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

Matthias Brüstle

Harald Welte

$Date$

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 pcec-lite.

Depending on your particular device, the driver either consists of a kernel and userspace part, or is implemented completely in userspace.

The following table provides an overview about the current situatiion:

Product	ProductID	Kernel	Userspace
REINER SCT cyberJack pinpad USB	0x100	yes	old
REINER SCT cyberJack e-com USB	0x100	yes	old
REINER SCT cyberJack pinpad_a USB	0x300	no	new

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.

1.1. How to check the hardware version of the reader?

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.

2. REINER SCT cyberJack pinpad/e-com USB (0x100)

Important: This section is only relevant for devices with ProductID `0x100`!

2.1. Kernel part

2.1.1. 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.

The kernel part is included in the official linux kernels starting with version 2.4.6.

2.1.2. LINUX Kernel version 2.6.0 and up

The Linux 2.6.x kernel series comes with numerous changes in the USB subsystem. The most noticeable are the new host controller drivers for UHCI, which cause way less problems for cyberjack than in the past with certain 2.4.x kernels.

Unfortunately, the Linux Kenrels 2.6.3 to 2.6.7 ship with a broken `cyberjack` driver, so you will have to apply a kernel patch provided in the `patches/2.6.x/` directory.

Kernels 2.6.8 and later include a working cyberjack driver. No additional action is required by you.

2.1.3. LINUX Kernel version 2.4.6 and up

If you have a distribution, which comes with a running 2.4.6 or above kernel, you probably don’t have to mess with the kernel. The correct drivers should be loaded upon insertion of the reader.

There exist two different USB host controllers on the PC:

- UHCI
- OHCI

The first is the more common. Details about this is written in the next few paragraphs. The second is less common and has not received as much testing. If you have an OHCI host controller you can ignore what is written here about the UHCI drivers.

The Linux kernel comes with two different drivers for the UHCI USB host controller. You have to make sure, that you use not the uhci driver, but the usb-uhci driver has been loaded. If the driver is loaded as a module, you can check the driver type by starting the programm **lsmod**. It prints something like:

```
Module Size Used by ppp_async 6348 1 (autoclean) ppp_generic 13084 3
(autoclean) [ppp_async] usb-uhci 21800 0 (unused) wvlan_cs 24816 1 8139too
11820 1 (autoclean) de4x5 42960 1 (autoclean)
```

You can see there the usb-uhci module loaded, which is correct. If there is no usb-uhci or uhci, it is either no USB support loaded - try inserting a USB device - or USB is statically linked to the kernel. You have to watch here for USB related output at boot time or shortly after a boot by typing **dmesg** at the shell prompt.

usb-uhci was the first driver for an UHCI USB host controller in Linux, but is not well structured. uhci is a totally rewritten driver, but has unfortunatelly still problems with the usb-serial line of drivers and therefor the cyberJack will not with with it. How to select the correct driver depends on the Linux distribution. (See e.g. Section)

If you have to compile and install a kernel for yourself, please look into the manual of your Linux distribution, how this is done properly. E.g. in SuSE Linux 7.2 it is described in the reference manual (Die Referenz) in chapter 9. The config options relevant for the cyberJack installation are in "USB support". First make sure, that you use "UHCI (Intel PIIX4, VIA, ...) support" and NOT "UHCI Alternate Driver (JE) support". The last one is the uhci driver and will not work, as has been described in the previous paragraph.

For activating the cyberJack driver, you need to enable "Prompt for development and/or incomplete code/drivers" in "Code maturity level options". After this, you can go to "USB support" -> "USB Serial Converter support" and active "USB Serial Converter support". The cyberJack driver is then near the bottom.

Overview of the options:

```
Code maturity level options
  -> Prompt for development and/or incomplete code/drivers
    -> Set to 'y'.
USB support
  -> Support for USB
    -> Set to 'y' or 'm'
USB support
  -> UHCI (Intel PIIX4, VIA, ...) support
    -> Set to 'y' or 'm'
  OR
  -> OHCI (Compaq, iMacs, OPTi, SiS, ALi, ...) support
    -> Set to 'y' or 'm'
USB support
  -> USB Serial Converter support
    -> USB Serial Converter support
      -> Set to 'y' or 'm'
USB support
  -> USB Serial Converter support
    -> USB REINER SCT cyberJack pinpad/e-com USB chipcard reader
      -> Set to 'y' or 'm'
```

If your installation includes 'hotplug', all which is then neccessary for loading the module is to plug in the reader. When 'hotplug' is not installed, you can add the lines:

```
alias char-major-188 usbserial
above usbserial cyberjack
```

to the file `/etc/modules.conf`. This should then install the `usbserial` module and the `cyberjack` module when accessing the reader the first time. (Remember, that a driver for your USB controller must be installed, but not the `uhci.o` driver.)

2.1.4. LINUX Kernel 2.4.0 to 2.4.5

For some earlier kernels (2.4.0-2.4.5), there are patches in the kernel directory. Apply these with the program 'patch' by going to the kernel source directory and starting patch with **`% cd /usr/src/linux-2.4... % patch -p1 <.../cyberjack-2.4.x.patch`**

For configuration options see the above section about the kernels with version 2.4.6 and up.

(There is also the `kernel` directory in the cyberjack driver distribution, which tries to detect the kernel version (2.4.0-2.4.5) and to compile a module for it, but this is mostly untested.)

Anyway, It is strongly suggested to use a kernel with version 2.4.6 or above.

2.1.5. LINUX Kernel up to 2.3.x

Are not supportet by REINER SCT. Please upgrade the kernel or try yourself the backports.

2.1.6. OHCI host controllers

Currently there are problems with OHCI host controllers and the cyberJack under Linux. It seems to be a problem of the ohci driver. The symptoms are, that a CT_init fails after the reader has been accessed a few times. 2.4.20 did work with a OHCI PCI card (OPTi 82C861). So please try to upgrade at least to 2.4.20 if you have any problems with an OHCI controller.

2.1.7. Large number of readers

Your distribution has probably only a few `ttyUSBx` nodes in `/dev`. If you want to use more readers than this you have to create the needed nodes yourself. You can use for this the script **MAKEUSBDEV**. This creates 64 nodes (`/dev/ttyUSB0` - `/dev/ttyUSB63`). You can change the number by editing the script. This sets also the correct permission for mandatory locking. (See above.)

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 to many devices are attached. Known to work is 2.4.20.
- There occure sometimes timeouts 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 gread 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)

Note: Using **MAKEUSBDEV** should only be neccessary, when you want to attach large numbers of readers to a single PC. In all other cases there should be enough devices created by the Linux distribution or RPM package.

2.1.8. Hotplugging

Linux supports hotplugging with USB devices. This is implemented by calling **/sbin/hotplug** from the kernel.

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

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

2.2. User space part

The user space part of the cyberJack support is included in this cyberJack driver package. It is in the `ctapi/` directory and implements the CT-API defined for the German health insurance cards and is used by most German banking applications.

3. REINER SCT cyberJack pinpad_a USB (0x300)

Important: This section is only relevant for devices with ProductID 0x300!

The driver for this new generation of cyberJack readers is implemented 100% in userspace. This means no trouble with different kernel versions, compiling/patching the kernel, ...

All accesses are done via the `usb devfs` in `/proc/bus/usb`.

Permission handling is done *only* via the hotplug mechanism. The `cyberjack.usermap` directs the `usb.agent` to execute the **usbcyberjack** script. This script then dynamically updates the permissions of the respective device, so users in the group `cyberjack` are able to access it.

4. distribution-specific notes

4.1. RedHat 7.3

There is a RPM package available, which can be installed by running e.g. **`rpm -i <package file>`**

4.2. Redhat 9.0

4.3. Fedora Core 2 (i386, amd64)

4.4. SuSE 7.3

The version of the cyberJack driver, which comes with SuSE 7.3 has a few problems. These are fixed in the new RPM packages for this system. You can install this RPM by running e.g. **`rpm -i <package file>`** or update it by running e.g. **`rpm -Fvh <package file>`**

Additional you need to edit "/etc/rc.config.d/hotplug.rc.config". Here the variable "HOTPLUG_USB_HOSTCONTROLLER_LIST" must be set from "uhci usb-uhci usb-ohci ehci-hcd" to "usb-uhci uhci usb-ohci ehci-hcd" Without this change, the uhci host controller driver is loaded instead of the usb-uhci driver. For a description of this issue, see the section about kernels 2.4.6 and up.

4.5. SuSE 8.0

This distribution does load the correct UHCI driver. So if your hardware has a USB UHCI host controller, you needn’t make any changes.

You can install this RPM by running e.g. **`rpm -i <package file>`** or update it by running e.g. **`rpm -Fvh <package file>`**

4.6. SuSE 8.1

It seems to work here out of the box. You need to update the SuSE-provided driver with the REINER SCT driver if you need the new features present in the new drivers, or if you have one of the new (pinpad_a, 0x300) readers.

4.7. SuSE 8.2

4.8. SuSE 9.0

SuSE 9.x includes it’s own very particular permission and ressource management system called ’resmgr’.

In order to give a Linux user permissions to access the cyberjac pinpad_a (0x300) reader, you have to make sure the following line is present in /etc/pam.d/login:

session optional pam_resmgr.so grant=desktop

In order to give a Linux user permissions to access the cyberjack e-com/pinpad (0x100) reader, you have to put him in the ’uucp’ group.

For managing Linux permisions, please see the SuSE Linux User Manual or any other basic Linux System Administration manual.

4.9. SuSE 9.1 (i386, amd64)

Unfortunately SuSE 9.1 contains a broken kernel driver for the cyberJack e-com/pinpad (0x100). If you have such a reader, you will have to apply a kernel patch and recompile your kernel in order to make the device work. The respective patch can be found at ’patches/2.6.x/cyberjack-1.01-2.6.4.patch’

Please refer to section “Section 4.8” for further instructions.

4.10. SLES9 (i386, amd64)

Unfortunately, the SuSE SLES9 kernel contains a broken kernel driver for the cyberJack e-com/pinpad (0x100). If you have such a reader, you will have to apply a kernel patch and recompile your kernel in order to make the device work.

The respective patch can be found at ’patches/sles9-cyberjack.patch’.

Please refer to section “Section 4.8” for further instructions.

4.11. Debian GNU/Linux

A debian package for the userspace part of this driver is in development by Andreas Gredler <jimmy@g-tec.co.at>. Please wait for one of the upcoming releases of this driver.

4.12. All other Distributions

There is currently no experiance 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 **make** ,

The include file `ctapi.h` and the resulting libraries `libctapi-cyberjack.a` and `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 libraries `/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. If this is the only CT-API library installed, you should rename it to `libctapi.so`, because probably most applications look by default for this file name.

5. Additional Information

5.1. 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` envoronment variable before starting your application.

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

5.2. 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. ”

5.3. 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.

5.4. 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, i.e. pn=1 -> /dev/ttyUSB0. This behaviour is specified in the CT-API documentation.

5.5. 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.

5.5.1. 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/.

5.5.2. 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 pcscd 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.

5.6. 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.

5.7. command size

The command size is currently limited to ISO7816 short commands.

5.8. Testing

5.8.1. cjgeldkarte

cjgeldkarte

Name

`cjgeldkarte` — show account balance of a Geldkarte

Synopsis

cjgeldkarte [-l *`library`*] [-p *`port`*]

DESCRIPTION

cjgeldkarte prints on success the balance of the GeldKarte inserted into the cyberJack chipcard reader.

OPTIONS

`-l library`

This can be used to specify, which CT-API library to load. *library* can be only the file name, which is then searched in the LD_LIBRARY_PATH, or it can be the full path to a CT-API library. Default value is ’libctapi.so’.

`-p port`

This is the port number of the reader. Default value is 1.

NOTES

Note: cjgeldkarte does some low-level tests and is therefore not usable with any other CT-API library than the library for the cyberJack.

5.8.2. ctsh

There has also been added ’ctsh’. This is a CT-API shell, which means, you can send CT-API commands in a shell-like environment. Start it with the CT-API library name as the first and the port number as the second parameter. You get on the prompt the help with ’he’:

ej - eject card co cmd[:xxxx...] - send command from script (xx: hex data appended to command) se d:xxxxxxxx... - send APDU (d: dad, xx: data as hex, spaces are ignored) qu - quit he - help ’ej’ is a shortcut for Eject ICC. ’co’ send the command shortcut ’cmd’, which is searched for in $HOME/.ctshrc. .ctshrc contains one command shortcut per line. It begins with the shortcut name following the destination address and command data. An example .ctshrc is provided. Additional data can be appended to the shortcut name on the shell prompt after a colon. ’se’ is used to send a command to the reader or card. The destination address is specified before the colon, the data bytes in hexagesimal notation after the colon. ’qu’ terminates the program.

Depending on your configration ctsh has a build in command history.

5.8.3. Logging

The cyberjack CT-API library supports logging of the communication with the reader. This is done, when 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.

5.9. Keypressed callback

`IS8 **CT_keycb**(IU16*ctn*, void (*\**cb*) (void));`

The function `CT_keycb` 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.

5.10. Behind the scenes

This chapter describes what is happening (or should happen) behind the scenes. This information is for technical interested people and in case something goes wrong and has to be fixed.

As already seen in the README, the cyberJack driver consists of two parts, the kernel-space driver and the user-space driver. The kernel-space driver handles the USB transport layer, i.e. it reads data blocks via a device file (/dev/ttyUSB*: character device with major number 188 and minor numbers starting with 0) and transmits it to the reader with bulk OUT transfers. It also writes data blockes received from the reader to the device file. The readers signals here the availability of data via an interrupt IN channel and the data itself is fetched with bulk IN transfers.

In the kernel-space, the cyberjack module is based on some other drivers. The base is the usbcore module containing basic USB stuff. On top of this are the host controller drivers, i.e. usb-uhci and ohci. Because the cyberjack driver is part of the USB serial drivers, the usbserial module comes then and finally the cyberjack module. All these modules should be loaded upon plugin of the reader via kernel hotplug interface and distribution specific software. If that failes, modules can be loaded with modprobe (or insmod), e.g. with "modprobe usb-uhci" and then "modprobe cyberjack", but this is only temporary. Forcing module loading at boottime with e.g. Debian is done by adding the modules to be loaded to /etc/modules.

The user-space driver handles: - transport protocol very similar to T=1 - user-space API CT-API Upon a CT_init call it opens the specified device file (/dev/ttyUSB*) and locks it with mandatory locking (see 4.1 Mandatory locking). After some initializing it sends directly the CT-BCS APDUs provided with the CT_data function via T=1 to the reader, fetches the response and returns it to the application, i.e. all the intelligence (handling the smart card protocol, processing inputs from the PIN pad, ...) is in the reader itself.

6. 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/OpenBSD distribution you use including version, e.g. SuSE 8.2, 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 loaded modules, e.g. on Linux the output from the command "/sbin/lsmod".
- - List of attached USB devices, e.g. on Linux the content of the file "/proc/bus/usb/devices" or the output of the ’lsusb’ command.