REINER SCT cyberJack pinpad/e-com USB chipcard reader driver

Matthias Brüstle

Harald Welte

Martin Preuss

$Date$

This is the user manual to the linux driver for REINER SCT cyberjack chipcard readers.

1. Overview

This driver for the REINER SCT cyberJack pinpad/e-com USB family of chipcard readers implements the CT-API 1.1 interface, as well as the PC/SC interface of pcsc-lite.

It is implemented 100% in userspace. This means no trouble with different kernel versions, compiling/patching the kernel etc.

All accesses are done via the `usb devfs` in `/proc/bus/usb` (or `/dev/bus/usb` for udev-based systems).

Permission handling is done *only* via udev. The `cyberjack.rules` if installed to `/etc/udev/rules.d` is automatically called by udev when a reader is plugged in. This scripts dynamically updates the permissions of the respective device, so users in the group `cyberjack` are able to access it.

For more information about the smart card reader itself please visit http://www.reiner-sct.com/. There is also a shop where the the readers can be ordered online.

2. Readers supported by this driver

The following Reiner-SCT readers are supported:

Product	ProductID
REINER SCT cyberJack pinpad USB	0x100
REINER SCT cyberJack e-com USB	0x100
REINER SCT cyberJack pinpad_a USB	0x300

You can use the **lsusb** command to list all devices connected to the USB bus of your machine. It will print out the vendor and device ID of all your devices, like :

```
Bus Nr Device Nr VeID:PrID Bus 002 Device 002: ID 0451:1446 Texas
Instruments, Inc. TUSB2040/2070 Hub Bus 002 Device 006: ID 0c4b:0300
```

The REINER SCT VendorID is `0c4b`. ProductID's can be looked up in the table above.

3. distribution-specific notes

You find all packages at http://www.reiner-sct.com/content/view/32/43/#linux.

Most packages need you to add the user which is to access the card reader to be added to the group *cyberjack*. This can be done by using the tool *kuser* or the system's administration tool (e.g. *yast* on SuSE).

You should reboot your machine after installing the driver package and adding the user to the group *cyberjack* in order for the changes to take effect.

3.1. RPM-based

Reiner-SCT provides RPM packages for the following distributions:

- SuSE 10.2
- SuSE 10.1
- SuSE 10.0
- SuSE 9.3
- Fedora Core 5
- Fedora Core 4

Just install the package corresponding to your system like this: **`rpm -i <package file>`**

If you already have one of Reiner-SCT's previous RPM packages installed you must update that existing package instead, like in: **`rpm -U <package file>`**

3.2. DEB-based

Reiner-SCT provides RPM packages for the following distributions:

- Debian unstable
- Ubuntu 6.06
- Ubuntu 6.10

Just install the package corresponding to your system like this: **`dpkg -i <package file>`**

3.3. All other Distributions

There is currently no experience with other Linux distributions. It should work in most cases as described above. If you get any problems with the RPM package, you can try to rebuild it on your system with **`rpm --rebuild <source package file>`** or **`rpmbuild --rebuild <source package file>`**

If you want to compile the source yourself just go into the main directory of the extracted archive and type **`./configure make`**

The include file `ctapi.h` and the resulting library `libctapi-cyberjack.so` from the directory `ctapi/` can then be copied to convenient places. For `ctapi.h` this would normally be `/usr/include` and for the library `/usr/lib`. The command **make install** can do that for you.

The name scheme `libctapi-cyberjack.*` has been choosen to make it possible to install more than one CT-API library on your system.

4. Support

Support of this driver is provided by REINER SCT. E-mail: support@reiner-sct.com Postal address: Schwabacher Str. 34, 90762 Fürth, GERMANY

In your problem description, please include as far as possible:

- Any error messages you get.
- Which Linux distribution you use including version, e.g. SuSE 10.1, Debian 3.0r1 testing, ...
- CPU type, e.g. on Linux the content of the file `/proc/cpuinfo`.
- Kernel version, e.g. on Linux the output from the command **uname -r**.
- List of attached USB devices, e.g. on Linux the output of the **lsusb** command.

5. Troubleshooting

5.1. How to check the kernel version

You can determine the version of the currently running kernel by executing **`uname -r`**

The version of the installed kernel sources, which are normally located below `/usr/src`, can be determined by looking at the source directory name or by looking into the main Makefile, where it is in the first three lines.

5.2. Large number of readers

The cyberJack has been tested with up to 52 devices attached simultaneously to a single PC via 7-port hubs. Some notes regarding this configuration:

- Linux at least up to 2.4.19 does result in a kernel panic, when too many devices are attached. Known to work is 2.4.20.
- Sometimes timeouts occur resulting in a shift of the T=1 blocks resulting in bad performance and sooner or later a failure of communcation. The problem seems to lie somewhere in the usb-uhci part and vanishes with a faster PC. (Try >2GHz)
- If there are still some problems try other hubs and other USB host controller cards. There seems to be a great difference in quality in these parts.

The performance does not degrade, when going from 1 up to 50 readers, even when doing constant I/O with cards. (Select and Read Binary)

5.3. Hotplugging

Linux supports hotplugging with USB devices. This is implemented via the udev-system.

You can find some udev scripts for the REINER SCT cyberjack reader family in the `etc/udev` directory of this archive.

Since udev-related scripts are highly distribution specific, REINER SCT can only provide limited support in this area. The provided RPM and Debian packages install those scripts to their respective places.

5.4. Logging

The cyberjack CT-API library supports logging of the communication with the reader. This is done, if at the moment `CT_init` is called the environment variable CJDEBUG exists. The default output file is `/tmp/cj.log`. The logging is done on T=1 level and each entry begins with a time stamp.

6. Known Issues

Unfortunately, all Linux kernel versions, at least up to (including) 2.6.12-rc5 have a severe bug in the handling of asynchronous URB’s (USB Request Blocks) in userspace. This bug is totally unrelated to the REINER-SCT cyberjack driver, but it will show as soon as the PC/SC daemon terminates (and you’re using a pinpad_a (0x300) reader. The bug can crash your kernel :(.

A bugfix has been developed (but not yet included into the mainline kernel). It is available as kernel patch in ’patches/usb-async_urb-devio-oops-fix.patch’.

It is strongly recommended to apply this kernel patch if you intend to use the PC/SC driver.

7. Additional Information

7.1. Beeping at Keypress

Starting with Version 2.0.5 of ctapi-cyberjack, the host PC will emit a beep sound at every key press. The driver tries to detect the best mechanism for beeping by itself, i.e. xBell when you run under X11, or sending a BEL ASCII character to STDOUT when running as a console application.

If you want to disable the beep, you can set the `CJCTAPI_NO_KEYBEEP` environment variable before starting your application.

Depending on your shell, this can be achieved with a command like **export CJCTAPI_NO_KEYPRESS**.

7.2. Mandatory locking

Normal locking is only advisory, i.e. the programs must be cooperative to do the locking properly. A non-cooperative program can ignore a lock and access the reader. Mandatory locking, which stops even a malicious program from access the reader when it is locked, requires setting special permissions of the device node.

From `linux/Documentation/mandatory.txt`: “ A file is marked as a candidate for mandatory locking by setting the group-id bit in its file mode but removing the group-execute bit. This is an otherwise meaningless combination, and was chosen by the System V implementors so as not to break existing user programs. ”

7.3. Permissions

If a normal user should be able to access and use the cyberJack chipcard reader, the permissions should be ’2666’. The ’2’ enables the mandatory locking described in the section before. The ’666’ enables read/write for all users.

7.4. CT-API

The CT-API specification can be downloaded at http://www.darmstadt.gmd.de/~eckstein/CT/mkt.html

Please note, that the port numbers start with one. This behaviour is specified in the CT-API documentation.

7.5. PC/SC

This driver package now contains a working PC/SC driver for pcsc-lite. The driver was tested with pcsc-lite-1.2.0 up to 1.3.1.

7.5.1. Installation

If you're installing the driver via a pre-built RPM package, make sure you install the "ctapi-cyberjack-ifd-handler" package.

If you're building the driver from source code, make sure you install the `pcsc/ifd-cyberjack.bundle` directory to the "usb plugdir" directory of your pcsc-lite installation. The default **make install** procedure puts it into `/usr/lib/pcsc/drivers/`.

7.6. Multithreading

The library is NOT save against multiple threads accessing at the same time the same reader. This gives you also most probably problems with your card anyway.

The library is save against multiple threads accessing multiple readers. So you could start 3 threads, each accessing their own card in their own reader.

7.7. command size

The command size is currently limited to ISO7816 short commands.

7.8. Keypressed callback

```
IS8 rsct_setkeycb(IU16ctn, void (*cb) (void *user_data));
```

The function `rsct_setkeycb` has been added to specify a callback to signal keypresses. The function specified in the second parameter is called whenever a C4 or F4 S-block is received from the reader. This information can be used to help the user, when entering a PIN on the cyberJack pinpad reader, which does not show how many keys have been pressed.

7.9. Obtaining Version Info

```
void rsct_version(IU8*vmajor, IU8*vminor, IU8*vpatchlevel, IU16*vbuild);
```

The function `rsct_version` returns the complete version of the driver.

7.10. Additional CT_init Replacement Function

```
IS8 rsct_init_name(IU16ctn, const char*device_name);
```

The function `rsct_init_name` can be used to directly specify the device to be used. The device name is specified like for PC/SC drivers: "usb:VENDOR_ID/PRODUCT_ID:libusb:BUS_ID:DEVICE_ID", so for a new cyberjack at /proc/bus/usb/003/002 the correct name would be:
"usb:0c4b/0300:libusb:003:002"

7.11. Verifying Pins Using PC/SC Function SCardControl

The following table shows values for the PSCS_VERIFY_STRUCTURE object which have been tested with ASCII and FPIN2 formatted pins.

Field	**ASCII**	**FPIN2**
bTimerOut	00	00
bTimerOut2	00	00
bmFormatString	82	81
bmPINBlockString	04	48
bmPINLengthFormat	00	04
wPINMaxExtraDigit	0408	0408
bEntryValidationCondition	02	02
bNumberMessage	01	01
wLangId	0904	0904
bMsgIndex	00	00
bTeoPrologue 0-2	00	00