

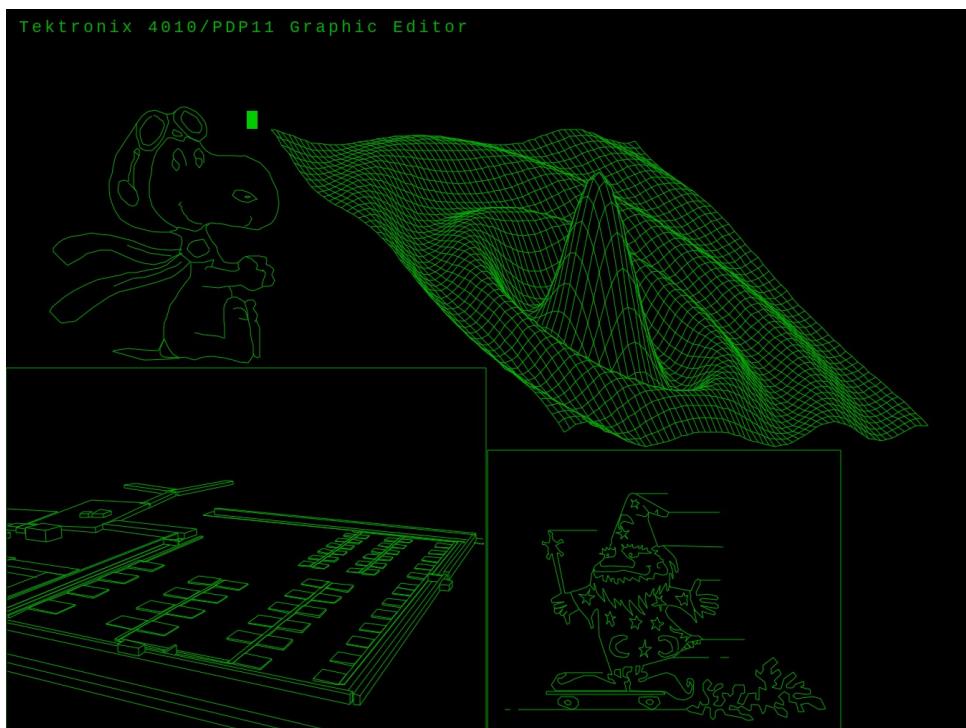
# Tektronix 4010 and 4014 Storage Tube Terminal Emulator

## Table of Contents

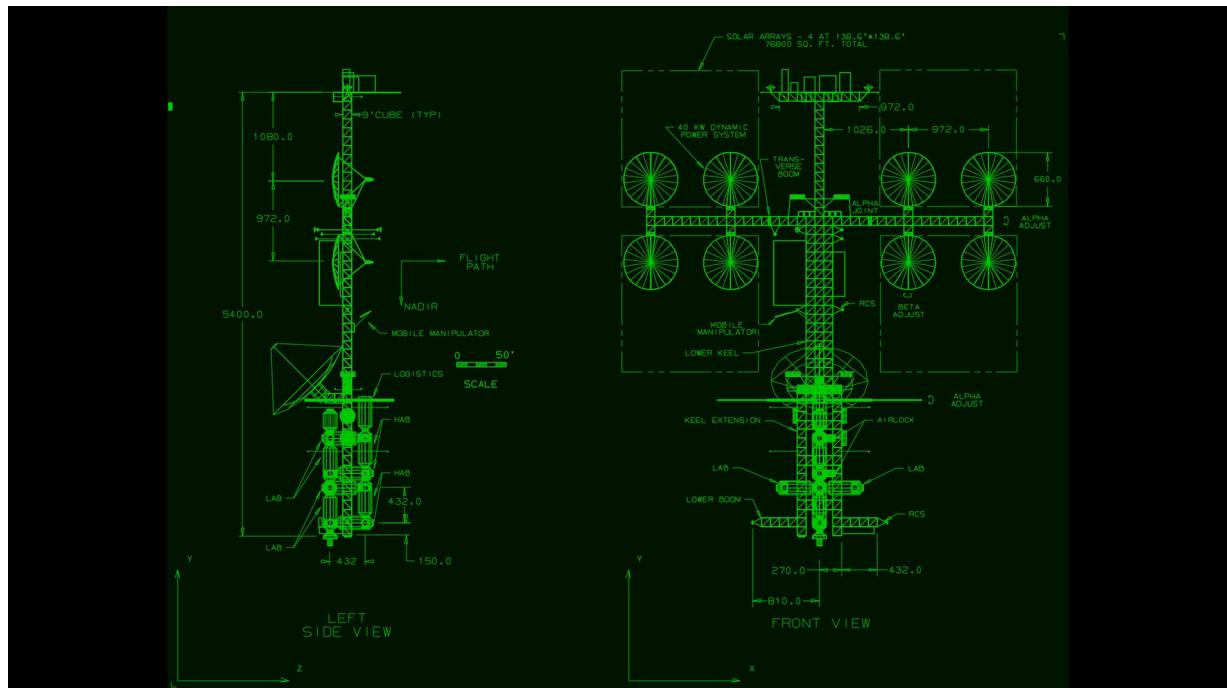
Tektronix 4010 and 4014 Storage Tube Terminal Emulator.....	1
1. Introduction.....	1
2. Important features of tek4010.....	3
3. Installation and first tests.....	4
4. Log directly into a remote historical Unix operating system.....	5
5. Log into the system running simh (same or different Raspberry Pi).....	6
6. Log into PiDP-11 running on the same Raspberry Pi, using the console.....	6
7. Using tek4010 with a serial link.....	7
8. Using tek4010 as a plotting device.....	7
9. Using tek4010 on the PiDP-8.....	7
10. Options of the command tek4010.....	8
11. APL mode.....	8
12. Reporting problems.....	9
13. Screen resolution.....	9
14. Version.....	9
15. Manuals.....	9
16. Historically interesting facts about the ARDS terminal.....	10
17. Using other fonts.....	10
18. Making grayscale images for tek4010.....	10
19. Further examples of using tek4010.....	10
20. Note for existing users of tek4010.....	10
21. Using tek4010 on a slow system.....	11
22. Contributors.....	11
23. The usual disclaimer.....	11

## 1. Introduction

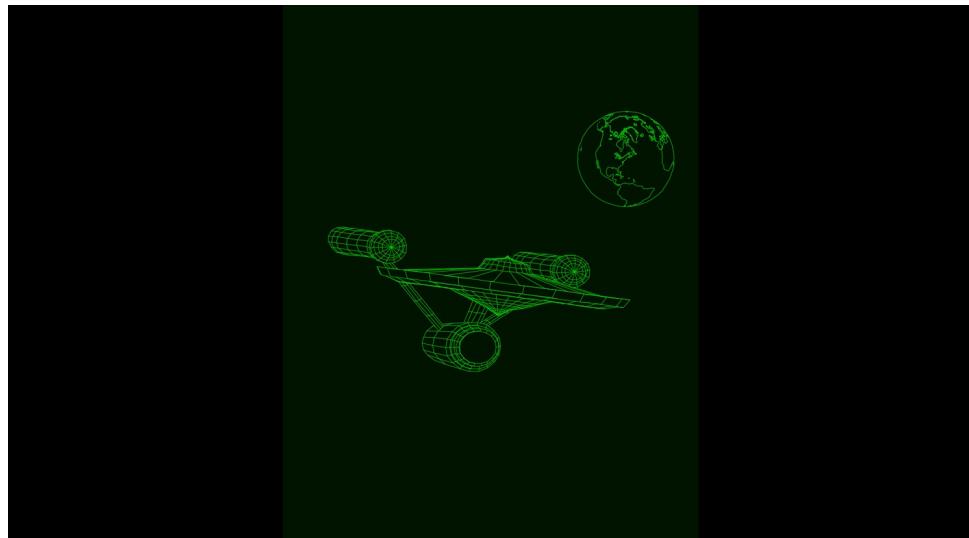
This is a [Tektronix 4010, 4013, 4014 and 4015](#) terminal emulator for the Raspberry Pi and other Linux systems.



Below is a screen shot of the spacelab from the ICEMDDN CAD package made on a CDC Cyber 175 mainframe emulator:



tek4010 can also display historical data for the MIT Project MAC 's ARDS (Advanced Remote Display Station):



tek4010 makes a large effort to emulate the [storage tube display](#) of the Tektronix 4010, including the fading bright drawing spot. If the look and feel is not important, you can use ["xterm"](#) instead. "xterm" does not support all graphics modes of the 4014.

It can be used to log into a historical Unix system such as [2.11 BSD](#) on the [PiDP-11](#) or a real historical system. It can also be used to display historical plot data.

The following picture shows a scale model of the Tektronix 4010 crafted by Dave Ault using tek4010:



## 2. Important features of tek4010

- Emulation of Tektronix 4010, Tektronix 4013, Tektronix 4014, Tektronix 4015 and ARDS
- Emulation of the fading bright drawing spot
- Standard resolution of window 1024 x 780 points
- Scaled smaller resolution for lower resolution screens
- Full screen resolution in -full mode
- Coordinate system: 1024 x 780 and 4096 x 3072 tek points
- All Tektronix 4014 modes, including graphical input mode (GIN) and write-through mode
- Supports grayscale images (Tektronix 4014 intensity chart)
- APL character set and keyboard for Tektronix 4013 and Tektronix 4015
- telnet and rsh connection to host and direct display of plot files
- simulated baud rates: 300 to 19200
- Can be compiled for Raspberry Pi and other Linux distributions such as Ubuntu
- Tested with SimH, PiDP-11, PDP-11/93, VAXstation 4000/90a, 2.11 BSD, RSX-11M+, PLOT10
- GIN mode tested with ICEMDDN CAD package on a CDC Cyber 175 mainframe emulator
- Raspberry Pi 4 Model B+ or faster recommended

### 3. Installation and first tests

**Important:** On Windows and macOS (Macintosh) you need to prepare your system before you can start with the installation as described here. See the files macintosh.txt or windows.txt in this repository.

Install the tek4010 emulator from this repo.

```
cd
sudo apt-get install git
git clone https://github.com/rricharz/Tek4010
cd Tek4010
```

Compile the program on the target system.:

```
cd ~/Tek4010
sudo apt-get install libgtk-3-dev
make clean
make
make install
```

If this is your first installation, reboot the system with

```
sudo reboot
```

**All further commands are executed in the main tek4010 directory:**

```
cd ~/Tek4010
```

There is a file "dodekagon.plt" in the repo, which you can use to test the tek4010 emulator. "dodekagon.plt" was produced in 2.11 BSD using my program "dodekagon". Type

```
tek4010 -noexit cat dodekagon.plt
```

If tek4010 screen refreshes are **too slow or choppy** in this test, use the **-fast** option in all calls to tek4010

```
tek4010 -noexit -fast cat dodekagon.plt
```

Or, for a ARDS display example type

```
tek4010 -noexit -ARDS cat ardsfiles/trek.pic
```

If you want to test text output, type for example

```
tek4010 -noexit head -n 32 src/tek4010.c
```

If you want to test an animation, type

```
tek4010 cat animation.plt
```

Don't forget the option **"-noexit"**, which tells tek4010 to stay alive after cat or head has finished so that you have a chance to look at the output. For a list of all possible options, see the chapter "Options of the command tek4010" below.

If you want to see a demo of historical Tektronix 4014 plot files, type

```
tek4010 demos/demo.sh
```

There are more Tektronix 4014 plot files in pltfiles/More\_pltfiles in this repo. You can find more ARDS plot files at [larsbrinkhoff/ards-files](#) in the folder pictures.

The emulator does use "rsh" or "telnet", because historical Unix systems do not support the secure ssh protocol, and because ssh does not allow using a virtual emulator such as tek4010 for security reasons. You need therefore to install rsh or telnet on the Raspberry Pi or Ubuntu running the tek4010 emulator:

```
sudo apt-get install rsh-client
```

or

```
sudo apt-get install telnet
```

If you want to use this emulator together with 2.11 BSD Unix, look also at [Using the historical Unix 2.11 BSD operating system on the PiDP-11](#)

If you want to use it with RSX-11, see this [discussion](#) to make sure that the “clear screen” escape sequence is properly executed.

## 4. Log directly into a remote historical Unix operating system

This can either be a real historical computer, or a virtual system using simh such as the PiDP-11.

First, you need to test the remote login from your client machine into your historical system, using

```
rsh -l user_name system
```

or

```
telnet system
```

where *user\_name* is the name of the user on the historical operating system, and *system* is the hostname of this system. If the historical operating system is running using an emulator, this is NOT the hostname of the system, on which the emulator is running. See the chapter below if you prefer to log into the system, on which the emulator is running. For example, type

```
rsh -l user_name pdp11
```

or

```
telnet pdp11
```

If this works properly, you can use the tek4010 emulator as follows:

```
tek4010 rsh -l user_name system
```

or

```
tek4010 telnet system
```

If the terminal window is closed right away, there is a problem with your rsh or telnet call. Test it first without tek4010.

The following keys are not transmitted to the Unix system, but are executed locally in the terminal emulator and clear the persistent screen:

```
home
page up
page down
ctrl arrow up
ctrl arrow left
```

These keys emulate the “page” key of the Tektronix 4010. You need to use one of these keys frequently to avoid to get a mess on the screen, as on a real Tektronix 4010.

The **hardcopy function** on the Tektronix 4010 is emulated with a screen dump.

Ctrl-w	Make a screen dump in current directory using scrot Can be typed on the keyboard or sent by the computer during alpha mode
--------	--

Note: On Wayland *scrot* does not work anymore. Instead of *scrot* one can use *grim*. To use *grim* when *ctrl-w* is typed or sent from the host, change the following in *src/main.c*:

```
from      system("scrot -focussed");
to       system("grim");
```

You can use the following ctrl key function to close tek4010:

```
ctrl-q      Close tek4010 window and quit tek4010.
```

## 5. Log into the system running simh (same or different Raspberry Pi)

This makes sense, if you have set up a virtual DZ11 for multiple user login, opening a telnet port for multiplexed terminals. On the PiDP-11 using 2.11 BSD, the distribution software has already set up port 4000 for 8 multiuser terminals. For RSX-11M+ a port at the address 10001 is already set up in the distribution software. Use therefore 10001 instead of 4000 if you are using RSX-11M+. First, you need to install and test telnet (2.11 BSD needs to be up and running in multiuser mode):

```
sudo apt-get install telnet
telnet raspi_hostname 4000      (or 10001 for RSX11M+)
```

or

```
telnet localhost 4000      (or 10001 for RSX11M+)
```

where “raspi\_hostname” is the hostname of your system running simh. If you are using telnet and tek4010 on the same system as the system running simh, use “localhost” instead of “raspi\_hostname”.

Once this works, you can start tek4010 as follows:

```
tek4010 telnet raspi_hostname 4000      (or 10001 for RSX11M+)
```

or

```
tek4010 telnet localhost 4000      (or 10001 for RSX11M+)
```

## 6. Log into PiDP-11 running on the same Raspberry Pi, using the console

This is the least preferred setup, only to be used if you cannot use one of the setups above. You cannot use the tek4010 emulator running screens, as it is done in the standard setup of the PiDP using the console, because screens filters the output stream of simh and is therefore unsuitable for graphics terminals such as the tek4010 emulator. If you don't want to change the standard setup, use ctrl-e to stop simh, and then “exit” to quit simh.

Because tek4010 needs rsh, you need to install rsh-server and rsh-client on the Raspberry Pi. You cannot use telnet here.

```
sudo apt-get install rsh-server
sudo apt-get install rsh-client
```

Now start tek4010 as follows:

```
tek4010 rsh -l pi localhost
```

This should give you a login prompt into your Raspberry Pi. If not, test the rsh call first.

Once your password has been accepted, be prepared to use the “home” key or any of the other keys described above frequently to avoid to get a mess on the dump 4010 terminal emulator! The following will start the PiDP software without using screens:

```
cd /opt/pidp11/bin
./pidp11.sh
```

Everything should run as expected, and you should be able to use the tek4010 terminal emulator with any of the historical operating systems.

**One word of caution!** If you run the PiDP-11 software this way without using screens, you **SHOULD NOT** detach

or quit the terminal while your historical operating system is running, because this will kill the PiDP-11 simh emulator right away. First run down your historical operating system and simh properly, before detaching the terminal emulator.

## 7. Using tek4010 with a serial link

Teunis van Beelen has written a nice [helper program](#) to use tek4010 with a serial link.

First, install the helper program. The helper program is on gitlab, not github.

```
cd
git clone https://gitlab.com/Teuniz/rs232-console
cd rs232-console
make
```

Put a copy of the executable file rs232-console somewhere where your shell will find it, for example in `~/bin` or `/usr/bin`.

Test the serial connection to your host using rs232-console with

```
rs232-console <-p port> <-b baudrate> <-m mode> <-f hardware>
```

Details of the command line parameters of rs232-console can be found in the README file of rs232-console.

Once this works, you can use it with tek4010 as follows:

```
tek4010 [options of tek4010] rs232-console [options of rs232-console]
```

It has happened that the serial port remained locked after tek4010 is quit. I think this has been fixed, but if it still happens, please report details to [richarz77@gmail.com](mailto:richarz77@gmail.com), and use,

```
pkill rs232-console
```

to kill any still running rs232-console process and unlock the serial port.

The serial connection was tested by Dave Ault on a PiDP-11 and a historical PDP-11/73 using RT11. The COPY FILENAME.PLT TT: command was used in RT11 to avoid the filtering of some control codes in TYPE. The tests were made using MX-Linux and a FTDI USB to RS232 converter to connect to the serial port of the PDP-11.

## 8. Using tek4010 as a plotting device

Some simulators do not allow to attach a terminal for standard input and output, but are able to send data to a file. An example of such a simulator is my own simulator based on a home built 6502 system [R65 simulator](#). The original system was not made for attaching a terminal, but could send formatted and raw data to an attached printer or plotter.

tek4010 can be attached to such a simulator as a pure plotter or printer, by monitoring the file created on the fly, and printing or plotting everything sent to that file while it is appended. The command to use in this case is

```
tek4010 tail -f printout.txt
```

where `printout.txt` is the output file created. The file needs to exist when tek4010 is started, and tek4010 displays whatever is already in that file, and keeps monitoring that file and displaying anything which is appended. Note that it is essential that the data is not filtered during the output process, because the tek4010 plotting code is 7 bits binary. Bit 8, the parity bit, is ignored.

## 9. Using tek4010 on the PiDP-8

Details, programs and plot files for the PiDP-8 are available at [tek4010-pidp8](#)

## 10. Options of the command tek4010

Call the command tek4010 using the following syntax:

```
tek4010 [options of tek4010] command [options of command]
```

"command" is a mandatory command to be run by tek4010, such as telnet, rsh or cat.

tek4010 has the following options:

-noexit	do not close window after completion of "command"
-raw	do not execute an automatic CR (carriage return) after a LF (line feed)
-tab1	execute a blank instead of a tab to the next 8-character column
-b100000, -b38400, -b19200, -b9600, -b4800, -b2400, -b1200, -b600, -b300	Emulate a baud rate. Without one of these arguments, the baud rate is 19200 baud. The original Tektronix 4010 had a maximal baud rate of 9600 baud. The 4014 could support up to 100000 baud with a special interface. With the small baud rates you can emulate 1970s style modem performance. Early modems had a baud rate of 300.
-full	in this mode the tek4010 emulator creates a full screen window and uses the full resolution of the 4014 with the enhanced graphics module installed, scaled down to the actual window size. Use ctrl-q to close the tek4010 window.
-fullv	in this mode the tek4010 emulator creates a decorated window using the maximal vertical space available. The full resolution of the 4014 with the enhanced graphics module installed is used, scaled down to the actual window size
-ARDS	display ARDS data
-APL	emulate Tektronix 4013/4015 with alternative APL character set. Details see below.
-autoClear	erase screen if a line feed is executed at the bottom of the screen. This makes it sometimes easier to use tek4010 as the only terminal. It is not the behaviour of the original hardware.
-keepsize	tek4010 sets the fontsize to normal whenever the screen is erased. This option keeps the currently selected font size until it is changed with a new escape sequence. Some historic plot files will not reset the font size back to normal if this option is set.
-hidcursor	hides the cursor. Do not set while using GIN mode.
-wait n	Close window n seconds after completion of "command". It does not make sense to use -noexit together with -wait.
-fast	Use fast rendering without fading. Recommended on slower systems. This mode can be used to avoid a slow and choppy display. -fast is automatically used for baud rates above 19200.
-h, --help	Show options

## 11. APL mode

If tek4010 is called with the -APL argument, a Tektronix 4013 and Tektronix 4015 is emulated with the alternative APL character set. In this mode, the following ctrl keyboard characters are active:

```
ctrl-n Switch to APL character set
ctrl-o Switch to normal character set
```

In this case one can also send “ESC ctrl-n” and “ESC ctrl-o” from the computer to switch the character set. If you want to use the APL mode, you need to install the Apl385 font. Starting from the Tek4010 directory, type

```
cd ~/Tek4010/apl  
sudo install_apl
```

You should see the following line displayed:

```
/usr/share/fonts/truetype/apl385/Apl385.ttf: APL385 Unicode:style=Regular
```

While still being in the apl directory, you can test the APL character set using

```
tek4010 -APL -noexit ./apltest
```

While the second (APL) character set is displayed, it is also possible to translate any key on your keyboard displaying a printable character (ASCII codes between 32 and 127) to any other printable character. The file “aplkeys” in the folder “apl” provides an example of such a conversion table. The first row in this file is the ASCII code of the key, and the second the translated code. If you want to use keys with the “left alt” key, add 128 to the code in the first row of the file. It is even possible to produce overstrike glyphs by adding a second code multiplied by 256 in the second row. You can modify this table to match your keyboard and country code. You can obtain the codes for the first and second row from the screenshot below. To install the “aplkeys” table, type

```
mkdir ~/.tek4010conf  
cp aplkeys ~/.tek4010conf
```

## 12. Reporting problems

If everything works properly for you, but your graphics application produces garbage on the tek4010 emulator, you can send me your data as follows: On Raspberry Pi OS, other Linux systems or a historical Unix system, type

```
your_graphics_program > captured_data.plt
```

I don't know how this can be done on other operating systems. You can then mail your captured\_data file together with a description of the problem to [rricharz77@gmail.com](mailto:rricharz77@gmail.com). Pack it with zip or something else to make sure that the mailing program does not alter it.

If you are registered on github, you can also open an issue.

## 13. Screen resolution

This tek4010 emulator creates a graphics window of 1024x780 points, which is the display size of the Tektronix 4010 terminal and the Tektronix 4014 terminal without enhanced graphics module. The Raspberry Pi can handle sufficiently high refresh rates at this resolution. This emulator executes Tektronix 4014 graphics code with the enhanced graphics module installed, so that such graphics codes can be displayed using this terminal emulator, but the lowest two bits of each axis are not used in this case, as in the Tektronix 4014 without the enhanced graphics module.

If called with the -full or -fully options, the tek4010 emulator uses the full 4K resolution of the 4014 with enhanced graphics module installed, scaled down to the actual window size.

You can use ctrl-q to close the tek4010 window.

The BORDER constant in main.h can be used to adjust the space left for the window decoration and the desktop panel bar when using the -fully option.

## 14. Version

See [versions.txt](#)

## 15. Manuals

The manuals of the historical terminals are available in the [manuals folder](#). This manual of the tek4010 emulator is in the main Tek4010 folder: Manual.pdf

## 16. Historically interesting facts about the ARDS terminal

The ARDS was a pioneering storage tube display terminal. It could already use a 3 button mouse as the graphical input device, as this 1971 thesis shows: Rhine, George Irvin Jr., "A hardware and software interface between a graphics terminal and the SCC 650 computer" (1971), Masters Theses, 5508, page 21, [http://scholarsmine.mst.edu/masters\\_theses/5508](http://scholarsmine.mst.edu/masters_theses/5508)

Detailed pictures of the terminal and the original mouse built by CDI can be found in the 1968 ARDS manual (see manuals folder)

## 17. Using other fonts

The font used by tek4010 is defined in tube.h. If you prefer to use a different font, define STANDARD\_FONT and STANDARD\_FONT\_SIZE in tube.h and recompile the program.

## 18. Making grayscale images for tek4010

The Tektronix 4014 was able to display grayscale images using the "special point plot mode", which was state of the art for the 1970s. The grayscale had only 64 values and was very nonlinear. There are very few such grayscale images still available, but you can make your own images using my tool [make-tek-image](#).



## 19. Further examples of using tek4010

[Moonlight Special MicroMouse Simulator](#)

If you want to show your own example here, please send a note with your link to [rricharz77@gmail.com](mailto:rricharz77@gmail.com)

## 20. Note for existing users of tek4010

tek4010 version 1.8 or later uses a much better simulation of the fading of the bright drawing spot. This requires more computer power. It therefore barely runs on a Raspberry Pi 3. Use a newer Raspberry Pi model or a Linux laptop instead. If you need to run tek4010 on a slow system, see the next paragraph for details on how to revert to version 1.7 of tek4010.

tek4010 runs on several systems requiring different binaries (for example the Raspberry Pi 5 and other Linux systems). **A binary is therefore not supplied anymore.** Compile tek4010 after a download (see chapter 3 for details).

Some files have been rearranged. The source is now in Tek4010/source and the demo scripts in Tek4010/demo.

All instructions are now in this manual (Manual.pdf), which can be found in the main Tek4010 directory.

## **21. Using tek4010 on a slow system**

If you use tek4010 on a slow system such as the Raspberry Pi 3, screen refreshes might be too slow and tek4010 will appear choppy. Call tek4010 with the *-fast* option in this case for a faster screen refresh without fading. *-fast* is automatically used for baud rates above 19200.

## **22. Contributors**

The storage tube emulator and the Tektronix 4010/4014 decoder were written by Rene Richarz. The ARDS decoder was written by Lars Brinkhoff. He also provided some interesting historical documents and the ARDS plot files. Teunis van Beelen has written the helper program “rs232-console” to connect to a host using a serial link. Dave Ault tested the serial link to his PDP-11/73. The historical plot data for the Tektronix 4014 was obtained from Jos Dreesen. The special plot mode pictures with variable brightness were obtained from Monty McGraw. He also helped to debug the special plot point mode. The historical plot data of the spacelab from the ICEMDDN CAD package on a CDC Cyber 175 mainframe emulator was obtained from Nick Glazzard. He also helped to improve the GIN mode substantially. The CP/M GSXBASIC plot files are from Udo Munk. Thanks to Ian Schofield for his critical comments and a code snippet for drawing dashed and dotted lines, and to Oscar Vermeulen and Mark Matlock for their support. The manuals were obtained from [bitsavers.org](http://bitsavers.org). Thanks also to all others who contributed important ideas, helped with the debugging and preserved the historical data. This program is the result of a community effort.

## **23. The usual disclaimer**

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.