers the *major numbers* of the device files. The *minor numbers* are used to identify the specific ports on each board. To create the device files, enter (as root) for each comport of every board

```
mknod /dev/<name> c <major> <minor>
```

Although you might use any `name` you like, the suggested scheme is listed in table 1, whereas the `minor` numbers *must* be the listed ones. The `major` number is, of course, the one which you either defined for the module or got from the kernel (see sections 3.1.2 and 3.1.3 above).

Check the access flags of the device files; if you want to grant access to them for other users, you usually will have to change them using `chmod`.

3.3 Reading status information

If you have support for the proc filesystem compiled into your kernel, you cat get some more specific information about the status of the driver and all boards by typing

```
cat /proc/hepc3
```

A typical output will look like this[2]:

```
Hunt Engineering HEPC3/HEPC4/HECPCI1 [C]PCI TIM40 motherboards
1 board(s), device major number is 62
driver version 1.1.0
port A uses DMA io, port B interrupt driven io

hepc30: Rev. 0, IRQ=11, IO at 0x1280, 0x12c0, 0x1300, 0x1340, 0x1400
  A: active (0x1), buffered (i/o) 0/0; byte swap disabled
  B: passive (0x4d), buffered (i/o) 0/0; byte swap enabled
```

Apart from the general information (number of boards, device major number, port A uses DMA or interrupt driven io), the following data is listed for each board:

- interrupt line and io addresses. This data is important only for debugging purposes;

[2] It will be a little bit more formatted on your terminal...

Board	Minor Number	Device File Name
1st	port A: 0	`hepc30_a`
	port B: 1	`hepc30_b`
2nd	port A: 4	`hepc31_a`
	port B: 5	`hepc31_b`
3rd	port A: 8	`hepc32_a`
	port B: 9	`hepc32_b`
n	port A: $4 \cdot (n-1)$	hepc3$(n-1)$_a
	port B: $4 \cdot (n-1)+1$	hepc3$(n-1)$_b

Table 1: device numbering and naming scheme

- whether a process uses a port ("active") or not ("passive"), the current port control flags, the number of bufferd input and output 4-byte words and the byte swapping scheme.

If the module was configured with the *with-verbose-errors* option (see section 2.2, page 3) some additional information about the state of the AMCC S5933 internal registers is printed.

4 Accessing comports from user programs

User programs communicate to a comport on a HEPC3, HEPC4 or HECPCI1 through simple file io operations.

4.1 lseek

This operation is not implemented (and does not make sense). It always returns `-ESPIPE` upon call.

4.2 read and write

These calls work as expected, but with the following exceptions:

- due to the fact that all communication to and from the dsp board has to be organised in 4-byte words, the `size_t count` parameter has to be a multiple of four, otherwise the functions returns `-EINVAL`;
- in blocking io mode, the driver guarantees that all data is transferred in a single call (the functions return either an error or `count` in this case). In non-blocking mode the behaviour is the expected one.

4.3 select and poll

These calls work as expected. Note: `poll` is implemented only for Linux 2.2.x and above.

4.4 ioctl

Note: To use `ioctl` calls for the hepc3 module, you need to include the file `hepc3.h`.

The following ioctl's are provided (a third argument has to be supplied only if indicated):

`HEPC3_IO_B_RES` resets the whole board and any hardware conntected to the RESET output of the board by asserting the reset line, so use this function with due care. Unless `hepc3_secure` (see section 3.1.1, page 4) has been set to zero, the calling process must have superuser privileges for this operation. Otherwise, `-EPERM` is returned.

Note: This operation is performed automatically during the installation of the driver.

`HEPC3_IO_P_RES` resets the port connected with device handle `d`.

`HEPC3_IO_G_ISTAT` ioctl has to be called with a `*int` as third argument. It is filled with the minimum number of 4-byte words which can be read from the device handle `d`.

HEPC3_IO_G_OSTAT ioctl has to be called with a `*int` as third argument. It is filled with the minimum number of 4-byte words which can be written to the device handle `d`.

HEPC3_IO_G_CNFG ioctl has to be called with a `*int` as third argument. It is filled with the status of the config line of device handle `d`.

HEPC3_IO_S_SWAP ioctl has to be called with a `*int` as third argument. If it points to a value $\neq 0$, internal byte swapping in the data path is enabled for the port connected with device handle `d`, otherwise it is disabled. On a big endian machine (like the Apple PowerMac) byte swapping is enabled by default, on little endian machines (like ix86) it is not.

HEPC3_IO_G_SWAP ioctl has to be called with a `*int` as third argument. It is filled with the current internal byte swap setting for the port connected with device handle `d` (see `HEPC3_IO_S_SWAP` above).

4.5 open and close

These calls work as expected. Both blocking and non-blocking io are supported. The file name to be used for the `open` call is the device name (see section 3.2, page 6).

5 Platform specific remarks

5.1 Intel

The hepc3 module works with Linux versions 2.0.x and 2.2.x. It supports PCI Bus Master DMA, but during the tests a HECPCI1 CompactPCI card failed with "hard" kernel lockups on a CC5 industrial pc made by or-SBS industrial computers. The reason for this is a bad layout of the cpu device.

Another problem arises on motherboards with Intel's 440 chipsets. PCI Bus Master DMA works in principle, but sometimes some words contain invalid data. The reason for this problem is a bad interaction between the 440 bridge and the AMCC S5933 (which serves as PCI chip on the Hunt cards) if the CPU goes to sleep mode[3]. During idle times, Linux issues a `hlt` instruction to the cpu which will then "sleep" until the next interrupt occurs. This has been good practice since the days of the PDP11; it reduces the power consumption of the cpu and increases its lifetime.

You may notice, that the S5933/Intel 440/DMA combination will actually work with Winbloze. For no good reason, Winbloze does *loop* during idle times, keeping the cpu pretty hot, but this time it prevents errors with DMA. You should, however, remember, that some day Microsnot might tell you, that their clever design engineers "invented" a new method to save power during cpu idle times. Then you will have a problem...

5.2 PowerPC

The module was tested with LinuxPPC Release 5, based upon a 2.2.6 kernel, on a Power Macintosh 7300/166, and worked without any problems both with and without PCI Bus Master DMA. As the PowerPC is a big endian machine, byte swapping is enabled by default (see section 4.4, page 7 above).

Apart form the byte-swapping code (which might also be useful on other platforms), the driver does not contain *any* PowerPC specific stuff[4]. Therefore, the driver should also work with other Linux 2.2 based PowerPC systems, but this was not tested. The module *might* also work with MkLinux which is 2.0 based, but again this was not tested. Please mail me your experiences.

[3] It became not quite clear, whether it is AMCC's or Intel's fault, but AMCC will upgrade their silicon.

[4] There were, however, some minor problems as listed in the next section. Furthermore, the device is not enabled

5.3 Other platforms and porting guide

Due to the high level of platform independence, the module should also work on other systems with PCI busses, e.g. Alpha's or Sparc's, but due to the lack of hardware, this was not tested[5].

If you want to test the module on such hardware or do some porting, here are a couple of hints and common pitfalls which might help in doing the job:

- *storage sizes:* I tried to avoid data types whose storage sizes might vary on different platforms, but I can not guarantee, that I found all critical variables.
- *endianess issues:* on the PowerPC (which is a big endian) "in" and "out" commands are translated (swapped) correctly, so endianess is not a problem. PCI Bus Master transfers, however, are *not*, therefore (if you look into the code) the setting of the byte swap flag is reverted in this case. Remember, that this might be totally different on another platform.
- *do not trust PCI registers:* due to remapping by the kernel, PCI registers might tell you other base addresses and interrupts than those the kernel uses (this was a problem on the PowerPC). Therefore, this data is read from the kernel structure, which is always correct and safer, anyway. You should, however, note, that this code is used only for Linux 2.2 (I do not have a working 2.0 version on a Power Macintosh).

Please mail me your experiences (good or bad)!

A License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

automatically in the PCI Command Register (as on Intel), maybe because the card does not "talk" correctly with *OpenFirmware*. Now these flags are set explicitly which is a "paranoia setting" on Intel (and maybe other platforms).

[5] If anybody has a spare Alpha or Sparc, my address is on the cover page ;-)

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

(c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.) These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright © 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright © 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.