

DANTRACKER APRS on a RASPBERRY PI

Revision :Wednesday, August 6, 2014

This document describes the both the installation of the basic Dantracker as per [weblog](#) as well as the extended version by Basil, N7NIX. For the latter you can send me an email to obtain a fully automatic installation script for the Raspian Wheezy distro.

Fresh installation for Raspberry PI Mod II – Debian - Wheezy

Take a **4Gb** sd disk of good quality.....

Install Wheezy as per Raspberry PI website (<http://www.raspberrypi.org/downloads/>)

Install it on the sd-card – and start your raspberry

Follow the onscreen instructions for updating your distro, tailor it to your local needs (Time, keyboard layout etc)

Make sure to change the PI password.... And create also a root password.

Then reboot.

Log-in as user pi

```
✚ ssh pi@10.0.1.13 -X (10.0.1.13 is the ip address of my RPI)
```

Install some prerequisite packages

```
✚ sudo apt-get install screen git python-serial libgtk2.0-dev gtk+-2.0 build-essential gcc  
pkg-config imagemagick automake autoconf libtool cvs curl libncurses-dev libssl-dev
```

```
✚
```

Download either the original dantracker source

```
✚ git clone https://github.com/kk7ds/dantracker
```

```
✚ or the N7NIX branch
```

```
✚ git clone https://github.com/n7nix/dantracker
```

```
✚ cd dantracker
```

Download and install libfab

```
✚ wget http://pakettiradio.net/downloads/libfab/1.3/libfab-1.3.tar.gz
```

```
✚ tar xvzf libfab-1.3.tar.gz
```

```
✚ sudo cp libfab-1.3/src/fap.h /usr/local/include/
```

```
✚ cd libfab-1.3
```

```
✚ The next part in red is only for N7NIX version
```

```
✚ # apply fap patch
```

```
✚ patch -p2 < ./<path_to_tracker_src>/fap_patch.n7nix
```

```
✚ sudo cp src/fap.h /usr/local/include/
```

```
✚ ./configure
```

```
✚ make
```

```
✚ sudo make install
```

```
✚ sudo ldconfig
```

```
✚ NOTE: libraries install to /usr/local/lib
```

```
✚ cd .. (go back to the dantracker directory)
```

Install iniparser - <http://ndevilla.free.fr/iniparser/>

```
✚ sudo wget http://ndevilla.free.fr/iniparser/iniparser-3.1.tar.gz
```

```
✚ tar xvzf iniparser-3.1.tar.gz
```

DANTRACKER APRS on a RASPBERRY PI

```
➤ cd iniparser
➤ sudo cp src/iniparser.h /usr/local/include
➤ sudo cp src/dictionary.h /usr/local/include
➤ make
➤ sudo cp libiniparser.* /usr/local/lib
```

Note – The next part applies to the N7NIX branch only

===== BOF N7NIX INSTALLATION RASPBERRY PI – DEBIAN WHEEZY =====

Install the ax25 library if it has not yet been installed (check by `dpkg -l '*ax25*'`)

Mine shows after installation..

/ Name	Version	Architecture	Description
ii ax25-apps	0.0.8-rc2+cvs20120204-2	armhf	AX.25 ham radio applications
ii ax25-tools	0.0.10-rc2+cvs20120204-3	armhf	tools for AX.25 interface configuration
un ax25-utils	<none>		(no description available)
ii libax25	0.0.12-rc2+cvs20120204-2	armhf	ax25 library for hamradio applications
ii libax25-dev	0.0.12-rc2+cvs20120204-2	armhf	ax25 library development files

If you intend to use soundmodem, do install the ax25 apps and tools as well....

```
cvs -d :pserver:cvs@cvs.linux-ax25.org:/home/ax25-cvs login
(use cvs as password)
```

```
cvs -d :pserver:cvs@cvs.linux-ax25.org:/home/ax25-cvs co libax25
cvs -d :pserver:cvs@cvs.linux-ax25.org:/home/ax25-cvs co ax25-tools
cvs -d :pserver:cvs@cvs.linux-ax25.org:/home/ax25-cvs co ax25-apps
```

Create the following script as `do_ax25.sh` and run it.

```
➤ nano do_ax25.sh

#!/bin/sh
# do_ax25.sh

for dir in libax25 ax25-apps ax25-tools; do
  cd $dir && autoreconf --install --force && \
  ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var --mandir=/usr/share/man
  && \
  make && sudo make install && cd ..
done
```

If this creates an error that it cannot change into one of the directories, do the steps manually in each of the three libraries

```
➤ cd libax25
➤ autoreconf --install --force && ./configure --prefix=/usr --sysconfdir=/etc --
  localstatedir=/var --mandir=/usr/share/man && make && sudo make install && cd ..

➤ cd ax25-tools
```

DANTRACKER APRS on a RASPBERRY PI

- + `autoreconf --install --force && ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var --mandir=/usr/share/man && make && sudo make install && cd ..`
- + `cd ax25-apps`
- + `autoreconf --install --force && ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var --mandir=/usr/share/man && make && sudo make install && cd ..`

Install node.js (npm will be installed in the same time for that matter)

- + `wget http://nodejs.org/dist/v0.10.4/node-v0.10.4.tar.gz`
- + `tar xvzf node-v0.10.4.tar.gz`
- + `cd node-v0.10.4`
- + `./configure`
- + `make`
- + `sudo make install`

Install json-c

- + `wget https://s3.amazonaws.com/json-c_releases/releases/json-c-0.11.tar.gz`
- + `tar -xvzf json-c-0.11.tar.gz`
- + `./configure`
- + `make`
- + `make install`
- + `ldconfig`

Download jquery

- + `wget http://code.jquery.com/jquery-1.9.1.min.js`
- + this file has to be put in the directory `usr/share/dantracker` later on.
- + since it is the latest jquery, you have to adapt the html files in `/usr/share/dantracker.....`
- + `sudo npm -g install ctype`
- + `sudo npm -g install iniparser`
- + `sudo npm -g install webhost`
- + `sudo npm -g install connect`

change back into the dantracker directory

- + `cd ..`

compile

- + `make`
- + if you get the following error
- + `# aprs-ax25.c:24:33: fatal error: netax25/kernel_ax25.h: No such file or directory`
`compilation terminated. make: *** [aprs-ax25.o] Error 1`
- + #then change the `aprs-ax25` file as follows
- + `-> #include <netax25/kernel_ax25.h> becomes #include <netax25/ax25.h>`
- + `make`
- + `sudo ./install.sh`

DANTRACKER APRS on a RASPBERRY PI

```
✚ cp *.js /usr/share/dantracker/
```

Now you copy the jquery file as well. Basil's install script does copy the file also, but I am not sure if he renames the file during copying – to be investigated....

adapt both /usr/share/dantracker aprs_tracker.html as well as spy.html for the correct library

```
</div>
<script src="jquery-1.9.1.min.js"></script>
<script src="spy-frontend.js"></script>
</body>
</html>
```

Edit config files /etc/tracker/aprs_tracker.ini & /etc/tracker/aprs_spy.ini
change [station] mycall, put your range in, adapt the tier2 server name (euro for Europe and noam for Noirth America etc...

Install iceweasel as a web socket compatible browser (Firefox branch) for local viewing if ou view it on the Raspberry screen, other while use another pc on your local network and use the ip address of your Raspberry (<http://ip-of-rapsberry:8080/tracker.html>)

Note:

The install script does not setup apps to start on a power up or reboot. To have the spy & tracker apps startup at boot time do the following.

As root copy tracker startup script to the init.d dir then use udpate-rc.d to add the daemon.

```
✚ su
✚ cd /etc/tracker
✚ cp tracker /etc/init.d/
```

On a Raspberry the script header had to be adapted because the requirement for ax25 to be started did not fit ell. Quick fix:

Edit the file as follows

```
# Required-Start:  $remote_fs $syslog
# Required-Stop:   $remote_fs $syslog
save and exit
```

Add a daemon with sequence of 95

```
✚ update-rc.d tracker defaults 95
```

The daemon script supports start, stop, restart & status
For example as root:

```
✚ /etc/init.d/tracker status
```

To start & stop de Dantracker – N7NIX branch, use /etc/tracker.tracker-up (and down for that matter)

DANTRACKER APRS on a RASPBERRY PI


Make sure you have adapted the aprs_tracker.ini file (and the aprs_spy.ini file for that matter as well) and use ax25 to have all functions working. For testing you can use NET but messages etc will not work.

One handy feature I have done is to have the GPS connected but use NET. When I am driving in the car, the Iphone is setup as an WiFi hotspot and connects to the Raspberry. The range is set for the range of my VHF rigs with repeaters, approx. 60 km

Here is a screen picture after about 11 minutes of operation in that mode.

This is the latest version with the tabbed screen in the browser.


APRS **MSG** **CTRL**

PI1SGM-2 26.3km S via TCPIP 

Telemetry #196


12 1 46 9 4 00000000

1:PA0ESH 0km E	5:PA3CXB-B 11km SE
2:PI1TWE-2 2km E	6:PA3CXB B 11km SE
3:PD0ADC-C 43km N	7:ER-DB0BOR 3m53s
4:EL-DB0BOH 47km S	8:PI1ZUT 37km W

Stationary, Alt 123 m 

52.25900N 6.75500E 01:00:00 Locked: 0 sats

PA0ESH-7 : ATREST

PA3DZX-13 9.9km E (11m50s ago) 

Wind NE 4/7m/s 23C 49%

Twente (bron: BuienRader.NL <http://www.buienradar.nl>)

Recv	Xmit	Retry	Messages	Weather	Encap Pkts	Error
65	0	0	3	1	0	39

DANTRACKER APRS on a RASPBERRY PI

===== EOF N7NIX installation =====
This is the part for the original dantracker again...

Change back into the dantracker directory

```
➤ cd ..
```

```
➤ $ touch .revision
```

```
➤ # needed because I did not clone drats you may have to add it later to the startscript as well, as it did not stay on my Raspberry after a reboot....
```

```
➤ Note – this may be necessary for the N7NIX branch as well.
```

Now execute the compilation

```
➤ make
```

Do the same for the images (not required for N7NIX branch)

```
➤ cd images
```

```
➤ make
```

DANTRACKER APRS on a RASPBERRY PI

cd ..

This is all and if you had no errors, then the test will show that it works.

Now the test

First check for the gps

```
➤ sudo ./detect_gps.py
```

In my case, I have a unknown GPS mouse and a TNC-X with FTDI USB to serial port
The output shows up as:

```
pi@raspberrypi ~/dantracker-ori $ sudo ./detect_gps.py
lrwxrwxrwx 1 root root 12 Apr 23 19:40 /dev/gps -> /dev/ttyUSB1
lrwxrwxrwx 1 root root 12 Apr 23 19:40 /dev/radio -> /dev/ttyUSB0
```

Copy the aprstest.ini file from the samples directory into your aprs directory and rename the call to your own call and change the position to some where near your home.
This will deliver a lot of data for the stress test

Now, run a screen session for the aprs function as follows

```
➤ screen -dmS aprs ./aprs -m /home/pi/dantracker/aprs.ini
```

The aprs job is now up and running on your raspberry in a (hidden) screen.

next start ui with the -i option

```
➤ xinit ./ui
```

That's it. On my screen it looks as in the youtube movie.

Now adapt the aprs.ini file from the samples directory to your real situation. Then have fun with the dantracker, like I had.

In case you wish to autostart the whole thing do as follows:

Create a startup file in the directory /etc/init.d, called dantracker

```
nano /etc/init.d/dantracker
```

Write the following into the file:

```
#!/bin/sh
# /etc/init.d/noip

### BEGIN INIT INFO
# Provides:      dantracker
# Required-Start: $remote_fs $syslog
# Required-Stop:  $remote_fs $syslog
# Default-Start: 2 3 4 5
# Default-Stop:  0 1 6
```

DANTRACKER APRS on a RASPBERRY PI

```
# Short-Description: Simple script to start the original dantracker at boot
# Description:      A simple script by pa0esh to start the original dantracker programs at boot in
two screens. PA0ESH @21 April 2013
### END INIT INFO
```

```
# If you want a command to always run, put it here
```

```
# Carry out specific functions when asked to by the system
case "$1" in
```

```
start)
```

```
    echo "Starting the Dantracker"
```

```
    # run application you want to start and change the working dir according to your system
```

```
    cd /home/pi/dantracker
```

```
    screen -dmS aprs ./aprs -m
```

```
    screen -dmS ui xinit /home/pi/dantracker-ori/ui
```

```
    ;;
```

```
stop)
```

```
    echo "Stopping the Dantracker"
```

```
    # kill application you want to stop
```

```
    killall screen
```

```
    killall ui
```

```
    ;;
```

```
*)
```

```
    echo "Usage: /etc/init.d/dantracker {start|stop}"
```

```
    exit 1
```

```
    ;;
```

```
esac
```

```
exit 0
```

```
sudo chmod 755 /etc/init.d/dantracker
```

To register your script to be run at start-up and shutdown, run the following command:

```
sudo update-rc.d dantracker defaults
```

If you ever want to remove the script from start-up, run the following command:

```
sudo update-rc.d -f dantracker remove
```

73's

Erik, PA0ESH

DANTRACKER APRS on a RASPBERRY PI

Somewhere in the file `aprs-is.c` I found data pertaining to the login for Dan I have changed that to my own data, and you should do so as well.

This is the part you are looking for.....

```
int sock;
double lat = 52.259;
double lon = 06.755;
double range = 1500;
int ret;
char buf[256];

sock = aprsis_connect("europe.aprs2.net", 14580, "PA0ESH-5",
                    lat, lon, range);
if (sock < 0) {
    printf("Sock %i: %m\n", sock);
    return 1;
}
```

After modifying execute make command...

DANTRACKER APRS on a RASPBERRY PI

Alternative 2

WIP: 24 April 2013

This document describes the installation of N7NIX branch of Dantracker, but using the Archlinux image....

Archlinux is a minimalistic package and could fit /will fit on a 2Gb sd card.

Step 1:

Fresh installation on Raspberry PI with Archlinux

Create a SD card with the latest Archlinux image

Login to the RPI as root (password root)

Change the password for root user

http://elinux.org/RPi_Easy_SD_Card_Setup

Step 2:

Prerequisites for the next step creating a normal user

```
pacman -S sudo adduser
```

add a normal user and put it into the sudoers group

```
adduser <name_of_user> and follow the instructions on screen
```

```
sudo nano /etc/suoders
```

Step 3:

```
sudo pacman -S gtk gtk-engines git cvs kernel26-headers file base-devel abs autoconf  
automake libtool ncurses python python-serial
```

(note: some of the packages may have been installed in the basic archlinux image – they will be reinstalled.)

Some of the packages are not in the official Archlinux repository, but can be installed from AUR by means of the yaourt programme

```
sudo pacman -S yaourt  
yaourt -Syy
```

DANTRACKER APRS on a RASPBERRY PI

The following packages are installed through yaourt.

Because of the architecture of the Raspberry, you have to edit the config files during installation by adding / replacing the current arch with `[arch=('i686' 'x86_64' 'armv6h')]`

The packages are”

- libax25
- ax25-tools
- ax25-apps
- soundmodem

example:

- yaourt -S libfab
- reply yes the first time to edit the config file and then add `arch=('i686' 'x86_64' 'armv6h')`
- (for the other packages replace the existing arch..... by `arch=('i686' 'x86_64' 'armv6h')`)

For json-c, node and libfab obtain the tarball at their websites and then install according to the attached RWSDME files

Cd into the dantracker N7NIX branch
(cd dantracker or wherever you have chosen)

```
✚ wget https://s3.amazonaws.com/json-c\_releases/releases/json-c-0.11.tar.gz
✚ tar -xzf json-c-0.11.tar.gz
✚ ./configure
✚ make
✚ make install
✚ ldconfig
```

```
✚ sudo wget http://pakkettiradio.net/downloads/libfab/1.3/libfab-1.3.tar.gz
✚ tar xvzf libfab-1.3.tar.gz
✚ sudo cp libfab-1.3/src/fap.h /usr/local/include/
✚ cd libfab-1.3
✚ make
✚ sudo make install
```

```
wget http://nodejs.org/dist/v0.10.4/node-v0.10.4.tar.gz
```

One last file need to be downloaded and copied to the `/usr/share/` directory
<http://jquery.com/download/>

```
wget http://code.jquery.com/jquery-1.9.1.min.js
```

Step 4: The final

Make sure that your shared libraries are linked:

DANTRACKER APRS on a RASPBERRY PI

```
sudo ldconfig /usr/local/lib
```

Finally change the Makefile in /dantracker and add `-lm` to the gcc flags

```
CFLAGS = -g -Wall -Werror -lm
```

```
# added -lm on Raspberry Arch Linux architecture /remove for Debian.
```

```
LIBS = -lfap -liniparser
```

Problems to be solved

```
In -s /usr/lib/node_modules/connect ./ctype
```

```
In -s /usr/lib/node_modules/connect ./websocket
```

```
In -s /usr/lib/node_modules/connect ./iniparser
```

```
In -s /usr/lib/node_modules/connect ./connect
```