Introduction

This document describes a firmware implementation for an Ekey-Udisk device based on ST72651.

The software is divided into 3 parts:

- **USB management**: the main USB operations are managed by a USB library with USB hardware, which is not the focus in this documentation. For further information on this subject, please refer to the ST7 USB software library documentation.
- **NAND Flash interface**: this part software is migrated from PFD solution. How to implement it is not mentioned in this documentation.
- **Smart Card interface**: this interface is implemented by timer and accords ISO7816-1, 2, 3.
1 System Structure

PC Application Software calls 2 interfaces through different drivers:

- **PC SIDE**
  - Microsoft USB driver
  - HID Interface
    - Ekey Function
    - Microsoft HID driver
  - Process HID Protocol

- **DEVICE SIDE**
  - Full Speed USB Lib (Based on ST7 chip)
  - Mass Storage Interface
    - Udisk Function
    - Microsoft Mass Storage driver
  - Process APDU Protocol
    - ISO 7816
  - Process BOT
    - RBC Protocol

NAND Flash Plugin
2 System Diagram

Main Polling USB event

New HID data received?
Yes

Parse APDU Request. Set ETU timer interrupt. Start communication with smart card.

APDU transaction is on-going?
Yes

APDU transaction has finished?
Yes

Send back APDU result

No

New BOT package received?
Yes

Flash operation

No

No

Flash operation has finished?
Yes

Send back BOT status

No
USB Interface

This solution uses a USB Composite Device to combine the Ekey and Udisk functions into one USB device.

The Ekey function uses an HID protocol, and the Udisk function uses a Mass Storage protocol. Both of these two functions use a Microsoft driver, so when an Ekey-Udisk device plugs in, Windows2000/XP recognizes one USB composite device comprised of one HID device and one Mass Storage device. No extra driver is needed.

The USB interface for Udisk is traditional.

The USB interface for Ekey is shown below, and the PC Application Software can use this data structure to communicate with Ekey through the HID driver.

<table>
<thead>
<tr>
<th>Report ID (1 Byte)</th>
<th>Message Length (1 Byte)</th>
<th>Data (1-128 Bytes)</th>
</tr>
</thead>
</table>

Enumerating a USB composite device with HID and a Mass Storage device is listed in the file descriptor.c

According to the HID protocol, PC communication is configured in USER_USB_Setup() in User_lib.c

Note: According to the limitation of the HID driver, in order to save time to send data to the device, PC Application Software should select the correct Report ID based on the Message Length. 9 report types are provided for use with supporting lengths from 0x0f to 0x8f. If necessary, additional descriptors can be created by modifying these to provide other report types.

Table 1. Report ID based on message length

<table>
<thead>
<tr>
<th>ID</th>
<th>Length</th>
<th>Data Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x01~0x0f</td>
<td>Data(0x01~0x0f bytes)</td>
</tr>
<tr>
<td>2</td>
<td>0x01~0x1f</td>
<td>Data(0x01~0x1f bytes)</td>
</tr>
<tr>
<td>3</td>
<td>0x01~0x2f</td>
<td>Data(0x01~0x2f bytes)</td>
</tr>
<tr>
<td>4</td>
<td>0x01~0x3f</td>
<td>Data(0x01~0x3f bytes)</td>
</tr>
<tr>
<td>5</td>
<td>0x01~0x4f</td>
<td>Data(0x01~0x4f bytes)</td>
</tr>
<tr>
<td>6</td>
<td>0x01~0x5f</td>
<td>Data(0x01~0x5f bytes)</td>
</tr>
<tr>
<td>7</td>
<td>0x01~0x6f</td>
<td>Data(0x01~0x6f bytes)</td>
</tr>
<tr>
<td>8</td>
<td>0x01~0x7f</td>
<td>Data(0x01~0x7f bytes)</td>
</tr>
<tr>
<td>9</td>
<td>0x01~0x8f</td>
<td>Data(0x01~0x8f bytes)</td>
</tr>
</tbody>
</table>
4 NAND Flash Interface

This part firmware is migrated from the ST PFD solution, and for related applications, please refer to the PFD application note.

This NAND Flash Interface provides two partitions where one is the Public Partition and the other is the password-protected Security Partition.

Two types of NAND Flash (512B/page & 2kB/page) are supported. Other Flash formats can be supported by modifying software.
Smart Card Interface

This interface provides the user with a set of functions for using the Timer and I/O to communicate with the smart card. It is composed of 3 files:

- **HID_usb.c**
- **UARTT0.c**
- **Int_7265.c**

The **HID_usb.c** file provides the protocol conversion from the HID protocol to the ISO7816 APDU protocol for the main loop program.

The **UARTT0.c** file implements the ISO7816 T0 protocol to communicate with the smart card.

- CardPowerOn() function is called by **HID_usb.c** to provide a voltage to smart card
- CardSend() function is called by **HID_usb.c** to start a UART communication
- T0SendData() function is called by timer interrupt to send one bit at a set time
- T0ReceiveData() function is called by timer interrupt to receive one bit at a set time

The **Int_7265.c** file provides one timer interrupt management and one external interrupt management to implement the ISO7816 T0 protocol.

- INT_TIM() function is coded to call T0SendData() function or T0ReceiveData() function to receive or send one bit when a specific time is reached.
- INT_EI2() function is coded to start the timer when a UART start bit is detected.

The MCO pin of ST72651 is used to provide a clock signal to the smart card. The output of the MCO pin is 6MHz. In this solution, it should reduce to 3MHz to use as smart card clock, if the user wants another clock frequency, an external crystal could provide different clock.

The UsbVddf pin of ST72651 is used to provide Voltage to the smart card. The voltage output by this pin is 3.3V, so this interface is 3.3V. If you want another voltage interface, you should modify the I/O pins.

In this solution, the baud rate of the ISO7816 interface uses a default value of 8.064kbps. If you require another baud rate, the timer value should be modified.
6 Revision history

Table 2. Document revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>02-Dec-2005</td>
<td>0.1</td>
<td>Initial release</td>
</tr>
<tr>
<td>05-Dec-2005</td>
<td>1.0</td>
<td>System flowcharts modified</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Table 1 on page 4 added</em></td>
</tr>
</tbody>
</table>
“THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.”