TERIDIAN SEMICONDUCTOR CORP.

Simplifying System Integration™

73S12xxF USB-CCID Host GUI Users Guide

April 27, 2009
Rev. 1.1
UG_12xxF_037

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

Figures

1. Introduction

1.1 Scope

This user guide describes the use of the USB CCID host application (USB-CCID.exe). This software provides users with a Windows-based GUI application to communicate to the Teridian USB CCID reader firmware running on 73S12xxF-based smart card reader devices. This program allows the user to interface directly to the USB CCID firmware using a PC running Windows XP® and exercise its functionality as a demonstration and test platform.

1.2 Setup

This software was developed in the C# programming language using Microsoft Visual Studio® 2008. This software requires that the Microsoft Windows Smart Card Library (SCLib.dll), also provided by Teridian, be in the same folder as the CCID-USB.exe program.

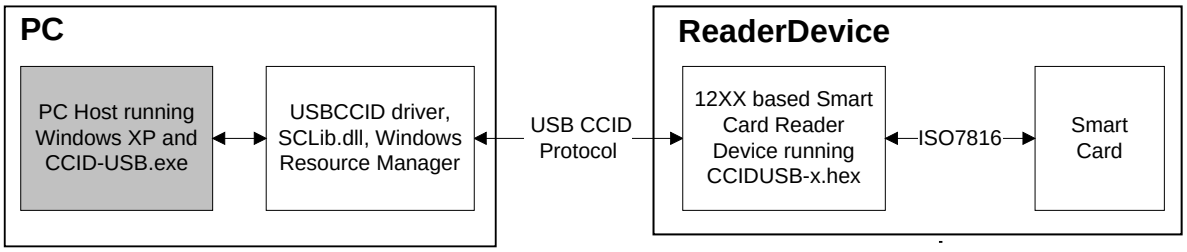

Figure 1: USB CCID System

The setup of the USB CCID system is shown in Figure 1. This document applies to the gray box. The Reader Device (herein referred to as Device) is connected to the host PC USB port via a standard USB cable. When the Device is plugged into a Windows host PC, the Windows Add New Hardware Wizard launches and prompts the user to install a driver for the newly found hardware. The CCID application firmware operates with both the standard Microsoft CCID driver (usbccid.sys) and the Teridian enhanced CCID driver (ccidtsc.sys). The software installation disk contains the Windows .inf file for both drivers. Browse to the directory containing the .inf files; the Wizard displays the dialog as shown in Figure 2. Select a driver and complete the driver installation.

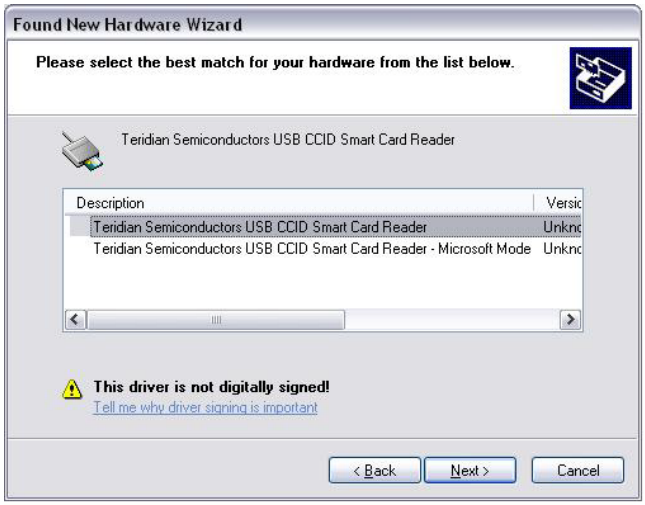

Figure 2: Add New Hardware Wizard Showing TSC CCID and Microsoft Drivers

The features provided by Microsoft's usbccid.sys are detailed in the *73S1212F, 73S1217F 80515 SoC Family with USB, ISO-7816/EMV and PINpad CCID Application Note*. The Teridian enhanced CCID

driver provides CCID compliant APDU level transfer of data and additionally addresses many of the shortcomings of the Microsoft driver, including:

- Support for multiple reader slots.
- PIN pad support.
- LCD support.
- Vendor/device-specific string name support in the device manager.
- Support for mechanical features.

The Device Reader should be powered up and ready to run. The USB-CCID.exe program can be invoked from the Windows taskbar **Run** button. Once the application is opened, the USB-CCID window appears as shown in Figure 3.

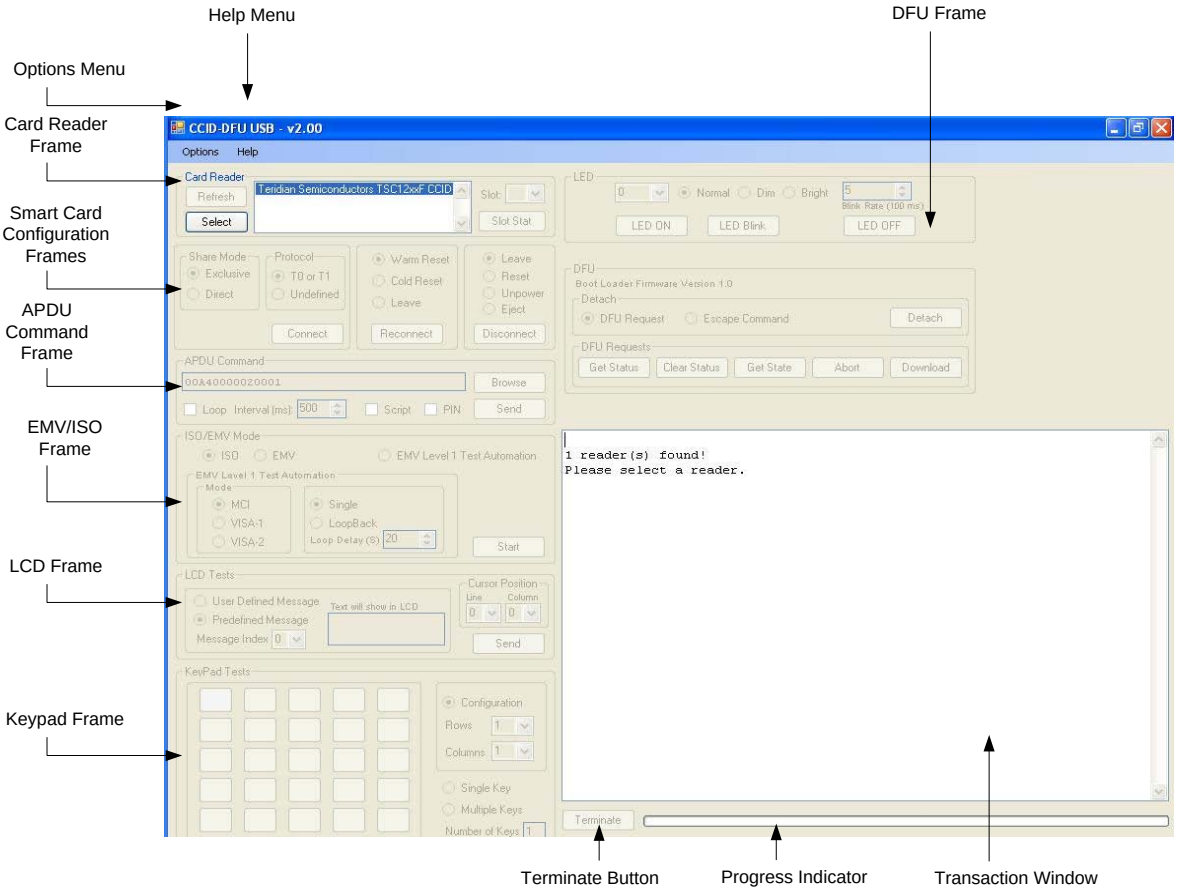

Figure 3: USB CCID Host Application Window

2. Operation

The application window has several different functional frames. A transaction/text window displays all the USB communication traffic based on the USB CCID command/response exchange. The architecture software is defined as a master/slave configuration where the host PC is the master and the reader device is the slave. Communications are defined as a command/response exchange pair where the host will send the command to the device and for each command; the device will always send the response. The C# application uses the SCLib.dll to send commands using the Microsoft API. For more information about the CCID protocol, PC/SC interface, Microsoft Smart Card Library APIs, PCSC APIs, and CCID command packets, refer to the following documents:

1. *Universal Serial Bus Device Class Specification for USB Chip/Smart Card Interface Devices* http://www.usb.org/developers/devclass_docs/DWG_Smart-Card_CCID_Rev110.pdf.
2. PC/SC Workgroup Specifications http://www.pcscworkgroup.com/specifications/specdownload.php
3. Microsoft PC/SC API http://msdn2.microsoft.com/en-us/library/aa380142.aspx

2.1 Frame Descriptions and Use

2.1.1 Card Reader Frame

The Card Reader frame has a list box that displays the active smart card readers. The USB CCID host application automatically detects any insertion and removal of smart card readers, and updates the list box immediately. If no available readers are displayed in the list box, the Refresh button can be used to force a scan for any active readers. Select a reader by highlighting the desired reader in the list box and click “Select”. This will enable the “Connect” frame and indicate which device was selected in the transaction window. The “Slot Stat” button will return the status of the current slot in the transaction window. It will indicate if a card is in the slot or not.

By Windows Resource Manager’s design, a functional Smart Card (i.e. a card with an acceptable ATR) has to be inserted into a slot in order to use any other function. This includes the LCD and KeyPad functions. The Windows resource manager stores the ATR in memory. The slot number drop down box will configure the active slot on the selected reader. For this C# application, the default slot is the slot 0, which is displayed when a reader is selected.

2.1.2 Smart Card Configuration Frames

2.1.2.1 Connect Configuration Frame

The Connect Configuration frame includes the Share Mode and Protocol sub frames. The parameter selection flags are used with the **ScardConnect**() PC/SC command. Refer to the Microsoft Developer Network for an explanation of these parameters. In most cases, these selections can be left at the default setting. Selecting the **Connect** button will connect and activate the card. The card is accessible and commands can be sent to the card. The Card Reader and Connect Configuration frames are disabled and the remaining are enabled, as shown in Figure 4.

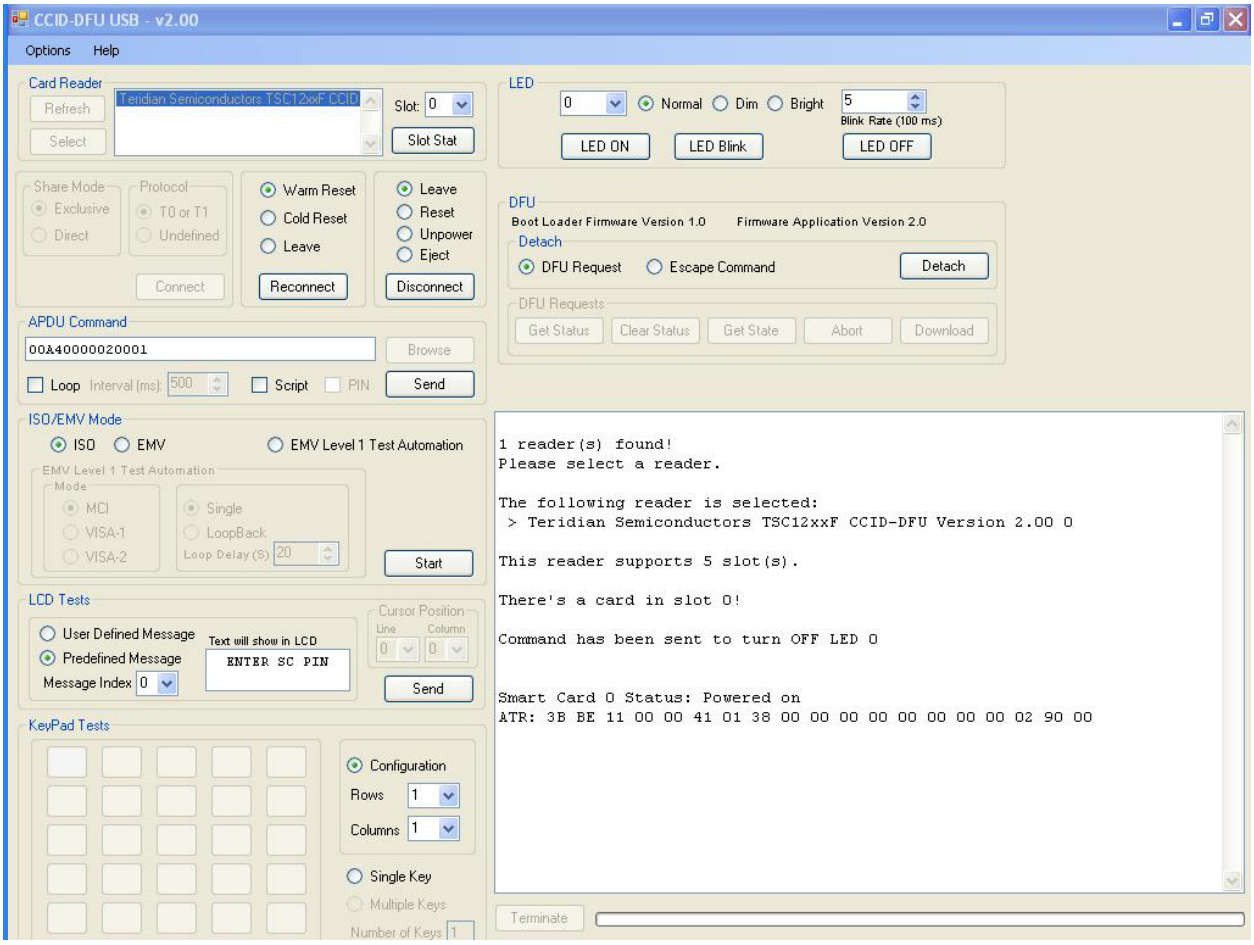

Figure 4: USB CCID Host Application Window after Connecting

2.1.2.2 Reconnect Configuration Frame

The Reconnect Configuration frame has three buttons: **Warm Reset**, **Cold Reset** and **Leave**. These are parameter selection flags used with the **ScardReconnect**() PC/SC command. Refer to the Microsoft Developer Network for a full explanation of these parameters. In most cases, these selections can be left at the default setting.

2.1.2.3 Disconnect Configuration Frame

The Disconnect Configuration frame contains four buttons: **Leave**, **Reset**, **Unpower** and **Reject**. These are parameter selection flags used with the **ScardDisconnect**() PC/SC command. Refer to the Microsoft Developer Network for a full explanation of these parameters. In most cases, these selections can be left at the default setting.

2.1.3 APDU Command Frame

The APDU command frame contains controls to transmit an APDU command to the current smart card slot and to allow repetition based on a specified time interval. The APDU command frame is shown in Figure 5.

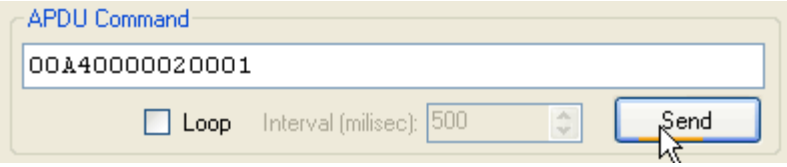

Figure 5: APDU Command Frame

2.1.3.1 APDU Command Text Box

The APDU command text box allows the user to configure a **ScardTransmit()** PC/SC command. The APDU command may be manually entered into this text box. A default APDU is placed into the text box upon the host program initiation. Any valid APDU command can be sent to the device by typing in the text box.

It is the user's responsibility to make sure the APDU commands are of correct format.

2.1.3.2 Loop Checkbox and Interval List Box

Selection of the **Loop** check box configures the current APDU command to repeat at the interval selected in the Interval list box. The **Interval** text box is enabled only when the Loop check box is checked. The interval is based on millisecond increments.

2.1.3.3 Send Button

When the **Send** button is clicked, the APDU command in the APDU command text box is transmitted to the Device via the **ScardTransmit** command. After the Device executes the command, its response is returned and the entire transaction is displayed in the transaction window.

If the **Loop** check box is checked, the command is transmitted repeatedly (see Section 2.1.3.2). All other controls are grayed (disabled) and the **Terminate** button is enabled (located at the left side of the activity bar). The activity bar will scroll to indicate the test is running. The transaction window displays the serial traffic to and from the reader where data preceded with "=>" are host to Device transfers (commands) and "<=" are Device to host transfers (responses). See Section 2.1.7 for the **Terminate** button details.

2.1.4 ISO/EMV Mode Frame

The ISO/EMV Mode frame has controls to configure the card interface for EMV or ISO-7816 compliance and execute the EMV Loopback test mode. The test mode is used for EMV Level I certification testing according to each test labs' test scenarios or test flowchart requirement. The USB CCID host application currently supports three EMV Loopback test modes:

- One for Master Card (MCI) according to Cetecom test lab flowchart.
- One test mode for Visa 1 – according to FIME and RFI test lab flowchart.
- One test mode for Visa 2 – according to ICT-K test lab flowchart.

The test details are described in Appendix A. The EMV/ISO Mode frame is shown in Figure 6.

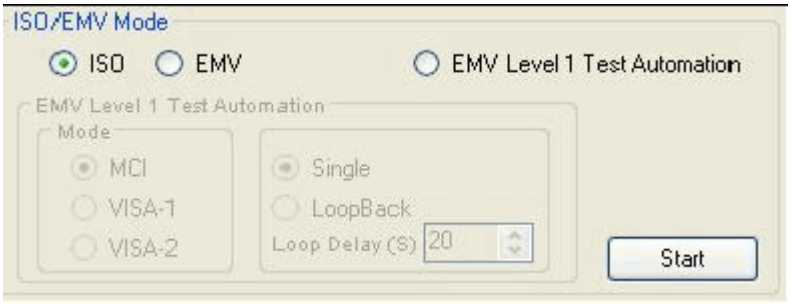

Figure 6: EMV/ISO Mode Frame

It is recommended that the loop delay be set to at least 20 seconds. Teridian's internal EMV Level I test has shown better results with delay setting at 25 seconds. This is due to the long cycle of some tests and especially the way Windows' Resource Manager handles automatic retry activations of Smart Card. For example, when a test might be wrapping up with test logs, if the delay is too short, the C# application might be calling to activate a card for the next test. The response will be a Card Mute because the card side is still wrapping up the last test. Windows Resource Manager will automatically make two more retries to activate the Smart Card. These subsequent activation commands can cause the card to fail the next test because the test is not anticipating them.

Caveat: There is a potential issue with the Windows Resource Manager if the EMV Level I test is run in a particular case. When a test indicates a failure instead of returning a good ATR (this is to test the reader handling of an error-ATR); if the next test has a good ATR and expects a Select File command to be sent, this next test will fail. When there is an error return code from the device in response to any command, the next successful activation (with a good ATR) will be followed immediately with a deactivation command by Resource Manager. This deactivation disrupts the test sequence causing the test failure. If this occurs, rerun the test making sure that the previous command prior to the rerun was a success. *This problem occurs whether the Teridian or the Microsoft CCID driver is used.*

2.1.4.1 ISO and EMV Select Buttons

The **ISO** and **EMV** select buttons allow the user to select the compliant mode of the card session for the selected slot. The interactions between the reader and card are different based on ISO-7816 or EMV compatibility. The mode selection must be made prior to sending any APDU commands or initiation of loopback testing.

2.1.4.2 EMV Level 1 Test Automation Button

Selecting the **EMV Level 1 Test Automation** button enables the EMV Level 1 Test Automation sub-frame. Within this sub-frame are two more sub-frames that allow the selection of the specific EMV tests and setup. The EMV test types are; MCI (Cetecom Lab), VISA-1 (RFI Global, FIME test labs) and VISA-2 (ICT-K lab) and iteration selections; single or loopback. If the **LoopBack** button is selected, the **LoopBack Interval** list box is enabled. This will configure the EMV test to repeat based on the interval selected based in one-second increments. The default repeat time is 20 seconds, which is the minimal recommended value. Based on our internal testing, 25 seconds is optimal for most labs.

2.1.4.3 Start Button

When the **Start** button is clicked, a sequence of commands is sent to the Device. The Device needs to be configured to go into EMV Test Mode and which Test Mode it should process. This process is done via the CCID's Escape(*) command. Windows requires that a Smart Card be activated BEFORE any commands are acceptable with a functional Smart Card inserted into slot 0. The Windows Resource Manager automatically sends an activation command to the Device to activate it. Once this is done, check the **LoopBack** button. If loopback test is desired, set the Loop Delay to 25 seconds. The Escape command will be sent to the Device to prepare it for the specific EMV test mode. Now exchange the functional Smart Card with the test card and click the **OK** button to start the loop test.

The EMV Test sequence begins with the PowerON command. After receiving a successful response (i.e. device response with an ATR string), a null APDU command will be sent to the device via the BlockTransfer command. A null APDU is interpreted by the device as the EMV PSE Select File test command based on the proper EMV test environment. After the device executes the command, a response will be returned with two status bytes. The application then sends a PowerOFF command to the device. The device will return a response to the PowerOFF command, which terminates the iteration of the Loopback test. The entire transaction is displayed in the transaction window

If the **LoopBack** button is selected when the **Start** button was clicked, a delay based on the value of the Interval list box will be initiated. When this delay completes, the entire EMV Test sequence is repeated indefinitely. The **Terminate** button is enabled, and all other controls are grayed (disabled). The activity bar will scroll to indicate the test is running. The test will repeat until the **Terminate** button is selected. See Section 2.1.7 for the **Terminate** button details.

If the Microsoft usbccid.sys driver is used, special Windows Registry setup is required in order for the Escape command to pass through Windows' Resource Manager to the device. It is recommended that the TSCccid.sys driver be used for EMV Testing. *Once the device is put in test Mode, it will be in test mode permanently until the next Escape command or until it is reset.*

2.1.5 LCD Tests Frame

The LCD Tests frame contains two sub-frames, one for a predefined message and another for a user defined message. Selecting the corresponding button will enable each frame and disable the other. Figure 7 shows the LCD Tests Frame.

Figure 7: LCD Tests Frame

2.1.5.1 Predefined Message

The predefined message frame allows the selection of four pre-defined messages. The **Message Index** dropdown box has three indexes listed as 0, 1 and 2. When each index is selected, the message is displayed in the LCD display box exactly as it appears on a 2 row by a 16 column LCD display. Messages are shown in Figure 8. Clicking the **Send** button transmits the command to the device and the corresponding message appears on the LCD display. Note the **Cursor Position** sub-frame is disabled when the predefined message button is selected as the line and column selections are not used.

This command should not be executed if the device does not contain an LCD display.

Message Index: 0

Row	Column
	0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
	0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
0	_ _ E N T E R _ S C _ P I N _ _
1	_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Message Index: 1

Row	Column
	0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
	0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
0	_ E N T E R _ N E W _ P I N _ _
1	_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Message Index: 2

Row	Column
	0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
	0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
0	C O N F I R M _ N E W _ P I N _
1	_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Figure 8: Predefined Messages

2.1.5.2 User Defined Message

The User Defined Message allows the user to write any alphanumeric message to the LCD. The message text is entered into the LCD text box and the **Cursor Position** sub-frame enables the **Line** and **Column** dropdown boxes. This allows the user to select exactly where in the LCD display the text will begin to be written. The default setting is Line 0 and Column 0. This corresponds to the upper leftmost cursor position in the LCD display. The lower rightmost cursor position is defined as Line 1, Column 15. Only one character can be written at this location as it is the last position in the LCD display and any subsequent characters will be concatenated. The message is shown in the LCD display box exactly as it appears on an LCD display. All other data on the display that is not defined in the specified text locations will remain unchanged. Clicking the **Send** button transfers the command and display the text to the device's LCD display.

This command should not be executed if the device does not contain an LCD display.

2.1.6 KeyPad Tests Frame

The KeyPad Tests frame contains two sub-frames, one that displays a six row by a five column image matrix and the other to configure the keypad matrix. Figure 9 shows the KeyPad Tests Frame. The number of rows and columns configuration must be configured before any keypad operations can be initiated. The image matrix remains grayed until the configuration is sent to the device. Clicking the **Send** button sends the configuration packet. The keypad image matrix enables those keys as defined by the **Rows** and **Columns** settings. The selection is justified at the upper left as being row one and column one. The **Configuration** button will be de-selected and sub-frame will be disabled. Once the configuration has been set up, it will maintain that configuration until the **Configuration** button is selected again. The **Single Key** and **Multiple Keys** buttons are enabled after the keypad is configured.

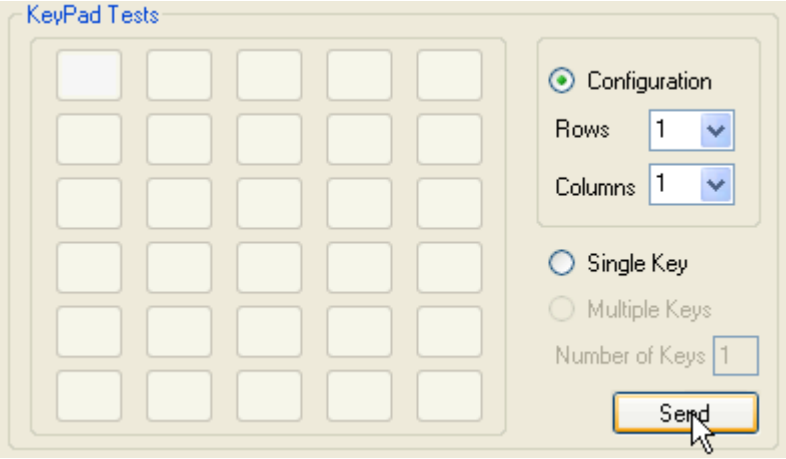

Figure 9: KeyPad Tests Frame

2.1.6.1 Single Key Button

Clicking the **Single Key** button enables the single key function in the device. This function waits for a keystroke on a specific key location. All other keys are ignored. The corresponding key location on the image can be clicked. This turns that key location a shade of Blue indicating that the corresponding key location on the devices keypad is expected to be pressed by the user. Clicking the **Send** button enables this function. All other frames are disabled. The device waits indefinitely until a keystroke is performed on the specific key location. When this occurs, the response message is noted in the transaction window and all frames are restored to the state before the single key command was performed.

2.1.6.2 Multiple Keys Button

Selecting the **Multiple Keys** button enables the multiple keys function in the device. This function waits for a specific number (n) of keystrokes and returns a log of the keystrokes in the response after the nth keystroke. The image matrix locations is displayed in the transaction window as well as embedded in the command response packet.

2.1.7 LED Frame

The LED functionality is active only in CCID mode and is not active in DFU mode. The LED frame contains LED number selection, three choices for the LED light intensity (**Normal**, **Dim** and **Bright**), blink rate menu bar selection (or type in) and three **LED ON**, **OFF** and **Blink** Command Icons. LED number is 0 as default. The **LED ON** and **LED OFF** Commands would turn ON and OFF the LED, respectively. The LED brightness can be checked from the three choices. The LED blink rate range is from 1 to 255; each count is equivalent to 100 msec LED blink duration. The default blink value is 5 which is equivalent to 500 msec (5 X 100 msec) blink duration. When the LED brightness and the blink rate are selected, then pressing the LED Blink Command would activate the LED and it would blink at inputted rate. When the Host GUI starts, the LED would be ON or OFF depending if a Smart Card is in or not, respectively. In order to for the LED start blinking, at least one Smart Card needs to be inserted and connected (HOST GUI **Connect** command is pressed).

2.1.8 DFU Frame

The DFU frame is divided into two sub-frames. The Detach sub-frame is used to initiate a DFU process. The DFU Requests sub-frame consists of five DFU command buttons that can be used to test DFU requests individually. Additionally, the DFU frame displays the version numbers of the boot loader and firmware application that are currently running in the smart card reader.

When a smart card reader is selected, the Detach sub-frame is enabled, but the DFU Requests sub-frame remains disabled until the smart card reader enters DFU mode.

Figure 10: DFU Frame

2.1.8.1 Detach Sub-frame

A smart card reader normally operates in CCID mode. To initiate a firmware upgrade, the host application must detach the smart card reader so that the reader will reset and boot up again as a DFU device. The host application performs this detachment by sending either a DFU Detach (DFU_DETACH) or CCID Escape command to the reader. Select either the **DFU Request** or **Escape Command** button, and then click the **Detach** button. The smart card reader resets and disappears from the Card Reader list. It then reboots in DFU mode. The host application automatically detects this DFU device and displays its name in the **Card Reader** list box.

At this point, the host application disables the Detach sub-frame. Selecting the DFU device activates the DFU Requests sub-frame.

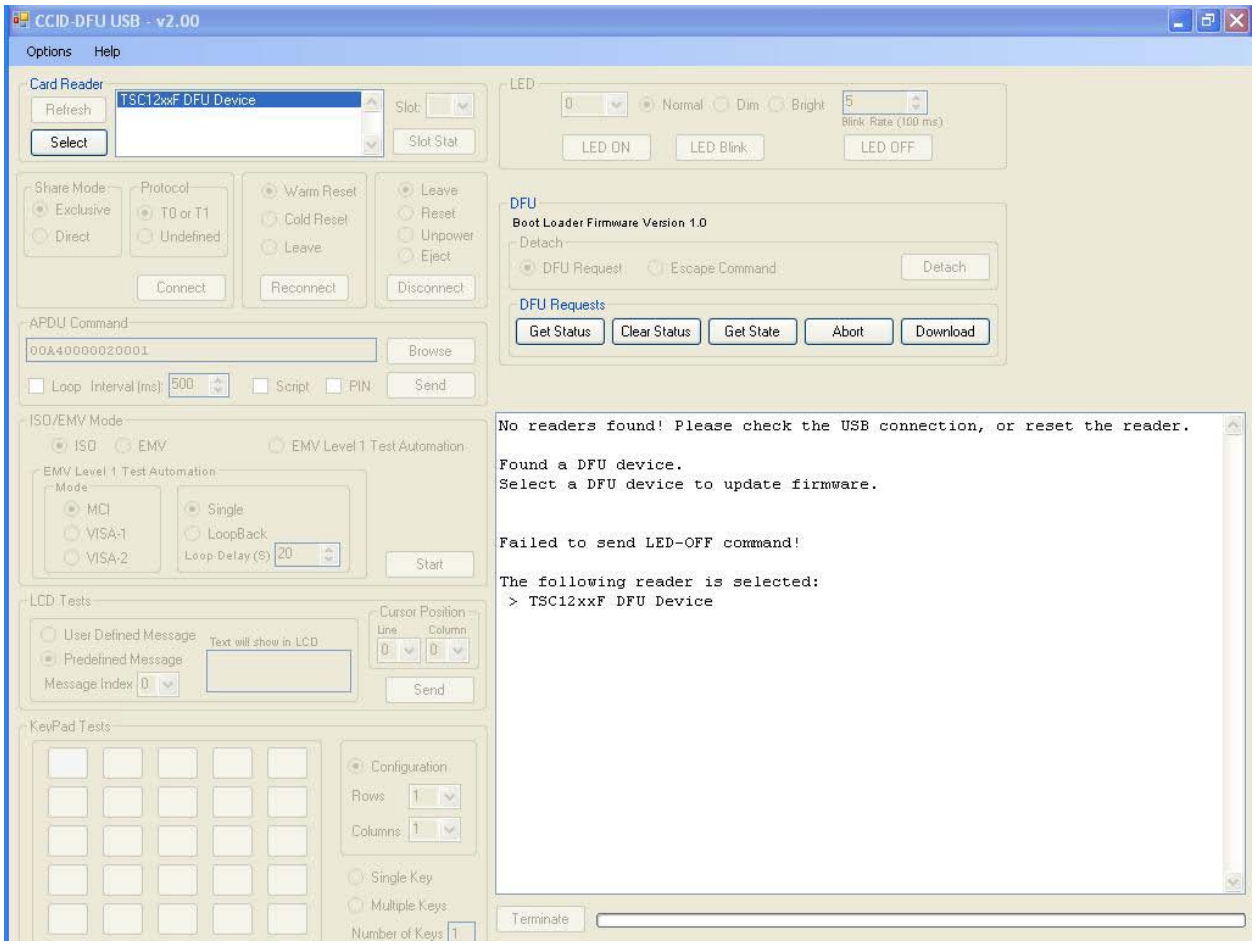

Figure 11: DFU Request Sub-frame Is Activated after a DFU Device Has Been Selected

2.1.8.2 DFU Requests Sub-frame

A smart card reader that has transitioned into the DFU mode is referred as a DFU device by the host application program. The DFU Requests sub-frame is enabled only after a DFU device in the Card Reader list box has been selected. This sub-frame contains five buttons each of which, when clicked, causes a particular DFU command to be sent to the DFU device. These DFU requests are: DFU_DNLOAD (DFU Download), DFU_GETSTATUS, DFU_CLRSTATUS (DFU Clear Status), DFU_GETSTATE, and DFU_ABORT.

2.1.8.2.1 Download

Clicking the **Download** button starts the operation of firmware download. A file browser window appears, allowing the user to specify a CCID firmware image file that will be downloaded to the DFU device. The firmware image file should be in Intel hex format with a .hex file name extension. Upon the file selection, the host application displays a warning message requesting a confirmation from the user to proceed with the download.

Figure 12: Download Warning Message

If the user chooses to continue, the host application shows a second warning message reminding the user not to power down the DFU device during the download.

If the user proceeds with the download, the host application starts transferring the selected firmware image to the DFU device. During the data transfer, the host application displays information indicating the progress of the download. As the data is transmitted to the DFU device, the boot loader in the DFU device writes the new version of the CCID firmware into the flash, permanently erasing the older version.

After the completion of the firmware upgrade, the smart card reader resets one more time, disappearing from the Card Reader list. It is now necessary to manually power down the smart card reader and then power back on. The smart card reader will boot up and run in CCID mode normally.

2.1.8.2.2 Get Status

Use the **Get Status** button to retrieve current status from the DFU device. This button causes a DFU_GETSTATUS command to be transmitted to the DFU device. When the host application receives a response, it displays the status information of the DFU device in the Transaction window.

2.1.8.2.3 Clear Status

When the **Clear Status** button is clicked, the host application sends out a DFU_CLRSTATUS command to the DFU device to force it to clear the current status and transition to the DFU Idle state. Most importantly, if the DFU mode firmware in the smart card reader has detected an error, it will remain in the DFU Error state until it has received a DFU_CLRSTATUS command from the host application.

2.1.8.2.4 Get State

The **Get State** button makes the host application transmit a DFU_GETSTATE command to the DFU device. Upon receipt of the command, the DFU device reports its current DFU state. The host application then displays this information in the Transaction window.

2.1.8.2.5 Abort

The host application sends out a DFU_ABORT request to the DFU device when the Abort button is pressed. This request forces the DFU device to exit from its current state and return to the DFU Idle state.

2.1.9 Terminate Button

To cancel any of the loopback tests, click the **Terminate** button – it becomes gray and the other controls will be restored to the state before clicking the associated **Start** button. The current sequence is allowed to complete before terminating the final iteration of the test.

2.1.10 Options Menu

The **Options** menu contains optional functions that assist the user operate the firmware and the host application more efficiently. Figure 13 shows the **Options** menu.

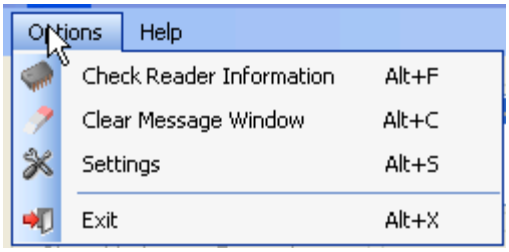

Figure 13: Options Menu

The first selection, **Check Reader Information**, opens a dialog box that contains information regarding the USB-CCID driver as shown in Figure 14.

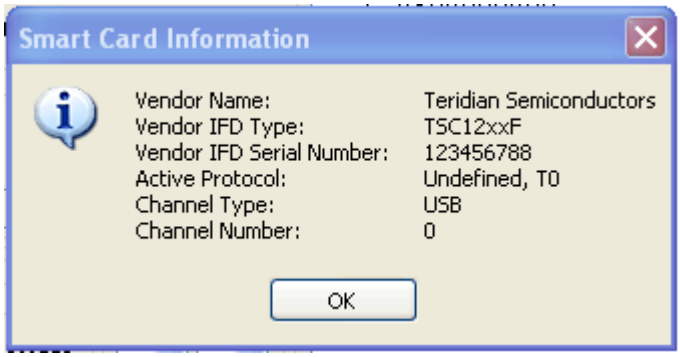

Figure 14: Smart Card Information Dialog Box

The second item, **Clear Message Window**, clears all the transactions in the transaction window. The last item, **Settings**, opens a dialog box with two tabs. The first tab is shown in Figure 15.

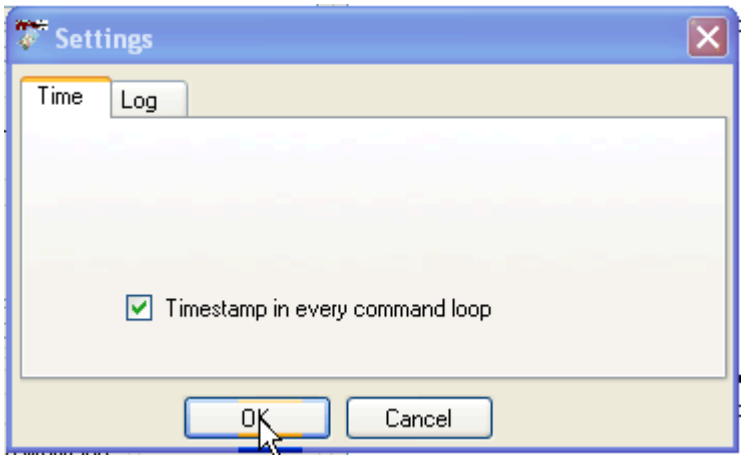

Figure 15: Time Stamp Selection Dialog Box

The Time tab enables or disables a time and date stamp when running loopback tests. This is applicable to both the transaction window and log files. The second tab configures the log file as shown in Figure 16. It enables or disables the log file and allows the log file to be named.

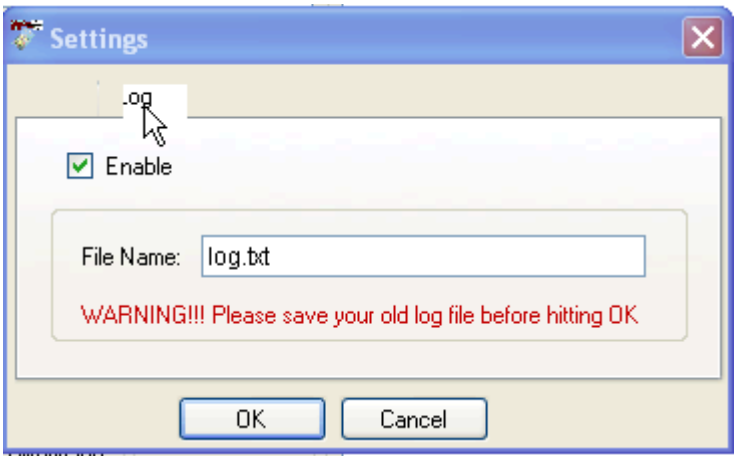

Figure 16: Log Dialog Box

.

3. 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

4. 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.

Appendix A – EMV Level I Certification Test

EMV compliant test suite follows its specification written for Payment System Environment. There are several test labs, listed on EMVco.com website, qualified to perform these tests. At this writing, there are two available Protocol test suites that either one, will/can be used to qualify for EMV Level I compliance. In other words, passing either one of these suites will qualify as EMV Level I compliance. The USB-CCID code is written to link to the two TSC libraries (LAPI and HAPI) that comply with both tests. However, since each lab has its own test scripts and the test scripts are different according to the lab's setup, the application layer must be written, and catered, specifically for each test lab's requirements.

EMV Test Mode

An EMV test (or session) is defined to be commands that run from Activation of the card to Deactivation of the card. A Block Transfer may or may not happen in the session depends on the the card's ATR response. The host may invoke the EMV PSE test environment via the Escape command. The first parameter byte (B1) of the Escape command needs to specifically tell whether a test mode is invoked and if so, it should be invoked using MCI, VISA-1 or VISA-2 test environment. Review the Escape command section for details about this test mode.

When the test starts, this host application will send a PowerOn command. Following the successful PowerOn command, the host needs to send the Block Transfer Command. The format and content of the APDU command, during EMV Test Mode, will be ignored by the device as it handles the Select File Command according to each lab's requirement. The Command loopback will be handled by the firmware and upon finishing up one test, the firmware will respond to the host with status of whether a test session run with successful return code or not. NOTE : an unsuccessful return code may or may not be a failed test. The test verdict (test passed or test failed) is determined only on the card side. The following flow chart depicts the minimal coding required on the host side to invoke EMV PSE test environment. For detailed description of each environment's flowcharts, see the CCIDAN_73S12xxV2.00.pdf document.

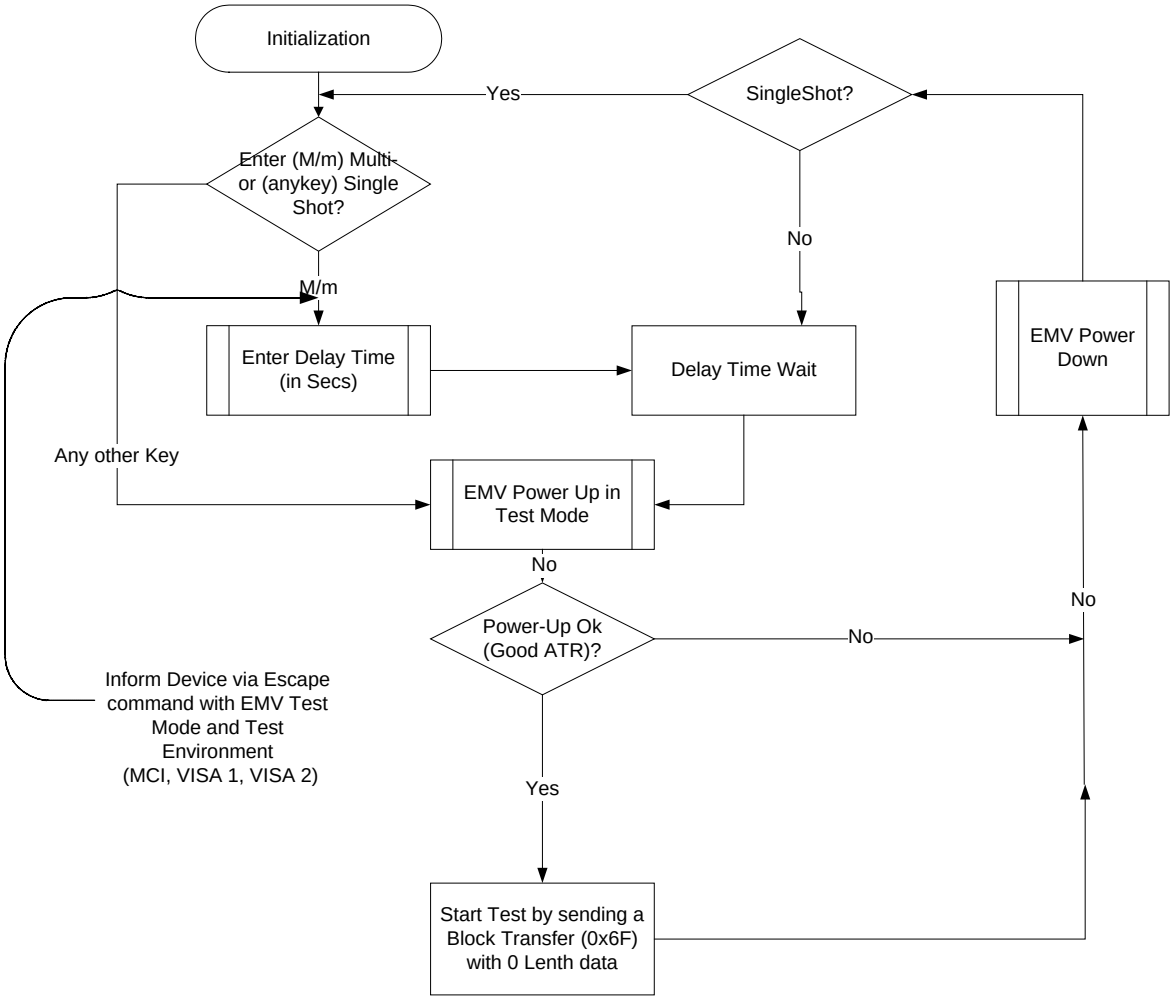

Figure 17: Invoking the EMV PSE Test Environment

Revision History

Revision	Date	Description
1.0	2/27/2009	First publication.
1.1	4/27/2009	Added Section 2.1.7 LED Frame and Section 2.1.8 DFU Frame. Reformatted document in Teridian style.