6pack Protocol

This is the 6pack-mini-HOWTO, written by

Andreas KÃ¶nsgen DG3KQ

Internet: ajk@comnets.uni-bremen.de
AMPR-net: dg3kq@db0pra.ampr.org
AX.25: dg3kq@db0ach.#nrw.deu.eu

Last update: April 7, 1998

1. What is 6pack, and what are the advantages to KISS?

6pack is a transmission protocol for data exchange between the PC and the TNC over a serial line. It can be used as an alternative to KISS.

6pack has two major advantages:

- The PC is given full control over the radio channel. Special control data is exchanged between the PC and the TNC so that the PC knows at any time if the TNC is receiving data, if a TNC buffer underrun or overrun has occurred, if the PTT is set and so on. This control data is processed at a higher priority than normal data, so a data stream can be interrupted at any time to issue an important event. This helps to improve the channel access and timing algorithms as everything is computed in the PC. It would even be possible to experiment with something completely different from the known CSMA and DAMA channel access methods. This kind of real-time control is especially important to supply several TNCs that are connected between each other and the PC by a daisy chain (however, this feature is not supported yet by the Linux 6pack driver).
- Each packet transferred over the serial line is supplied with a checksum, so it is easy to detect errors due to problems on the serial line. Received packets that are corrupt are not passed on to the AX.25 layer. Damaged packets that the TNC has received from the PC are not transmitted.

More details about 6pack are described in the file 6pack.ps that is located in the doc directory of the AX.25 utilities package.

2. Who has developed the 6pack protocol?

The 6pack protocol has been developed by Ekki Plicht DF4OR, Henning Rech DF9IC and Gunter Jost DK7WJ. A driver for 6pack, written by Gunter Jost and Matthias Welwarsky DG2FEF, comes along with the PC version of FlexNet. They have also written a firmware for TNCs to perform the 6pack protocol (see section 4 below).

3. Where can I get the latest version of 6pack for LinuX?

At the moment, the 6pack stuff can obtained via anonymous ftp from db0bm.automation.fh-aachen.de. In the directory /incoming/dg3kq, there is a file named 6pack.tgz.

4. Preparing the TNC for 6pack operation

To be able to use 6pack, a special firmware for the TNC is needed. The EPROM of a newly bought TNC does not contain 6pack, so you will have to program an EPROM yourself. The image file for 6pack EPROMs should be available on any packet radio box where PC/FlexNet can be found. The name of the file is 6pack.bin. This file is copyrighted and maintained by the FlexNet team. It can be used under the terms of the license that comes along with PC/FlexNet. Please do not ask me about the internals of this file as I don't know anything about it. I used a textual description of the 6pack protocol to program the Linux driver.

TNCs contain a 64kByte EPROM, the lower half of which is used for the firmware/KISS. The upper half is either empty or is sometimes programmed with software called TAPR. In the latter case, the TNC is supplied with a DIP switch so you can easily change between the two systems. When programming a new EPROM, one of the systems is replaced by 6pack. It is useful to replace TAPR, as this software is rarely used nowadays. If your TNC is not equipped with the switch mentioned above, you can build in one yourself that switches over the highest address pin of the EPROM between HIGH and LOW level. After having inserted the new EPROM and switched to 6pack, apply power to the TNC for a first test. The connect and the status LED are lit for about a second if the firmware initialises the TNC correctly.

5. Building and installing the 6pack driver

The driver has been tested with kernel version 2.1.90. Use with older kernels may lead to a compilation error because the interface to a kernel function has been changed in the 2.1.8x kernels.

How to turn on 6pack support:

- In the linux kernel configuration program, select the code maturity level options menu and turn on the prompting for

development drivers.
- Select the amateur radio support menu and turn on the serial port 6pack driver.
- Compile and install the kernel and the modules.

To use the driver, the kissattach program delivered with the AX.25 utilities has to be modified.

- Do a cd to the directory that holds the kissattach sources. Edit the kissattach.c file. At the top, insert the following lines:

```
#ifndef N_6PACK
#define N_6PACK (N_AX25+1)
#endif
```

  Then find the line:

```
int disc = N_AX25;
```

  and replace N_AX25 by N_6PACK.

- Recompile kissattach. Rename it to spattach to avoid confusions.

Installing the driver:

- Do an insmod 6pack. Look at your /var/log/messages file to check if the module has printed its initialization message.
- Do a spattach as you would launch kissattach when starting a KISS port. Check if the kernel prints the message '6pack: TNC found'.
- From here, everything should work as if you were setting up a KISS port. The only difference is that the network device that represents the 6pack port is called sp instead of sl or ax. So, sp0 would be the first 6pack port.

Although the driver has been tested on various platforms, I still declare it ALPHA. BE CAREFUL! Sync your disks before insmoding the 6pack module and spattaching. Watch out if your computer behaves strangely. Read section 6 of this file about known problems.

Note that the connect and status LEDs of the TNC are controlled in a different way than they are when the TNC is used with PC/FlexNet. When using FlexNet, the connect LED is on if there is a connection; the status LED is on if there is data in the buffer of the PC's AX.25 engine that has to be transmitted. Under Linux, the 6pack layer is beyond the AX.25 layer, so the 6pack driver doesn't know anything about connects or data that has not yet been transmitted. Therefore the LEDs are controlled as they are in KISS mode: The connect LED is turned on if data is transferred from the PC to the TNC over the serial line, the status LED if data is sent to the PC.

6. Known problems

When testing the driver with 2.0.3x kernels and operating with data rates on the radio channel of 9600 Baud or higher, the driver may, on certain systems, sometimes print the message '6pack: bad checksum', which is due to data loss if the other station sends two or more subsequent packets. I have been told that this is due to a problem with the serial driver of 2.0.3x kernels. I don't know yet if the problem still exists with 2.1.x kernels, as I have heard that the serial driver code has been changed with 2.1.x.

When shutting down the sp interface with ifconfig, the kernel crashes if there is still an AX.25 connection left over which an IP connection was running, even if that IP connection is already closed. The problem does not occur when there is a bare AX.25 connection still running. I don't know if this is a problem of the 6pack driver or something else in the kernel.

The driver has been tested as a module, not yet as a kernel-builtin driver.

The 6pack protocol supports daisy-chaining of TNCs in a token ring, which is connected to one serial port of the PC. This feature is not implemented and at least at the moment I won't be able to do it because I do not have the opportunity to build a TNC daisy-chain and test it.

Some of the comments in the source code are inaccurate. They are left from the SLIP/KISS driver, from which the 6pack driver has been derived. I haven't modified or removed them yet -- sorry! The code itself needs some cleaning and optimizing. This will be done in a later release.

If you encounter a bug or if you have a question or suggestion concerning the driver, feel free to mail me, using the addresses given at the beginning of this file.

Have fun!

Andreas