



更多

下一个博客»

创建博客 登录

Ludovic Rousseau's blog

My activities related to smart card and Free Software (as in free speech).

Thursday, April 22, 2010

PCSC sample in Perl

Here is the PCSC sample in Perl language I promised in PC/SC sample in different languages.

Installation

Get the source code from <http://ludovic.rousseau.free.fr/softwares/pcsc-perl/>. The current version is 1.4.8. If you distribution does not provide a package (Debian does with [libpcsc-perl](#)) you can install it by hand using:

```
pcsc-perl-1.4.8$ perl Makefile.PL
osname: linux
LDDFLAGS:
INC: `pkg-config --cflags libpcsclite`
Checking if your kit is complete...
Looks good
Writing Makefile for Chipcard::PCSC::Card
Writing Makefile for Chipcard::PCSC
pcsc-perl-1.4.7$ make
[...]
pcsc-perl-1.4.7$ make test
[...]
pcsc-perl-1.4.7$ make install
[...]
```

The wrapper works on GNU/Linux, Mac OS X and Windows.

API

The API documentation is available online at <http://ludovic.rousseau.free.fr/softwares/pcsc-perl/PCSC.html> and <http://ludovic.rousseau.free.fr/softwares/pcsc-perl/Card.html>.

You can also have a look at the [project page](#) project page on [CPAN](#) (Comprehensive Perl Archive Network).

Source code

```
#!/usr/bin/perl -w

use Chipcard::PCSC;

# create a new object
$hContext = new Chipcard::PCSC();
die ("Can't create the PCSC object: $Chipcard::PCSC::errno\n")
    unless defined $hContext;

# get the reader list
@ReadersList = $hContext->ListReaders();
die ("Can't get readers' list: $Chipcard::PCSC::errno\n")
    unless defined $ReadersList[0];

# connect to the first reader
$hCard = new Chipcard::PCSC::Card($hContext, $ReadersList[0]);
die ("Can't connect: $Chipcard::PCSC::errno\n")
    unless defined $hCard;

# send the Select Applet APDU
```

Google+ Badge

Ludovic Rousseau blog



Follow

Blog Archive

- 2017 (33)
- 2016 (49)
- 2015 (51)
- 2014 (61)
- 2013 (38)
- 2012 (27)
- 2011 (46)
- ▼ 2010 (55)
 - December (5)
 - November (5)
 - October (9)
 - September (1)
 - August (8)
 - July (1)
 - June (10)
 - May (6)
 - ▼ April (10)
 - PCSC sample in Python
 - Free software Tokend above PKCS#11 (for Mac OS X)
 - Source code of PKCS#11 for .NET cards
 - PCSC sample in Perl
 - My Ohloh page
 - pcsc-lite and CCID beta versions
 - PCSC sample in C
 - CCID driver in Mac OS X
 - PC/SC sample in different languages
 - Parsing an ATR

Search This Blog

Subscribe To

- Posts ▼
- Comments ▼

Google+ Followers

```
$cmd = Chipcard::PCSC::ascii_to_array("00 A4 04 00 0A A0 00 00 00 62 03 01 0C 06 01");
$RecvData = $hCard->Transmit($cmd);
die ("Can't transmit data: $Chipcard::PCSC::errno") unless defined $RecvData;
print Chipcard::PCSC::array_to_ascii($RecvData)."\n";

# send the test APDU
$cmd = Chipcard::PCSC::ascii_to_array("00 00 00 00");
$RecvData = $hCard->Transmit($cmd);
die ("Can't transmit data: $Chipcard::PCSC::errno") unless defined $RecvData;
print Chipcard::PCSC::array_to_ascii($RecvData)."\n";

$hCard->Disconnect();
```

Output

```
90 00
48 65 6C 6C 6F 20 77 6F 72 6C 64 21 90 00
```

Lessons learned

Portability

The same code can be used on any platform. No more #ifdef like in C.

Low level API

The API is still low level and just wrap PC/SC calls from C to Perl.

Higher level API

TransmitWithCheck() is a little more easy to use than Transmit(). This method does the split between data and status word.

In the example above replace the two last blocks with:

```
# Send the Select Applet APDU
($sw, $RecvData) = $hCard->TransmitWithCheck("00 A4 04 00 0A A0 00 00 00 62 03 01 0C 06 01", "90 00");
die ("Can't transmit data: $Chipcard::PCSC::errno") unless defined $sw;
print $RecvData."\n";
print Chipcard::PCSC::Card::ISO7816Error($sw) . " ($sw)\n";

# Send the test APDU
($sw, $RecvData) = $hCard->TransmitWithCheck("00 00 00 00", "90 00");
die ("Can't transmit data: $Chipcard::PCSC::errno") unless defined $sw;
print $RecvData."\n";
print map { chr hex $_ } split ' ', $RecvData;
print "\n";
print Chipcard::PCSC::Card::ISO7816Error($sw) . " ($sw)\n";
```

This sample code also uses Chipcard::PCSC::Card::ISO7816Error(\$sw) to transform the status word is something human readable like Normal processing. for 90 00.

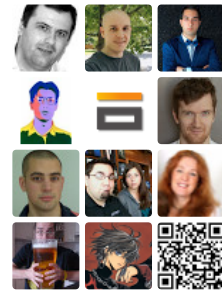
Output

```
Normal processing. (90 00)
48 65 6C 6C 6F 20 77 6F 72 6C 64 21
Hello world!
Normal processing. (90 00)
```



Ludovic Rousseau b...

Follow



336 have us in circles

[View all](#)

Labels: [Perl](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Bitcoin



License: [by-nc-sa](#)



This blog by [Ludovic Rousseau](#) is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](#).

Simple theme. Powered by [Blogger](#).