



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).

Wednesday, May 27, 2015

PCSC sample in C for UEFI

To continue the list of PC/SC wrappers initiated more than four years ago with "[PC/SC sample in different languages](#)" I now present a PC/SC sample written in C but for UEFI.

UEFI

UEFI stands for [Unified Extensible Firmware Interface](#). UEFI is an evolution of EFI. EFI should replace BIOS on "modern" "Intel PC" computers.

Any 64-bits Intel (or compatible) PC computer should have UEFI (and maybe also a BIOS for compatibility).

See my previous article "[UEFI Smart Card Reader Protocol](#)" for some more details.

Sample source code

```
#include <Uefi.h>
#include <Library/UefiLib.h>
#include <Library/ShellCEntryLib.h>
#include <Library/UefiBootServicesTableLib.h>
#include <Library/BaseMemoryLib.h>

#include <Protocol/SmartCardReader.h>

int HelloWorld(EFI_SMART_CARD_READER_PROTOCOL *SmartCardReader)
{
    EFI_STATUS Status;
    UINT32 ActiveProtocol;
    int i;
    UINT8 CAPDU_select[] = {0x00, 0xA4, 0x04, 0x00, 0x0A, 0xA0, 0x00, 0x00, 0x00,
0x62, 0x03, 0x01, 0x0C, 0x06, 0x01};
    UINT8 CAPDU_command[] = {0x00, 0x00, 0x00, 0x00};
    UINTN CAPDULength, RAPDULength;
    UINT8 RAPDU[256+2];

    /*
     * SCardConnect
     */
    Status = SmartCardReader->SCardConnect(SmartCardReader,
        SCARD_AM_CARD,
        SCARD_CA_COLDRESET,
        SCARD_PROTOCOL_T0 | SCARD_PROTOCOL_T1,
        &ActiveProtocol);
    if (EFI_ERROR(Status))
    {
        Print(L"ERROR: SCardConnect: %d\n", Status);
        return 0;
    }

    /*
     * SCardTransmit Select
     */
    CAPDULength = sizeof CAPDU_select;
    RAPDULength = sizeof RAPDU;
    Status = SmartCardReader->SCardTransmit(SmartCardReader,
        CAPDU_select, CAPDULength,
        RAPDU, &RAPDULength);
    if (EFI_ERROR(Status))
```

Google+ Badge

Ludovic Rousseau blog



Follow

Blog Archive

- 2017 (33)
- 2016 (49)
- ▼ 2015 (51)
 - December (8)
 - November (6)
 - October (4)
 - September (3)
 - August (5)
 - July (1)
 - June (4)
 - ▼ May (3)
 - PCSC sample in C for UEFI
 - UEFI Smart Card Reader Protocol
 - New version of libccid: 1.4.19
- April (3)
- March (2)
- February (5)
- January (7)
- 2014 (61)
- 2013 (38)
- 2012 (27)
- 2011 (46)
- 2010 (55)

Search This Blog

Subscribe To



Posts



Comments



Google+ Followers

```

{
    Print(L"ERROR: SCardTransmit: %d\n", Status);
    return 0;
}
Print(L"RAPDU: ");
for (i=0; i<RAPDULength; i++)
    Print(L"%02X ", RAPDU[i]);
Print(L"\n");

/*
 * SCardTransmit Command
 */
CAPDULength = sizeof CAPDU_command;
RAPDULength = sizeof RAPDU;
Status = SmartCardReader->SCardTransmit(SmartCardReader,
    CAPDU_command, CAPDULength,
    RAPDU, &RAPDULength);
if (EFI_ERROR(Status))
{
    Print(L"ERROR: SCardTransmit: %d\n", Status);
    return 0;
}
Print(L"RAPDU: ");
for (i=0; i<RAPDULength; i++)
    Print(L"%02X ", RAPDU[i]);
Print(L"\n");
for (i=0; i<RAPDULength; i++)
    Print(L"%c", RAPDU[i]);
Print(L"\n");

/*
 * SCardDisconnect
 */
Status = SmartCardReader->SCardDisconnect(SmartCardReader,
    SCARD_CA_NORESET);
if (EFI_ERROR(Status))
{
    Print(L"ERROR: SCardDisconnect: %d\n", Status);
    return 0;
}

return 0;
}

INTN
EFIAPI
ShellAppMain (
    IN UINTN Argc,
    IN CHAR16 **Argv
)
{
    EFI_STATUS Status;
    UINTN HandleIndex, HandleCount;
    EFI_HANDLE *DevicePathHandleBuffer = NULL;
    EFI_SMART_CARD_READER_PROTOCOL *SmartCardReader;

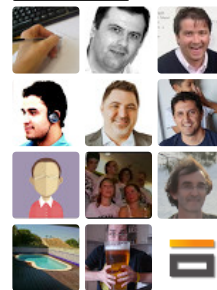
    /* EFI_SMART_CARD_READER_PROTOCOL */
    Status = gBS->LocateHandleBuffer(
        ByProtocol,
        &gEfiSmartCardReaderProtocolGuid,
        NULL,
        &HandleCount,
        &DevicePathHandleBuffer);

    if (EFI_ERROR(Status))
    {
        Print(L"ERROR: Get EFI_SMART_CARD_READER_PROTOCOL count fail.\n");
        return 0;
    }
}

```

Ludovic Rousseau b...

Follow



336 have us in circles

[View all](#)

```

Print(L"Found %d reader(s)\n", HandleCount);
for (HandleIndex = 0; HandleIndex < HandleCount; HandleIndex++)
{
    ZeroMem(&SmartCardReader, sizeof SmartCardReader);

    Status = gBS->HandleProtocol(
        DevicePathHandleBuffer[HandleIndex],
        &gEfiSmartCardReaderProtocolGuid,
        (VOID**)&SmartCardReader);

    if (EFI_ERROR(Status))
    {
        Print(L"ERROR: Open Protocol fail.\n");
        gBS->FreePool(DevicePathHandleBuffer);
        return 0;
    }

    HelloWorld(SmartCardReader);
}
gBS->FreePool(DevicePathHandleBuffer);

return(0);
}

```

Output

```

Found 1 reader(s)
RAPDU: 90 00
RAPDU: 48 65 6C 6C 6F 20 77 6F 72 6C 64 21 90 00
Hello world!

```

Test platform

I used a Dell laptop model E6420 (shipped in August 2013) with an integrated Broadcom smart card reader. This reader was already in the "should work" list of my CCID driver. Of course the SmartCardReader API was not included in the UEFI. I had to write the driver myself.

I made tests with older Dell laptops and had problems with the USB UEFI layer. It looks like UEFI is still a work in progress and bugs/limitations are common.

I started by using a software simulator qemu + UEFI but I could not use a USB device with the simulator. So I rapidly had to use a real hardware. Developing a UEFI driver involves a **lot** of reboot to try a new version of the driver.

Remarks

I added error checks and logging feature in the source code.

The API is not high level but mostly the direct equivalent of WinSCard (the classic C API for smart cards).

Conclusion

It was fun to work on UEFI and learn a new technology.

UEFI is very powerful. Time will tell if the SmartCardReader API will be deployed and used.



Labels: C, ccid, code

[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](#).