TERIDIAN SEMICONDUCTOR CORP.

Simplifying System Integration™

73S1215F, 73S1217F CCID USB Linux Driver Installation Guide

April 27, 2009
Rev. 1.00
UG_12xxF_041

© 2009 Teridian Semiconductor Corporation. All rights reserved.
Teridian Semiconductor Corporation is a registered trademark of Teridian Semiconductor Corporation.
Simplifying System Integration is a trademark of Teridian Semiconductor Corporation.
Microsoft is a registered trademark of Microsoft Corporation.
Windows XP is a registered trademark of Microsoft Corporation.
Linux is a registered trademark of Linus Torvalds.
All other trademarks are the property of their respective owners.

Teridian Semiconductor Corporation makes no warranty for the use of its products, other than expressly contained in the Company's warranty detailed in the Teridian Semiconductor Corporation standard Terms and Conditions. The company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice and does not make any commitment to update the information contained herein. Accordingly, the reader is cautioned to verify that this document is current by comparing it to the latest version on http://www.teridian.com or by checking with your sales representative.

Teridian Semiconductor Corp., 6440 Oak Canyon, Suite 100, Irvine, CA 92618
TEL (714) 508-8800, FAX (714) 508-8877, http://www.teridian.com

Table of Contents

1 Introduction

This document provides the installation procedures for the Linux® user space CCID device driver in support of the Teridian 12xx Smart Card Chip. This document assumes that this installation is as userid `root`.

2 Components

This section lists the web sites and software components necessary to create an operational CCID environment.

2.1 Basic USB Requirements

2.1.1 Kernel

Some systems may have trouble enumerating some USB devices. If you encounter trouble, add support for USB OHCI controllers to your kernel in addition to EHCI and UHCI support. This should clear up any problems enumerating USB devices.

2.1.2 libusb

Your Linux platform should already have `libusb` installed. The CCID code requires at least `libusb-0.1.7`. This document is based on a system with `libusb-0.1.12` installed. If you wish to upgrade, the libusb website is http://libusb.sourceforge.net/.

If you do upgrade, be sure to update your system's shared library list via the `ldconfig` command as follows:

```
/sbin/ldconfig
```

The primary site for Linux CCID support is: http://pcsclite.alioth.debian.org/ccid.html

The software packages, mostly, install to `/usr/local/bin` by default. As a result, you might want to place the packages in `/usr/local/src`. To keep `/usr/local/src` tidy, we recommend you create a directory for the CCID packages. You may place such a directory, in any appropriate location on your system. We use `/usr/local/src` for our examples.

Create the CCID directory as follows:

```
mkdir  /usr/local/src/CCID
```

Change to the new directory:

```
cd /usr/local/src/CCID
```

Proceed to http://pcsclite.alioth.debian.org/ccid.html

Note: This document uses `ccid-1.2.1` and `pcsc-lite-1.4.0` in its examples. You should download the latest version available at the CCID website for all of the software packages mentioned below.

Follow the Source Download link: `ccid-1.2.1` to https://alioth.debian.org/frs/?group_id=30105&release_id=936

The following software packages are required for a complete driver:

```
pcsc-lite-1.4.0.tar.gz
ccid-1.2.1-tar.gz
libmusclecard-1.3.3.tar.gz
HandlerTest-0.3.1.tar.gz
```

Next, proceed to:

http://ludovic.rousseau.free.fr/softwares/

The following software packages are also required:

http://ludovic.rousseau.free.fr/softwares/pcsc-tools/index.html
`pcsc-tools-1.4.11.tar.gz`

http://ludovic.rousseau.free.fr/softwares/pcsc-perl/index.html
`pcsc-perl-1.4.6.tar.gz`

3 Installation

The sequence of installation is important as well as the running of the commands documented in this text, at the proper times as indicated.

The documentation that accompanies the software packages above, may assume package management tools for installation of the software. This document makes no such assumptions and as such provides the additional commands necessary to achieve a working driver environment on the first attempt, assuming no steps are missed.

3.1 Location

Since the above software packages, mostly, install to `/usr/local/bin`, by default. This document will now assume that you have installed the tar balls in `/usr/local/src`.

The downloaded tar balls should look as follows:

```
root@slack:/usr/local/src/CCID# ls -al
total 2076
drwxr-xr-x  2 root root   4096 2007-04-30 13:18 ./
drwxr-xr-x 20 root root   4096 2007-04-30 12:58 ../
-rw-r--r--  1 root root  18768 2007-04-13 10:45 HandlerTest-0.3.1.tar.gz
-rw-r--r--  1 root root 547610 2007-04-13 10:45 ccid-1.2.1.tar.gz
-rw-r--r--  1 root root 561772 2007-04-13 10:45 libmusclecard-1.3.3.tar.gz
-rw-r--r--  1 root root 862058 2007-04-13 10:45 pcsc-lite-1.4.0.tar.gz
-rw-r--r--  1 root root  41828 2007-04-30 13:17 pcsc-perl-1.4.6.tar.gz
-rw-r--r--  1 root root  55157 2007-04-30 13:18 pcsc-tools-1.4.8.tar.gz
root@slack:/usr/local/src/CCID#
```

3.2 Exploding the Tar Balls

The software is in the `.tar.gz` format and can be opened with the following command:

```
tar xzf  package_name.tar.gz
```

After all the packages are exploded, you should end up with the following directories in `/usr/local/src` or your chosen installation location:

```
pcsc-lite-1.4.0
ccid-1.2.1
libmusclecard-1.3.3
HandlerTest-0.3.1
pcsc-tools-1.4.8
pcsc-perl-1.4.6
```

The `/usr/local/src/CCID` directory should now look as follows:

```
root@slack:/usr/local/src/CCID# ls -al
total 2100
drwxr-xr-x  8 root root   4096 2007-04-30 13:33 ./
drwxr-xr-x 20 root root   4096 2007-04-30 12:58 ../
drwxr-xr-x  4 1000 1000   4096 2006-05-30 13:21 HandlerTest-0.3.1/
-rw-r--r--  1 root root  18768 2007-04-13 10:45 HandlerTest-0.3.1.tar.gz
drwxrwxrwx  7 1000 1000   4096 2007-01-27 08:48 ccid-1.2.1/
-rw-r--r--  1 root root 547610 2007-04-13 10:45 ccid-1.2.1.tar.gz
drwxrwxrwx  6 1000 1000   4096 2006-05-02 13:14 libmusclecard-1.3.3/
-rw-r--r--  1 root root 561772 2007-04-13 10:45 libmusclecard-1.3.3.tar.gz
drwxrwxrwx  7 1000 1000   4096 2007-02-13 12:26 pcsc-lite-1.4.0/
-rw-r--r--  1 root root 862058 2007-04-13 10:45 pcsc-lite-1.4.0.tar.gz
drwxr-xr-x  6 1000 1000   4096 2007-03-07 12:29 pcsc-perl-1.4.6/
-rw-r--r--  1 root root  41828 2007-04-30 13:17 pcsc-perl-1.4.6.tar.gz
drwxr-xr-x  2 1000 1000   4096 2006-11-26 06:49 pcsc-tools-1.4.8/
-rw-r--r--  1 root root  55157 2007-04-30 13:18 pcsc-tools-1.4.8.tar.gz
root@slack:/usr/local/src/CCID#
```

You may wish to move the tar balls to a separate directory as follows:

```
mkdir tar-balls
mv *.tar.gz tar-balls/
```

That command sequence and the resulting CCID directory are:

```
root@slack:/usr/local/src/CCID# mkdir tar-balls
root@slack:/usr/local/src/CCID# mv *.tar.gz tar-balls/
root@slack:/usr/local/src/CCID# ls -al
total 36
drwxr-xr-x  9 root root 4096 2007-04-30 13:39 ./
drwxr-xr-x 20 root root 4096 2007-04-30 12:58 ../
drwxr-xr-x  4 1000 1000 4096 2006-05-30 13:21 HandlerTest-0.3.1/
drwxrwxrwx  7 1000 1000 4096 2007-01-27 08:48 ccid-1.2.1/
drwxrwxrwx  6 1000 1000 4096 2006-05-02 13:14 libmusclecard-1.3.3/
drwxrwxrwx  7 1000 1000 4096 2007-02-13 12:26 pcsc-lite-1.4.0/
drwxr-xr-x  6 1000 1000 4096 2007-03-07 12:29 pcsc-perl-1.4.6/
drwxr-xr-x  2 1000 1000 4096 2006-11-26 06:49 pcsc-tools-1.4.8/
drwxr-xr-x  2 root root 4096 2007-04-30 13:39 tar-balls/
root@slack:/usr/local/src/CCID#
```

3.3 Required Package Build Order

The following order of package builds must be adhered to for proper operation of the resultant user space device driver.

1. pcsc-lite-1.4.0
2. ccid-1.2.1
3. pcsc-perl-1.4.6
4. pcsc-tools-1.4.8

Installation will eventually result in creation of:

```
/usr/local/pcsc/drivers
```

The following packages are optional:

```
libmusclecard-1.3.3
HandlerTest-0.3.1
```

3.4 Building the Packages

The building order, additional post-build installation commands and startup order, are critical to proper operation. Please maintain the order shown below and enter the commands in the sequence they are presented.

3.4.1 pcsc-lite

Enter the following commands:

```
cd pcsc-lite-1.4.0
/configure
make
make install
```

This installs /usr/local/sbin/pcscd, the CCID daemon.

3.4.2 ccid-1.2.1

Run the following commands:

```
cd  /usr/local/src/CCID/ccid-1.2.1
./configure
make
```

Now you must edit a file in the ./readers directory, to define the Teridian device:

```
ls -al readers/supported_readers.txt
```

The output should look as follows:

```
root@slack:/usr/local/src/CCID/ccid-1.2.1# ls -al
readers/supported_readers.txt
-rw-r--r-- 1 1000 1000 2527 2007-01-04 12:26 readers/supported_readers.txt
root@slack:/usr/local/src/CCID/ccid-1.2.1#
```

Edit this file and add the following text at the end (about line 107):

```
#Teridian
0x1862:0x0000:TSC12xxF
```

Save the changes.

Now install the ccid-1.2.1 build output, as follows:

```
make install
```

This step will install the following file, containing your change:

```
/usr/local/pcsc/drivers/ifd-ccid.bundle/Contents/Info.plist
```

Verify your change as follows:

```
grep -n TSC12xxF /usr/local/pcsc/drivers/ifd-ccid.bundle/Contents/Info.plist
```

```
root@slack:/usr/local/src/CCID/ccid-1.2.1# grep -n TSC12xxF /usr/local/pcsc/
drivers/ifd-ccid.bundle/Contents/Info.plist
304:            <string>TSC12xxF</string>
root@slack:/usr/local/src/CCID/ccid-1.2.1#
```

This also installs:

```
 /usr/local/pcsc/drivers/ifd-ccid.bundle/Contents/Linux/libccid.so.1.2.1
```

Listing the contents of this new directory, gives the following:

```
root@slack:/usr/local/src/CCID/pcsc-lite-1.4.0# ls -al
/usr/local/pcsc/drivers/  ifd-ccid.bundle/Contents/Linux/
total 220
drwxr-xr-x 2 root root   4096 2007-04-13 13:26 ./
drwxr-xr-x 3 root root   4096 2007-04-13 13:26 ../
-rwxr-xr-x 1 root root 211134 2007-04-13 13:26 libccid.so.1.2.1*
root@slack:/usr/local/src/CCID/pcsc-lite-1.4.0#
```

A new shared library has been created and installed in a non-standard location. **It must be added to your system's shared library list before the other packages can be correctly built.**

First, identify a location that the new library can be found from by listing the sequence of system libraries that are to be searched for shared libraries:

```
cat /etc/ld.so.conf
```

```
root@slack:/usr/local/src/CCID/pcsc-lite-1.4.0# cat /etc/ld.so.conf
/usr/local/lib
/usr/X11R6/lib
/usr/i486-slackware-linux/lib
/opt/kde/lib
/usr/lib/qt/lib
root@slack:/usr/local/src/CCID/pcsc-lite-1.4.0#
```

Notice that `/usr/local/lib` is present in the search list. Either place a link in `/usr/local/lib` to the new shared library `libccid.so.1.2.1` or copy it into `/usr/local/lib`.

For this installation, copy it to `/usr/local/lib`.

To copy the new shared library, enter the following:

```
cd /usr/local/pcsc/drivers/ifd-ccid.bundle/
cp -v Contents/Linux/libccid.so.1.2.1/usr/local/lib/ /sbin/ldconfig -v |grep ccid
```

This sequence of commands installs the new shared library in a system recognized location and add it to the system search list.

```
root@slack:/usr/local/src/CCID/pcsc-lite-1.4.0# cd
/usr/local/pcsc/drivers/ifd-ccid.bundle/
root@slack:/usr/local/pcsc/drivers/ifd-ccid.bundle# cp -v Contents/
Linux/libccid.so.1.2.1 /usr/local/lib/
`Contents/Linux/libccid.so.1.2.1' -> `/usr/local/lib/libccid.so.1.2.1'
root@slack:/usr/local/pcsc/drivers/ifd-ccid.bundle# /sbin/ldconfig -v |grep
ccid
        libccid.so.0 -> libccid.so.1.2.1 (changed)
root@slack:/usr/local/pcsc/drivers/ifd-ccid.bundle#
```

3.4.3 pcsc-perl-1.4.6

This section uses Perl to drive the installation of the package. Change to the `pcsc-perl` directory:

```
cd  /usr/local/src/CCID/pcsc-perl-1.4.6
```

The `README.Unix` file contains the following:

```
root@slack:/usr/local/src/CCID/pcsc-perl-1.4.6# cat README.Unix
```

To build the wrapper for Unix:

```
$ perl Makefile.PL
$ make
```

Insert a smart card in the (first) PCSC reader.

```
$ make test
$ sudo make install
```

```
$Id: README.Unix,v 1.6 2006-08-12 17:35:49 rousseau Exp $
root@slack:/usr/local/src/CCID/pcsc-perl-1.4.6#
```

Install the package using Perl according to the above README:

```
perl Makefile.pl
make
```

This step has created another shared library: **PCSC.so**

At this point, **make test**, per the `README.Unix` file, will fail since the daemon is running yet.

Do not install a smart card. Enter:

```
make test
make install
```

3.4.4 pcsc-tools-1.4.8

Enter the following command sequence:

```
cd /usr/local/src/CCID/
make
make install
```

This will build and install **pcsc_scan** in **/usr/local/bin/.**

3.4.5 CCID Driver Operation

At this point, all four required packages have been built and installed. It is now time to activate the newly installed driver and test the smart card device. Enter the following:

```
/usr/local/sbin/pcscd
```

This produces the following system console message sequence if you have set `klogd` to message level 7:

```
Apr 30 15:45:05 slack pcscd: pcscdaemon.c:542:main() pcsc-lite 1.4.0 daemon ready.
Apr 30 15:45:05 slack pcscd: hotplug_libusb.c:395:HPEstablishUSBNotifications()
Driver ifd-ccid.bundle does not support IFD_GENERATE_HOTPLUG
Apr 30 15:45:05 slack pcscd: hotplug_libusb.c:404:HPEstablishUSBNotifications()
Polling forced every 1 second(s)
```

Now plug in your USB card reader.

The console, if you have set `klogd` to message level 7, should show the following list of messages:

```
Apr 30 15:47:01 slack kernel: [492775.626570] hub 1-0:1.0: state 7 ports 8
chg 0000 evt 0010
Apr 30 15:47:01 slack kernel: [492775.626652] ehci_hcd 0000:00:02.2:
GetStatus port 4 status 001803 POWER sig=j CSC CONNECT
Apr 30 15:47:01 slack kernel: [492775.626730] hub 1-0:1.0: port 4, status
0501, change 0001, 480 Mb/s
Apr 30 15:47:01 slack kernel: [492775.730346] hub 1-0:1.0: debounce: port 4:
total 100ms stable 100ms status 0x501
Apr 30 15:47:01 slack kernel: [492775.781252] ehci_hcd 0000:00:02.2: port 4
full speed --> companion
Apr 30 15:47:01 slack kernel: [492775.781329] ehci_hcd 0000:00:02.2:
GetStatus port 4 status 003001 POWER OWNER sig=se0 CONNECT
Apr 30 15:47:01 slack kernel: [492775.781442] hub 3-0:1.0: state 7 ports 4
chg 0000 evt 0004
Apr 30 15:47:01 slack kernel: [492775.781491] ohci_hcd 0000:00:02.1:
GetStatus roothub.portstatus [1] = 0x00010101 CSC PPS CCS
Apr 30 15:47:01 slack kernel: [492775.781562] hub 3-0:1.0: port 2, status
0101, change 0001, 12 Mb/s
Apr 30 15:47:01 slack kernel: [492775.885034] hub 3-0:1.0: debounce: port 2:
total 100ms stable 100ms status 0x101
Apr 30 15:47:01 slack kernel: [492775.946915] ohci_hcd 0000:00:02.1:
GetStatus roothub.portstatus [1] = 0x00100103 PRSC PPS PES CCS
Apr 30 15:47:01 slack kernel: [492775.997810] usb 3-2: new full speed USB
device using ohci_hcd and address 3
Apr 30 15:47:01 slack kernel: [492776.061676] ohci_hcd 0000:00:02.1:
GetStatus roothub.portstatus [1] = 0x00100103 PRSC PPS PES CCS
Apr 30 15:47:02 slack kernel: [492776.126547] usb 3-2: ep0 maxpacket = 16
Apr 30 15:47:02 slack kernel: [492776.148377] usb 3-2: skipped 1 descriptor
after interface
```

```
Apr 30 15:47:02 slack kernel: [492776.151366] usb 3-2: default language
0x0409
Apr 30 15:47:02 slack kernel: [492776.164344] usb 3-2: new device strings:
Mfr=1, Product=2, SerialNumber=3
Apr 30 15:47:02 slack kernel: [492776.164415] usb 3-2: Product: TSC12xxF
Apr 30 15:47:02 slack kernel: [492776.164447] usb 3-2: Manufacturer: Teridian
Semiconductors
Apr 30 15:47:02 slack kernel: [492776.164493] usb 3-2: SerialNumber:
123456788
Apr 30 15:47:02 slack kernel: [492776.164563] usb 3-2: uevent
Apr 30 15:47:02 slack kernel: [492776.164611] usb 3-2: usb_probe_device
Apr 30 15:47:02 slack kernel: [492776.164681] usb 3-2: configuration #1
chosen from 1 choice
Apr 30 15:47:02 slack kernel: [492776.167351] usb 3-2: adding 3-2:1.0 (config
#1, interface 0)
Apr 30 15:47:02 slack kernel: [492776.167442] usb 3-2:1.0: uevent
Apr 30 15:47:02 slack kernel: [492776.167601] drivers/usb/core/inode.c:
creating file '003'
Apr 30 15:47:02 slack kernel: [492776.167662] hub 3-0:1.0: state 7 ports 4
chg 0000 evt 0004
Apr 30 15:47:03 slack pcscd: hotplug_libusb.c:448:HPAddHotPluggable() Adding
USB device: 003:003
Apr 30 15:47:03 slack pcscd: readerfactory.c:1108:RFInitializeReader()
Attempting startup of TSC12xxF (123456788) 00 00 using
/usr/local/pcsc/drivers/ifd-ccid.bundle/Contents/Linux/libccid.so.1.2.1
Apr 30 15:47:03 slack pcscd: readerfactory.c:977:RFBindFunctions() Loading
IFD Handler 3.0
Apr 30 15:47:03 slack pcscd: ifdhandler.c:1231:init_driver() LogLevel: 0x0003
Apr 30 15:47:03 slack pcscd: ifdhandler.c:1241:init_driver() DriverOptions:
0x0000
Apr 30 15:47:03 slack pcscd: ifdhandler.c:77:IFDHCreateChannelByName() lun:
0, device: usb:1862/0000:libusb:003:003
Apr 30 15:47:03 slack pcscd: ccid_usb.c:229:OpenUSBByName() Manufacturer:
Ludovic Rousseau (ludovic.rousseau@free.fr)
Apr 30 15:47:03 slack pcscd: ccid_usb.c:239:OpenUSBByName() ProductString:
Generic CCID driver v1.2.1
Apr 30 15:47:03 slack pcscd: ccid_usb.c:245:OpenUSBByName() Copyright: This
driver is protected by terms of the GNU Lesser General Public License version
2.1, or (at your option) any later version.
Apr 30 15:47:03 slack pcscd: ccid_usb.c:393:OpenUSBByName() Found
Vendor/Product: 1862/0000 (TSC12xxF)
Apr 30 15:47:03 slack pcscd: ccid_usb.c:395:OpenUSBByName() Using USB
bus/device: 003/003
Apr 30 15:47:03 slack pcscd: ccid_usb.c:779:get_data_rates() declared: 9600
bps
Apr 30 15:47:03 slack pcscd: ccid_usb.c:779:get_data_rates() declared: 14400
bps
Apr 30 15:47:03 slack pcscd: ccid_usb.c:779:get_data_rates() declared: 19200
bps
Apr 30 15:47:03 slack pcscd: ccid_usb.c:779:get_data_rates() declared: 28800
bps
Apr 30 15:47:03 slack pcscd: ccid_usb.c:779:get_data_rates() declared: 38400
bps
Apr 30 15:47:03 slack pcscd: ccid_usb.c:779:get_data_rates() declared: 57600
bps
Apr 30 15:47:03 slack pcscd: ccid_usb.c:779:get_data_rates() declared: 115200
bps
```

```
Apr 30 15:47:03 slack pcscd: ifdhandler.c:271:IFDHGetCapabilities() lun: 0,
tag: 0xFAE
Apr 30 15:47:03 slack pcscd: ifdhandler.c:313:IFDHGetCapabilities() Reader
supports 2 slots
Apr 30 15:47:03 slack pcscd: ifdhandler.c:271:IFDHGetCapabilities() lun: 0,
tag: 0xFAC
Apr 30 15:47:03 slack pcscd: readerfactory.c:1108:RFInitializeReader()
Attempting startup of TSC12xxF (123456788) 00 01 using
/usr/local/pcsc/drivers/ifd-ccid.bundle/Contents/Linux/libccid.so.1.2.1
Apr 30 15:47:03 slack pcscd: readerfactory.c:822:RFLoadReader() Warning
library pointer not NULL
Apr 30 15:47:03 slack pcscd: readerfactory.c:977:RFBindFunctions() Loading
IFD Handler 3.0
Apr 30 15:47:03 slack pcscd: ifdhandler.c:77:IFDHCreateChannelByName() lun:
1, device: usb:1862/0000:libusb:003:003
Apr 30 15:47:03 slack pcscd: ccid_usb.c:229:OpenUSBByName() Manufacturer:
Ludovic Rousseau (ludovic.rousseau@free.fr)
Apr 30 15:47:03 slack pcscd: ccid_usb.c:239:OpenUSBByName() ProductString:
Generic CCID driver v1.2.1
Apr 30 15:47:03 slack pcscd: ccid_usb.c:245:OpenUSBByName() Copyright: This
driver is protected by terms of the GNU Lesser General Public License version
2.1, or (at your option) any later version.
Apr 30 15:47:03 slack pcscd: ccid_usb.c:332:OpenUSBByName() Opening slot: 1
```

Note that in the above log, `ccid_usb.c` recognizes the smart card.

Run the `pcsc_scan` utility to extract the details of the card:

```
/usr/local/bin/pcsc_scan
```

This will produce the following output in your `xterm/kconsole` window:

```
root@slack:/usr/local/src/CCID/pcsc-tools-1.4.8# /usr/local/bin/pcsc_scan
PC/SC device scanner
V 1.4.8 (c) 2001-2006, Ludovic Rousseau <ludovic.rousseau@free.fr>
Compiled with PC/SC lite version: 1.4.0
Scanning present readers
0: TSC12xxF (123456788) 00 00
1: TSC12xxF (123456788) 00 01

Mon Apr 30 15:53:06 2007
 Reader 0: TSC12xxF (123456788) 00 00
  Card state: Card removed,

Mon Apr 30 15:53:06 2007
 Reader 1: TSC12xxF (123456788) 00 01
  Card state: Card removed,
```

This shows that `pcsc_scan` has found the Teridian reader and that it does not have a smart card in it.

Now insert a smart card...

Insertion of a test smart card gives the following additional output:

```
Mon Apr 30 15:56:03 2007
 Reader 0: TSC12xxF (123456788) 00 00
  Card state: Card inserted,
```

```
  ATR: 3B EF 00 FF 81 31 86 45 49 42 4D 20 4D 46 43 34 30 30 30 30 38 33 31
43


ATR: 3B EF 00 FF 81 31 86 45 49 42 4D 20 4D 46 43 34 30 30 30 30 38 33 31 43
+ TS = 3B --> Direct Convention
+ T0 = EF, Y(1): 1110, K: 15 (historical bytes)
  TB(1) = 00 --> VPP is not electrically connected
  TC(1) = FF --> Extra guard time: 255 (special value)
  TD(1) = 81 --> Y(i+1) = 1000, Protocol T = 1
-----
  TD(2) = 31 --> Y(i+1) = 0011, Protocol T = 1
-----
  TA(3) = 86 --> IFSC: 134
  TB(3) = 45 --> Block Waiting Integer: 4 - Character Waiting Integer: 5
+ Historical bytes: 49 42 4D 20 4D 46 43 34 30 30 30 30 38 33 31
  Category indicator byte: 49 (proprietary format)
+ TCK = 43 (correct checksum)
```

Refer to the `pcsc_scan man` page for a description of the command and its option set.

4 Final Notes

Some of the packages contain `man` pages. They may or may not be correctly installed for you system.

Refer to the package directories for details.

If you have problems with finding a `man` page refer to the following example:

```
root@slack:/usr/local/src/CCID# find /usr/local/ -name pcsc_scan.1.gz
/usr/local/src/CCID/pcsc-tools-1.4.8/pcsc_scan.1.gz
/usr/local/share/man/man1/pcsc_scan.1.gz
root@slack:/usr/local/src/CCID#
```

The example has the `pcsc_scan man` page, `pcsc_scan.1.gz`, in `/usr/local/share/man/man1/`.

The system we installed on, Slackware 11.0, does not use this `man` path.

The problem can be resolved as follows:

```
cp /usr/local/share/man/man1/pcsc_scan.1.gz /usr/local/man/man1/
```

Testing the result gives:

```
root@slack:/usr/local/src/CCID# man pcsc_scan
PCSC_SCAN(1)

NAME
       pcsc_scan - regularly scans every PC/SC readers connected to the host

SYNOPSIS
       pcsc_scan [options]

DESCRIPTION
       This manual page documents briefly the pcsc_scan command.

       pcsc_scan is a program that regularly scans every PC/SC readers
connected to the host
```

```
       When  pcsc_scan is started it asks pcscd the list of available smart
card readers. The list is printed. A sequence number is printed
       before each reader.

       Example:
        PC/SC device scanner
        V 1.1.0 (c) 2001-2002, Ludovic Rousseau
        PC/SC lite version: 1.1.1
        0: GemPC410 0 0
        1: GemPC430 0 0

       When a card is inserted in any reader some information is printed:

       date and time:
              Thu Jun 13 18:56:14 2002

       reader name:
              Reader 0 (GemPC410 0 0)

       card state and occured event:
              Card state: State has changed, Card inserted,

       ATR in case of card insertion:
              ATR: 3B 82 00 86 1E

       print an ATR analysis if the ATR_analysis command is available:
               ATR: 3B 82 00 86 1E
               + TS = 3B --> Direct Convention
               + T0 = 82, Y(1): 1000, K: 2 (historical bytes)
                 TD(1) = 00 --> Y(i+1) = 0000, Protocol T = 0
               -----
               + Historical bytes: 86 1E

OPTIONS
       -h     print help

       -V     print version number

       -n     do not print ATR analysis

SEE ALSO
lines 1-55
```

5 Related Documentation

The following 73S12xxF documents are available from Teridian Semiconductor Corporation:

71S1215F Data Sheet
71S1217F Data Sheet
73S12xxF Smart Card Terminal Controller Family Software User's Guide
73S12xxF Evaluation Board User's Guide
Teridian Flash Programming Tool
73S1215F, 73S1217F Boot Loader – DFU Class Firmware Application Note
73S1215F, 73S1217F Windows XP 32 USB CCID and DFU Drivers Installation Guide
73S1215, 73S1217F CCID Application Note

6 Contact Information

For more information about Teridian Semiconductor products or to check the availability of the 73S12xxF, contact us at:

6440 Oak Canyon Road
Suite 100
Irvine, CA 92618-5201

Telephone: (714) 508-8800
FAX: (714) 508-8878
Email: scr.support@teridian.com

For a complete list of worldwide sales offices, go to http://www.teridian.com.

Revision History

Revision	Date	Description
1.00	4/27/2009	First publication.