Installation of Smart Card Software

Damien SAUVERON*
damien.sauveron@labri.fr
LaBRI, Laboratoire Bordelais de Recherche en Informatique
UMR 5800 – Université Bordeaux 1
351 cours de la Libération, 33405 Talence CEDEX, FRANCE.

1st September 2003

Abstract

The installation and the usage of some Smart Card software is a complex job. This document could help you to set up an environment in order to take a good way for the Smart Card's world.

KEYWORDS: *PC/SC, OCF, Smart card, Java Card.*

1 Introduction

I have written this document in order to share my experience on usage of Smart Card on a Linux environment (Slackware 8.1). I hope this will help you to cope with the installation and using problems.

This document describes the intallation of:

- PC/SC Lite and some drivers;
- many Java Development Kits and Java Card Development Kits;
- OpenCard Framework (OCF), OCF to PC/SC bridge, GemXpresso RAD3;
- JPCSC library and JCOP Tools.

Warning: For many operations you must have the root privileges on the host.

2 Greetings

Thanks for all the volunters who work on the development of PC/SC Lite, free drivers, Muscle, etc. I don't forget all the people who answer on Muscle and other mailing list, newgroups, etc.

*PhD at the LaBRI and engineer in the ITSEF center of SERMA Technologies.

3 PC/SC

3.1 Install of PC/SC Lite

Download `pcsc-lite-1.2.0-rc1.tar.gz` [3]
```
tar xvzf pcsc-lite-1.2.0-rc1.tar.gz
cd pcsc-lite-1.2.0-rc1
```

If you want use USB readers you must choose the way for the hotplug:

- It is recommended to used them through libusb which is a portable API available on many OS. If you want used the CCID Driver for your USB CCID reader you must used it. If libusb is not installed, install it (*cf.* Section 4).
  ```
  ./configure --enable-libusb=/usr/local --enable-daemon --enable-debug
  --enable-threadsafe --prefix=/usr/local/pcsc --sysconfdir=/etc
  --enable-runpid=/var/run/pcscd.pid
  ```

  Warning: `/usr/local` is my path for my libusb install.

- Else you can also used the deprecated `--enable-usb` that allows you to use directly the USB through your OS but it is less interoperable.
  ```
  ./configure --enable-usb --enable-daemon --enable-debug
  --enable-threadsafe --prefix=/usr/local/pcsc --sysconfdir=/etc
  --enable-runpid=/var/run/pcscd.pid
  ```

Warning: The threadsafe option implies to compile the clients of the PC/SC middleware with `-lpthread` (library pthread).

```
make
```
Get the root privileges.
```
make install
echo ‘‘/usr/local/pcsc/lib’’ >> /etc/ld.so.conf
ldconfig
mkdir /dev/pcsc
mkdir /usr/local/pcsc/drivers
echo ‘‘/usr/local/pcsc/sbin/pcscd’’ >> /etc/rc.d/rc.local
```

Warning: For security reasons we can create a special account with the good privileges to start the daemon.

In order to testing PC/SC there is in the directory `src/` the tool `testpcsc`
Go in the directory `src/` and if you have followed the previous steps just do `make testpcsc` else if you wish compile only this tool, delete in `testpcsc.c` the line `#include "config.h"` and compile it with:
```
gcc -o testpcsc testpcsc.c -L/usr/local/pcsc/lib -lpcsclite -I. -lpthread
```

In my opinion: A problem is that the PC/SC Lite includes the MuscleCard Framework API. A good idea shall be to move the MuscleCard Framework API from this archive to the muscleframework package. For a beginner, this decreases the complexity.

But the MuscleCard Framework API is now here and you can test it with the tool `muscletest` in the directory `src/`
Compile the source with:
`gcc -o muscletest muscletest.c -I. -L/usr/local/pcsc/lib/ -lpcsclite -lpthread`

3.2 Install of SERIAL readers

Your kernel or one of its module must handle the serial port.

3.2.1 Gemplus GCR410

Get the root privileges.
`cd /dev/pcsc`
`ln -s ../ttyS0 1` (if you connect the reader on the COM1)
`cd /usr/local/pcsc/drivers`

Download `ifd-gempc-0.8.0.tar.gz` [2]
`tar xvzf ifd-gempc-0.8.0.tar.gz`
`cd /usr/local/pcsc/drivers/ifd-gempc-0.8.0/GemPC410`
`make`
Add in `/etc/reader.conf` the following entry:

```
FRIENDLYNAME      "Gemplus GemPC410 Reader"
DEVICENAME        GEMCORE
LIBPATH           /usr/local/pcsc/drivers/ifd-gempc-0.8.0/GemPC410/libGemPC410.so
CHANNELID         1
```

3.2.2 SmartMouse SM1 RS232

Get the root privileges.
`cd /dev/pcsc`
`ln -s ../ttyS0 1` (if you connect the reader on the COM1)
`cd /usr/local/pcsc/drivers`

Download `ifd-sm-x.y.z.tar.gz` (where `x.y.z` is the revision) [**?**]
`cd /usr/local/pcsc/drivers/`
`tar xvzf ifd-sm-x.y.z.tgz`
`cd ifd-sm-x.y.z`
`make`
Add in `/etc/reader.conf` the following entry:

```
FRIENDLYNAME      "SM1 Reader"
DEVICENAME        SM1
LIBPATH           /usr/local/pcsc/drivers/ifd-sm-x.y.z/libsm_ifd.so
CHANNELID         1
```

3.3 Install of USB readers

Your kernel or one of its module must handle the USB.
For me:
```
modprobe usbcore
modprobe usb-uhci irq 11
mount -t usbdevfs none /proc/bus/usb
```

3.3.1 Generic CCID reader

Download `ccid-x.y.z.tar.gz` [3]
```
tar xvzf ccid-x.y.z.tar.gz
cd ccid-x.y.z.tar.gz
```
This driver needs libusb (*cf.* Section 4).
I have installed PC/SC Lite in the `/usr/local/pcsc/` and I must modify line 70 the `/usr/local/include` by `/usr/local/pcsc/include` the file `src/check`.
```
make
```
Get the root privileges.
```
make install
```

3.3.2 SCM Microsystems SCR 331

1) Plug your reader and see the information with `lsusb -v` or `cat /proc/bus/usb/devices`
If you get for the SCR331 an `idProduct` value set to `0xe000`, you must upgrade your firmware.
Go to the next step.
If you get `0xe001` skip the next step.

2) The firmware in the USB reader can be updated to the latest version, which should work with the GNU/Linux driver on SCM's web site:

- Find a windows machine (an unfortunate necessity)
- Download and install the `SCRx31_USB_1.40_signed.zip` windows driver from `http://www.scmmicro.com/security/secureCard-downloads.html` (Click on the SCR331 PCSC link.) Personally I use `SCR331 Installer V2.05.zip`.
- Download the `USB-FWUpdate.zip` program.
- Download the new firmware (`SCRx31_Firmware_4.13.zip`).
- Run the `USB-FWUpdate` program and select the `.bin` file from the firmware package.
- Now the USB reader should be upgraded to the CCID version [7].
- Throw your windows system away and plug the reader into your GNU/Linux box.

Now `lsusb -v` or `cat /proc/bus/usb/devices` shall give an `idProduct` set up at `0xe001`.

3) Install the GNU/Linux driver from

`ftp://ftp.scmmicro.com/security/drivers/scr331ccidDriver-0.9.0-1.i386.rpm`
On my Slackware, I converted the `rpm` to a `tgz` file using `rpm2tgz`
`tar xvzf scr331ccidDriver-0.9.0-1.i386.tgz`
Get the root privileges and move the driver to `/usr/local/pcsc/drivers/`

3.3.3 OMNIKEY CardMan 2020

1) `tar xzvf cm2020_installer_vX_Y_Z_src.tar.gz`
`cd cm2020_installer_vX_Y_Z_src`
Get the root privileges.
`sh install`

3.4 Install of PCMCIA readers

3.4.1 Gemplus GPR400

1) Installation of the module for the pcmcia card GRP400:
This is necessary to handle the PCMCIA. Download the sources of the pcmcia-cs [9]
`tar xvzf pcmcia-cs-3.1.34.tar.gz`
Download the module for the GPR400 [1]
Following the `README` of `gpr400_cs-0.9.6.tar.gz` i.e:
`cd pcmcia-cs-3.1.34`
`tar xvzf gpr400_cs-0.9.6.tar.gz`
Change the value of `#define PCMCIA_DEBUG` from `1` to `0` in the `gpr400_cs.c`
`make config`
`make all`
Get the root privileges.
`make install`
`mknod /dev/gpr400 c 123 0`
`chmod 0666 /dev/gpr400`

2) Installation of the GPR400 PC/SC driver:
Download the source `ifd-gpr400-0.3.tar.gz` [1]
Get the root privileges.
`cd /dev/pcsc`
`ln -s ../gpr400 5`
`cd /usr/local/pcsc/drivers`
`tar xvzf ifd-gpr400-0.3.tar.gz`
`cd ifd-gpr400-0.3`
Adding at the beginning of the `CFLAGS -I/usr/local/pcsc/include` in the `Makefile`
Comment in the `pcscdefines.h` file the following lines:

```
/*
  typedef unsigned long      DWORD;
  typedef unsigned long*     PDWORD;
  typedef unsigned char      UCHAR;
  typedef unsigned char*     PUCHAR;
  typedef char*              LPSTR;
  typedef long               RESPONSECODE;
*/
```

Copy `ifdhandler.h` from `ifd-devkit-1.0.0.tar.gz` or from an old version of pcsc-lite (for instance 1.0.1) in the directory.
`make`
Adding in `/etc/reader.conf` the following entry:

```
FRIENDLYNAME      "Gemplus GPR400 Reader"
DEVICENAME        GPR400
LIBPATH           /usr/local/pcsc/drivers/ifd-gpr400-0.3/libgpr400_ifd.so
CHANNELID         5
```

Warning: I have submitted a modify version of this driver to Joe Phillips.

4 Install of libusb

Download `libusb-0.1.7.tar.gz` [8]
`tar xvzf libusb-0.1.7.tar.gz`
`cd libusb-0.1.7 ./configure`
`make`
Get the root privileges.
`make install`

5 Installation of the JDKs

Download the JDKs.

5.1 SUN JDK 1.2.2

Get the root privileges. `cd /usr/local`
`tar xvzf jdk-1_2_2_011-linux-i386.tar.gz`

5.2 SUN JDK 1.3.1

Get the root privileges. `cd /usr/local`
`tar xvzf j2sdk-1_3_1_04-linux-i586.bin`

5.3 IBM JDK 1.3.1

Get the root privileges. `cd /usr/local`
`tar xvzf IBMJava2-SDK-131-linux.tgz`

Do some scripts to set up the environment

5.4 Installation of fonts

For the SUN JDK 1.2.2 and the SUN JDK 1.3.1
Download `symbol.ttf`
Get the root privileges.

`cp symbol.ttf $JAVA_HOME/jre/lib/fonts/`
Add to the file `$JAVA_HOME/jre/lib/fonts/fonts.dir` the following line
`symbol.ttf -urw-symbol-medium-r-normal--0-0-0-0-p-0-adobe-fontspecific`
and modify the number of fonts at the beginning of the file (adding 1).

5.5 Installation of the communication APIs

Get on the Gemplus CD of GemXPresso RADIII the `linux_commapi_conf.tar`
`tar xvf linux_commapi_conf.tar`
Get the root privileges.
`cp comm.jar $JAVA_HOME/jre/lib/ext/`
`cp javax.comm.properties $JAVA_HOME/jre/lib/`
`cp jcl.jar $JAVA_HOME/jre/lib/ext/`

- For the SUN JDK
  `cp libSerial.so $JAVA_HOME/jre/lib/i386/`
  `cp libParallel.so $JAVA_HOME/jre/lib/i386/`

- For the IBM JDK
  `cp libSerial.so $JAVA_HOME/jre/lib/ext/`
  `cp libParallel.so $JAVA_HOME/jre/lib/ext/`

For using the serial port the user must have the rights `rw` on it.
As root do `chmod 666 /dev/ttyS?`

6 Installation of the Java Card Development Kits

Download the JCDKs [10].
Get the root privileges. `cd /usr/local`
`mkdir javacard`
`cd javacard`

6.1 JCDK 2.1.1

`tar xvzf java_card_kit-2_1_1-unix[1].tar.Z`
`mv jc211 java_card_kit-2_1_1`

6.2 JCDK 2.1.2

`unzip java_card_kit-2_1_2-solsparc.zip`

6.3 JCDK 2.2

`unzip java_card_kit-2_2-solsparc-gl.zip`
Do some scripts to set up the environment.

7 Installation of the GemXpresso RAD III kit

The install does not work with the JDK 1.2.2
Get the root privileges
mount the `'cdrom'` with the GemXPresso RAD III CD.
`cd 'cdrom'/Unix`
I have chosen to install the different components in `/opt/gemxpresso.rad3/`
To install the RAD III:
`./RAD3_INS.bin`
To install the card profiles:
`./CP_V1.bin`
`./CP_V2.bin`
`./CP_PK.bin`
To install the crypto:
`./FPK_SP.bin`
To install the OpenCard Framework:
`./OCF_INS.bin`

Modify in `/opt/gemxpresso.rad3/bin/GxpRADInit.sh` the `JAVA_HOME` if necessary.
For using the GemXpresso RAD III do with bash `export RAD_HOME=/opt/gemxpresso.rad3` if you wish using the communication APIs installed by yourself or do `source /etc/gxp_rad_profile` if you choose to use the APIs installed by the kit.
Do some scripts to set up the environment.
Put the rights `rw` on the device used by the OCF reader.

Note: For using PC/SC with the OCF to PCSC bridge, it is not necessary to have the rights on the device because the pcscd daemon run with the rights allowing the device access.

I have chosen to use my readers on PC/SC and an OCF to PC/SC brigde because most of my reader work with PC/SC and only one with OCF (the GCR410).

8 Installation of the OpenCard Framework

The installOCF.class provided by [5] does not work on my Linux.
Get the root privileges.
`/usr/local/`
`ln -s /opt/gemxpresso.rad3/Ocf1.2/ ocf`
Create a file opencard.properties in the directory `$JAVA_HOME/jre/lib/`
Do some scripts to set up the environment (put the .jar in the CLASSPATH)

9 Installation of the OCF to PC/SC bridge

Download `OCFPCSC1-0.0.1.tar.gz` [1]
Contrary to this is written in the documentation it is possible to have more than one reader in

the /etc/readers.conf

tar xvzf OCFPCSC1-0.0.1.tar.gz
cd OCFPCSC1-0.0.1
Modify the Makefile

```
JDK_HOME  = /usr/local/jdk1.2.2
PCSC_HDRS = -I/usr/local/pcsc/include
PCSC_LIBS = -L/usr/local/pcsc/lib -lpcsclite -lpthread
INCLUDE   = -I$(JDK_HOME)/include -I$(JDK_HOME)/include/linux $(PCSC_HDRS)
```

make
Get the root privileges.
make install

For using the bridge, put libOCFPCSC1.so in your LD_LIBRARY_PATH
Modify the opencard.properties files involved with the line:
OpenCard.terminals = com.ibm.opencard.terminal.pcsc10.Pcsc10CardTerminalFactory

In order to using the readers with the GemXpresso RAD III and the OCF to PC/SC brigde put in the file opencard.properties in the directory /opt/gemxpresso.rad3/conf:
OpenCard.terminals = com.ibm.opencard.terminal.pcsc10.Pcsc10CardTerminalFactory
Modify the CLASSPATH in the file /opt/gemxpresso.rad3/bin/GxpRADInit.sh in the way that follows:

```
# I set OCF_LIB
OCF_LIB=$RAD_HOME/Ocf1.2/lib
```

and

```
#=============================OCF=================================
CLASSPATH=\$CLASSPATH\
:$RAD_LIB/base-core.jar\
:$RAD_LIB/base-opt.jar\
:$RAD_LIB/pcsc_wrapper.jar\
:$OCF_LIB/reference-terminals-windows.jar\
export CLASSPATH
```

10 Installation of JPCSC 0.7.2

unzip jpcsc-72.zip [4]
cd jpcsc
make
Get the root privileges.
make install
If the line ‘‘/usr/local/pcsc/lib’’ doesn’t exist in /etc/ld.so.conf add it.
echo ‘‘/usr/local/pcsc/lib’’ >> /etc/ld.so.conf
ldconfig
For using it, put jpcsc.jar in CLASSPATH et libjpcsc.so in the LD_LIBRARY_PATH

11 Installation of the JCOP 2.1 kit

Get the root privileges.

```
cd /opt
tar xvzf jctools-2.1-linux.tgz
cd /opt/jctools-2.1-linux/etc/ide
chmod 666 classtemplate.txt
chmod 666 scriptheader.txt
chmod 666 methodheader.txt
chmod 666 fieldheader.txt
```

Do some scripts to set up the environment.
For using the examples the user must copy them in an own directory.

12 Installation of the OpenSC

`tar xvzf opensc-0.7.0.tar.gz` [11]

```
./configure --prefix=/usr/local/opensc --with-pcsclite=/usr/local/pcsc
make
```

13 Installation of the Muscle Framework

14 FAQ

Q1: Why does pcsc-lite fail to communicate with the serial port? A1: Be sure that /dev/pcsc/1 is a link to /dev/ttyS0 and not to /dev/ttys0.

References

[1] MUSCLE web site.
`http://www.linuxnet.com/`

[2] GemCore based PC/SC reader drivers.
`http://ludovic.rousseau.free.fr/softwares/ifd-GemPC/`

[3] Muscle PC/SC Lite
`http://alioth.debian.org/projects/pcsclite/`

[4] The JPC/SC specifications and driver.
`http://www.linuxnet.com/middleware/`

[5] OpenCard Framework.
`http://www.opencard.org/`

[6] PC/SC Specifications.
`http://www.pcscworkgroup.com/`

[7] Chip/Smart Card Interface Devices (CCID).
`http://www.usb.org/developers/`

[8] The libusb project home.
`http://libusb.sourceforge.net/`

[9] Linux PCMCIA Information Page.
`http://pcmcia-cs.sourceforge.net/`

[10] Sun microsystems.
Java Card™ 2.2 Specifications.
`http://java.sun.com/products/javacard/`

[11] OpenSC.
`http://www.opensc.org/`

[12] OpenCT.
`http://www.opensc.org/cvs/openct/`