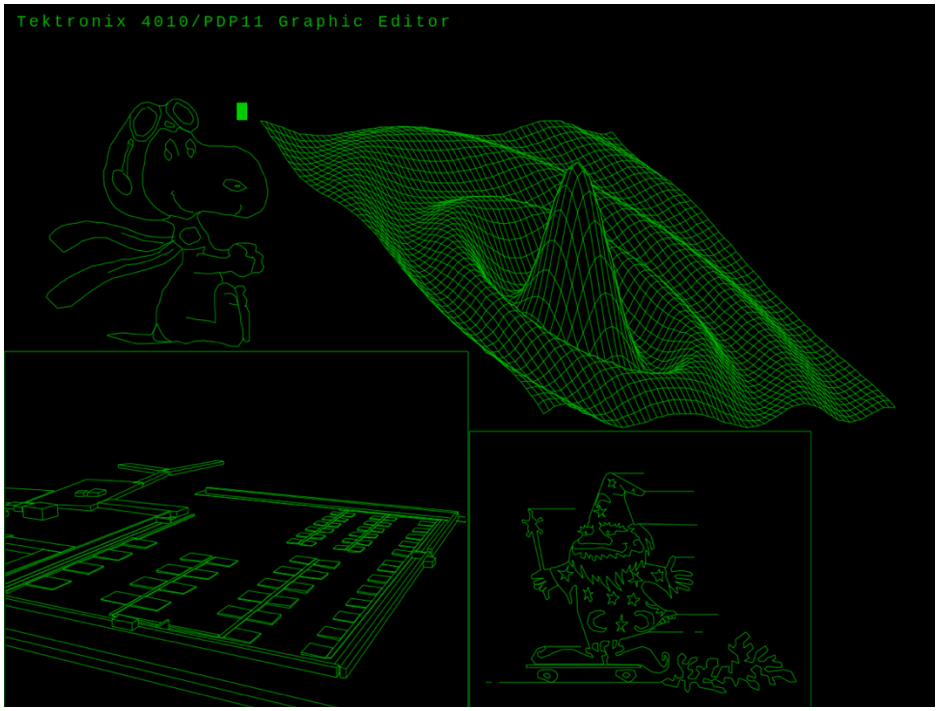
Tektronix 4010 and 4014 Storage Tube Terminal Emulator

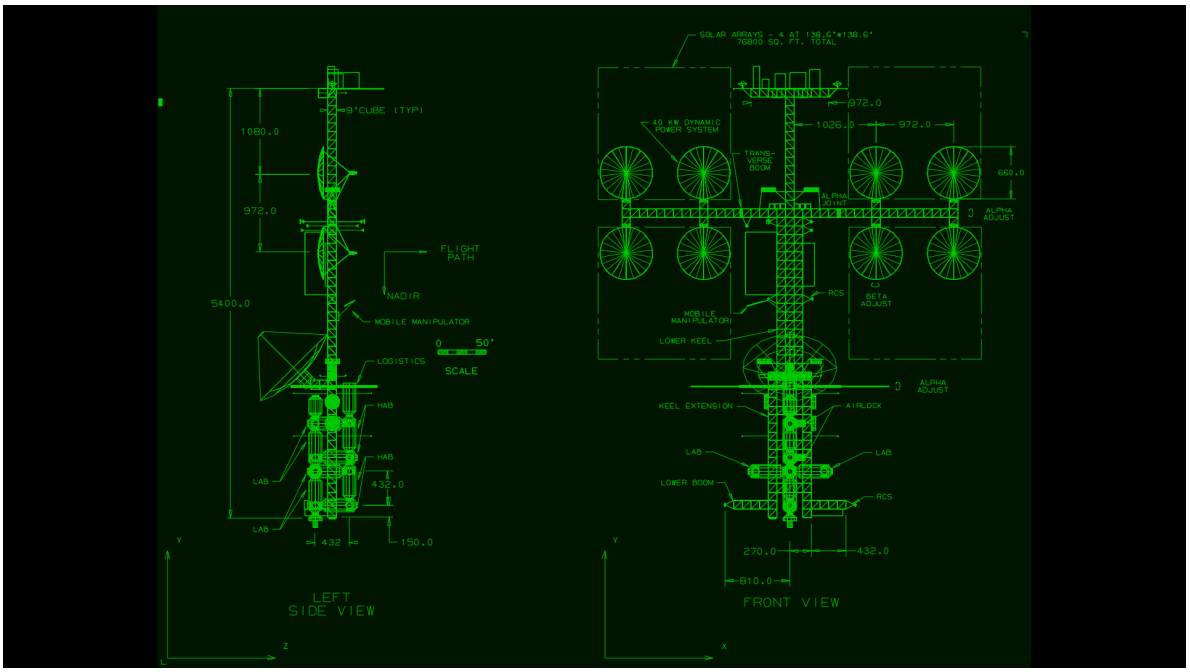
Table of **Contents**

Introduction

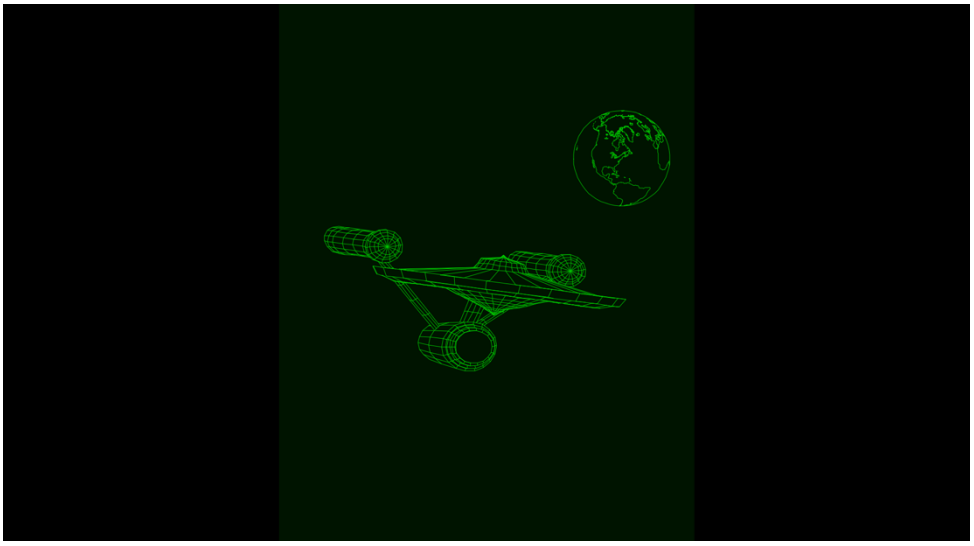
tek4010 is a Tektronix 4010, 4013, 4014 and 4015 terminal emulator for the Raspberry Pi and other Linux systems.



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



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



tek4010 makes a strong effort to reproduce the behavior of the Tektronix storage tube display, including the fading bright drawing spot.

If this look and feel is not important, a standard terminal such as xterm may be used instead. However, xterm does not support all graphics modes of the 4014.

Tek4010 can be used to log into historical Unix systems such as BSD 2.11 on the PiDP-11 or to connect to real historical systems. 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:

Getting started

Important features of tek4010

- Emulation of Tektronix 4010, 4013, 4014, 4015 and ARDS
- Emulation of the fading bright drawing spot
- Standard window resolution of 1024 × 780 pixels
- Scaled resolution for lower-resolution screens
- Full-screen mode with -full option
- Coordinate systems: 1024 × 780 and 4096 × 3072 tek points
- All Tektronix 4014 modes, including graphical input mode (GIN) and write-through mode
- Support for grayscale images (Tektronix 4014 intensity chart)
- APL character set and keyboard for Tektronix 4013 and 4015
- telnet and rsh connection to host systems and direct display of plot files
- Simulated baud rates from 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 the ICEMDDN CAD package on a CDC Cyber 175 emulator
- Raspberry Pi 4 Model B+ or faster recommended

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
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

```
tek401 -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

Connecting to systems

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 getting 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.

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

Log into the system running *SimH* (same or different Raspberry Pi)

This method is useful if a virtual DZ11 has been configured for multiple user logins, providing a *telnet* port for multiplexed terminals.

On the PiDP-11 with 2.11 BSD, the standard distribution already provides port 4000 for up to eight terminals. For RSX-11M+, port 10001 is typically configured instead.

First, install and test telnet (2.11 BSD must be running in multiuser mode):

```
sudo apt-get install telnet
telnet raspi_hostname 4000
```

or

```
telnet raspi_hostname 10001   (RSX-11M+)
```

Alternatively, if tek4010 and SimH run on the same system:

```
telnet localhost 4000
```

or

```
telnet localhost 10001   (RSX-11M+)
```

Here, *raspi_hostname* is the hostname of the system running SimH.

Once this works, start tek4010 as follows:

```
tek4010 telnet raspi_hostname 4000
```

or

```
tek4010 telnet raspi_hostname 10001   (RSX-11M+)
```

or, on the same system:

```
tek4010 telnet localhost 4000
```

or

```
tek4010 telnet localhost 10001   (RSX-11M+)
```

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.

Using tek4010 with a serial link

Using tek4010 with a serial link

A helper program by Teunis van Beelen allows tek4010 to be used over a serial connection.

First, install the helper program (it is hosted on GitLab):

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

Copy the executable rs232-console to a location in your PATH, for example *~/bin* or */usr/bin*.

Test the serial connection using:

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

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

Once this works, use it together with tek4010:

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

In some cases, the serial port may remain locked after tek4010 exits. If this happens, terminate any remaining rs232-console process with:

```
kill rs232-console
```

The serial connection has been tested by Dave Ault using a PiDP-11 and a PDP-11/73 running RT-11. The command

```
COPY FILENAME.PLT TT:
```

was used in RT-11 to avoid filtering of control codes that occurs with TYPE. Tests were performed using MX Linux and an FTDI USB-to-RS232 converter.Using tek4010 as a plotting device.

Using Tek4010 as a plotter

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. 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.

Using tek4010 as a terminal and device

tek4010 on the PiDP-8

Details, programs and plot files for the PiDP-8 are available at tek4010-pidp8

Options of the command tek4010

The command tek4010 is called using the following syntax:

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

The command is mandatory and can be, for example, *telnet*, *rsh* or *cat.*

The following options are available:

-noexit
Do not close the window after the command has finished.

-raw
Do not execute an automatic carriage return (CR) after a line feed (LF).

-tab1
Insert 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. The default is 19200 baud. The original Tektronix 4010 supported up to 9600 baud, while the 4014 could reach 100000 baud with a special interface. Lower baud rates can be used to emulate 1970s modem performance.

-full
Open a full-screen window. The full resolution of the 4014 with the enhanced graphics module is used, scaled to the window size. Use Ctrl + Q to close the window.

-fullv
Open a decorated window using the maximum vertical space available. The full resolution of the 4014 is used, scaled to the window size.

-ARDS
Display ARDS data.

-APL
Emulate a Tektronix 4013/4015 terminal with the APL character set.

-autoClear
Clear the screen automatically when a line feed occurs at the bottom of the screen. This simplifies usage as a primary terminal but does not reflect original hardware behavior.

-keepsize
Keep the current font size when the screen is cleared. Some historical plot files may not reset the font size correctly when this option is used.

-hidecursor
Hide the cursor. Do not use this option in GIN mode.

-wait n
Close the window n seconds after the command has finished. This option should not be combined with -noexit.

-fast
Use fast rendering without fading. Recommended on slower systems. This option is automatically enabled for baud rates above 19200.

-h, --help
Display a list of available options.

Advanced features

APL mode

If tek4010 is started with the -APL option, a Tektronix 4013 or 4015 terminal is emulated using the alternative APL character set.

In this mode, the following key combinations are active:

```
Ctrl + N    switch to APL character set
Ctrl + O    switch to normal character set
```

These commands can also be sent by the host system as ESC Ctrl + N and ESC Ctrl + O.

To use APL mode, the font Apl385.ttf must be installed. Starting from the Tek4010 directory, type

```
cd apl ~/Tek4010/apl
./install_apl
```

On Linux, WSL, and Raspberry Pi OS, the installation script updates the font cache and displays a line similar to

```
Apl385.ttf: “APL385 Unicode” “Regular”
```

This confirms that the font is correctly installed.

On macOS, the font is installed for the current user. You may need to restart tek4010 if it was already running.

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

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

While the APL character set is active, it is possible to remap keyboard input. Any key producing a printable ASCII character (codes 32 to 127) can be translated to another character.

An example mapping file named aplkeys is provided in the apl directory. To install it, type

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

The file is now located at *~/.tek4010conf/aplkeys*. This is the file that is actually used by tek4010. If you want to modify the keyboard mapping, edit this file in *~/.tek4010conf*, not the original file in the apl directory.

The file consists of two rows. The first row contains the ASCII codes of the input keys, and the second row contains the corresponding output codes. Each position defines one mapping.

To use keys together with the left Alt key, add 128 to the value in the first row.

It is also possible to generate overstrike glyphs by adding a second character code multiplied by 256 in the second row.

You can modify the aplkeys file in *~/.tek4010conf* to match your keyboard layout and country-specific settings.

Screen resolution

The tek4010 emulator creates a graphics window with a resolution of 1024 × 780 points. This corresponds to the display size of the Tektronix 4010 and the Tektronix 4014 without the enhanced graphics module.

Modern systems such as the Raspberry Pi can handle sufficiently high refresh rates at this resolution.

The emulator also supports Tektronix 4014 graphics code with the enhanced graphics module. In this case, the full coordinate range is accepted, but the lowest two bits of each axis are not used, as on a 4014 without the enhanced graphics module.

When started with the -full or -fullv options, tek4010 uses the full 4K resolution of the 4014 with the enhanced graphics module, scaled to the actual window size.

Use Ctrl + Q to close the tek4010 window.

The BORDER constant in main.h can be adjusted to control the space reserved for window decorations and desktop panels when using the -fullv option.

Background and documentation

Manuals

Documentation for the original Tektronix terminals is available in the “manuals” folder. This manual for the tek4010 emulator is provided in the *Tek4010/docs* directory as Manual.pdf.

Historically interesting facts about the ARDS terminal

The ARDS was a pioneering storage tube display terminal. It already supported a three-button mouse as a graphical input device, as described in the 1971 thesis:

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

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

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.

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 in the 1970s. The grayscale was limited to 64 levels and was highly nonlinear.

Very few such images are still available today, but new images can be created using the make-tek-image tool.

Using tek4010 on a slow system

On slower systems such as the Raspberry Pi 3, screen refreshes may be too slow and the display may appear choppy.

In this case, start tek4010 with the -fast option to enable faster rendering without fading.

The -fast option is automatically enabled for baud rates above 19200.

Miscellaneous

Contributors

The storage tube emulator and the Tektronix 4010/4014 decoder were written by René Richarz.

The ARDS decoder was written by Lars Brinkhoff, who also provided historical documents and ARDS plot files.

Teunis van Beelen developed the helper program rs232-console for serial connections.

Dave Ault tested the serial link with a PDP-11/73.

Historical Tektronix 4014 plot data was provided by Jos Dreesen.

Special plot mode images with variable brightness were provided by Monty McGraw, who also contributed to debugging this mode.

The spacelab plot data from the ICEMDDN CAD package on a CDC Cyber 175 emulator was provided by Nick Glazzard, who also made substantial improvements to GIN mode.

CP/M GSXBASIC plot files were provided by Udo Munk.

Thanks to Ian Schofield for critical feedback and a code snippet for dashed and dotted lines, and to Oscar Vermeulen and Mark Matlock for their support.

The manuals were obtained from bitsavers.org.

Thanks also to all others who contributed ideas, assisted with debugging, and helped preserve historical data.

This program is the result of a community effort.

The usual disclaimer