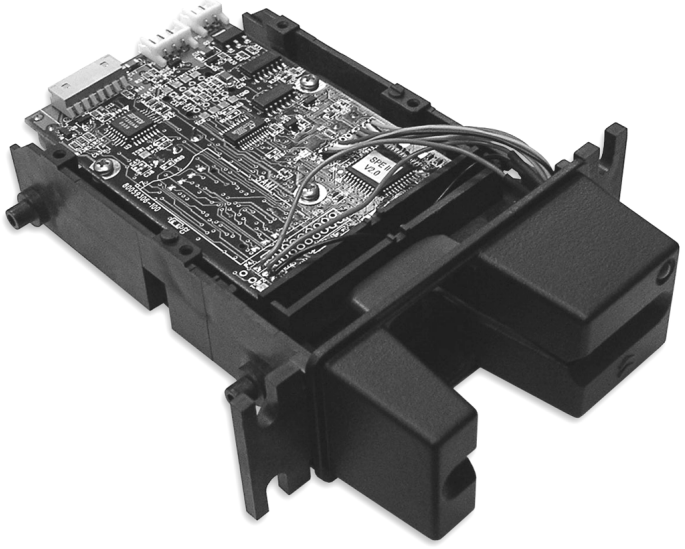
SPECTRUM™ II

Full Insert Magnetic Stripe Reader and Smart Card Reader-Writer PC/SC

User's Manual



Designed for
Microsoft®
Windows®XP
Windows®2000

EMV™

CE

Version 2.1 Rev. A
80059506-002
6/04
#465

IDTECH®
Value through Innovation

Warning

This equipment has not been tested to comply with Part 15 of the FCC Rules for a digital device. This device is designed to be incorporated into a product that will be tested to comply with all regulatory requirements.

Certifications

A letter of approval, affirming reader hardware and firmware conformance to EMV 2000 ICC Specifications for Payment Systems Version 4.0, was issued on May 21, 2004.

The PC/SC driver was certified and signed by the Windows Hardware Quality Control Lab on March 24, 2004. This driver is certified for Windows 2000 and Windows XP (32 bit versions only).

CE certification has also been obtained.

Limited Warranty

ID TECH warrants this product to be in good working order for a period of one year from the date of purchase. If this product is not in good working order as warranted above, or should this product fail to be in good working order at any time during the warranty period, repair or replacement shall be provided by ID TECH.

This warranty does not cover incidental or consequential damages incurred by consumer misuse, or modification of said product. For limited warranty service during the warranty period, please contact ID TECH to obtain an RMA number and instructions for returning the product.

ID TECH
1047 South Placentia Avenue
Fullerton, CA 92831
(714) 680-5868
www.idt-net.com

Table of Contents

Section 1
INTRODUCTION

Description

The ID TECH Spectrum II Hybrid Insert Reader can be configured to read 1, 2, or 3 tracks of magnetic stripe data from cards conforming to ISO 7810 and 7811 standards. It can also read and write data to IC cards conforming to ISO 7810, 7816, and Europay, MasterCard, and Visa (EMV) standards.

APPEARANCE AND FEATURES

ID TECH's Spectrum II Hybrid Insert Reader can read and write data to IC cards conforming to ISO 7810, 7816, and Europay, MasterCard, and Visa (EMV) standards. It can also be configured to read 1, 2, or 3 tracks of magnetic stripe data from cards conforming to ISO 7810 and 7811 standards.

All communication (i.e. reader and message configuration set-up) is accomplished via PC/SC, a standard interface established by leading personal computer and smart card companies to enable smart card applications to work independently of a particular reader manufacturer. Specifically, the ID TECH Hybrid Insert Reader works in PC/SC on the Windows 2000 and Windows XP platforms (32 bit only).

PC/SC is a standard for smart cards only. Because the PC/SC driver converts PC/SC commands to the commands used natively by the reader, configuring the reader to read smart cards alone is largely unnecessary. However, the PC/SC interface only defines T=0 and T=1 microprocessor interface cards. Other functions, such as reading memory cards, using Security Access Modules (SAMS), and reading cards with magnetic stripes, are supported with vendor-specific calls. ID TECH's PC/SC driver, for instance, has been designed to support a latch feature for its readers (to latch smart cards in place during processing) and the reading of magnetic stripe cards. Settings for the magnetic stripe functions are set at the factory, but these default settings can be changed using the commands in this manual.

MECHANICAL CHARACTERISTICS

The reader supports 3 and 5 VDC microprocessor or memory cards and has automatic hardware protection for card removal, ESD, supply voltage drop, short circuit, and overheating in compliance with EMV requirements. Its chassis is molded from a glass and lubricant-filled engineering plastic that supports the magnetic head and electronics, and serves as a guide for the card. The head is spring-loaded for good contact with warped or bowed cards. The smart card connector has gold-plated landing-style contacts that "land" on the ICC's contacts and create a .02 inch "wiping" action to ensure a dependable connection and long life.

The card latch mechanism is designed to secure the card in the insert reader while a card transaction is in process. The programmable latch function is driven by a DC electric motor controlled by either the host or the reader, as desired.

The reader has an open internal design that allows foreign matter entering through the bezel (up to half a card) to fall away and not clog or otherwise obstruct the card path. It has multiple mounting options, and can be securely mounted in any position. A separate molded bezel attaches to the chassis to guide the card, and serves as a mount for a tri-colored LED.

ENVIRONMENTAL CHARACTERISTICS

Environmentally, the reader is designed to function at temperatures ranging from 32° to 131° F (0° to 55° C) in humidity up to 95% non-condensing. It can be stored or shipped at more extreme temperatures ranging from -40° to 158° F (-40° to 70° C).

FACTORY-INSTALLED OPTIONS

-A single tri-colored LED.
-Standard or metal bezel.
-Conformal coating for active electrical components.
-Latching Mechanism (to secure a smart card in the reader).
-Gate (to guard against dirt, dust, and casual vandalism).
-SAMs (Secure Access Modules).

OPTIONAL ACCESSORIES

-DB9 serial cable assembly with female plug for AC/DC adapter.
-AC/DC power adapter complying to user-defined power requirements.
-A sample initialized T=0 microprocessor smart card.
-A sample initialized memory smart card.
-PC/SC sample demonstration programs in Visual C++ and Visual Basic.

DEVELOPMENT TOOLS

To aid in application software development, a Dynamic Link Library (DLL) has been created for use with Windows 2000 and XP editions. In addition, a Windows-based demonstration program and associated test card(s) are available to test the reader and familiarize the user with its various configuration options. Please see Appendix H: Demonstration Software and Card for details.

EMV HYBRID READER AND PC/SC EMV HYBRID READER DIFFERENCES

This PC/SC hybrid reader has been derived from the Spectrum II RS-232 hybrid reader. The connector and cable between the reader and the PC have been changed to eight wires from seven. (This change was made to support Plug-and-Play. The reader must be configured for Plug-and-Play in order to support the PC/SC interface.)

Section 2
PC/SC

General Description

The purpose of PC/SC is to simplify the integration of smart cards and the PC operating system. (Currently, this Spectrum reader supports Windows2000 and WindowsXP.) It does this by creating a common application interface to smart cards.

What that means is that PC/SC makes the card application independent of reader manufacturer. If an application accesses one manufacturer's reader via PC/SC, that same application should be able to access the card though any smart card reader, regardless of manufacturer.

PC/SC is not, however, a full solution for a reader that includes features beyond smart cards. The insufficiencies of PC/SC when using a hybrid reader include:

· No built-in support for magnetic stripe reading
· No built-in support for card latching
· No built-in support for memory cards
· Requires a "seated" smart card for normal operation
· Access to the reader is only available through the PC/SC interface.

All of these problems have been solved with the development of an ID TECH DLL (supplied) that makes the interfacing to these functions seamless to the application.

PC/SC AND EMV

The EMV certification does not require PC/SC and PC/SC certification does not require EMV. To some extent, in fact, the certifications are incompatible. The purpose of EMV is to support interoperable financial transactions, while the purpose of PC/SC is to support easy connection to smart cards.

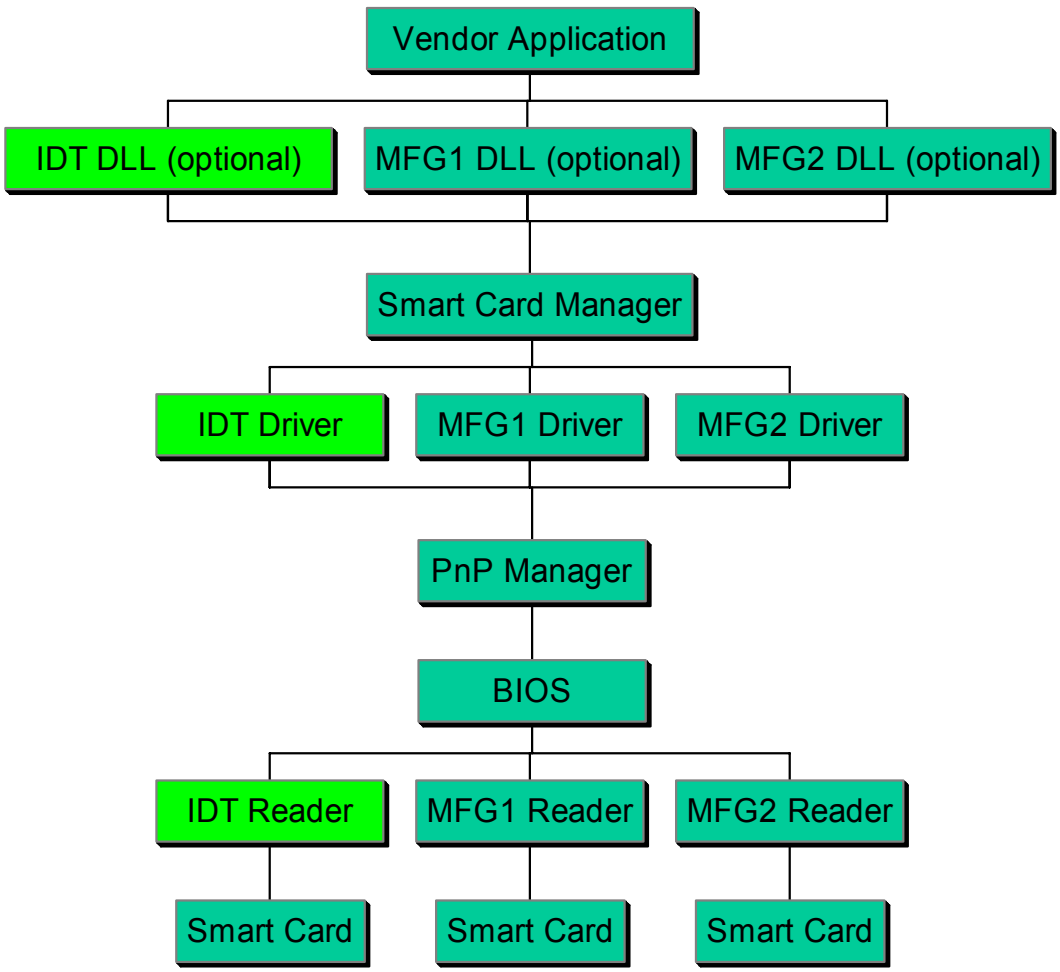
PC/SC certification requires support for a range of cards that are incompatible with EMV. To handle this, the Spectrum reader has the configuration setting EMV/ISO. If the reader is configured for EMV (the default) it will not properly handle a range of cards that are supported by other PC/SC readers. And if the reader is configured for ISO cards, it will not properly handle certain EMV transactions.

EMV was created by the banking world to provide true interoperability and is frequently required for certified banking applications. EMV is a voluminous and detailed specification based on ISO7816, but EMV is more restrictive and requires extensive certification testing. Europay, MasterCard, and Visa created this specification and defines the guidelines governing debit and credit transactions. This ID TECH reader is EMV level I certified. A system into which the reader is incorporated will require EMV level II certification.

IDTLIB.DLL

The purpose of the IdtLib.dll is to simplify access to reader/vendor-specific functions like magnetic stripe reading and latch control. This library is not required; all of these functions can be done by the application without requiring this DLL. This DLL is provided, however, to make the programming of an application to utilize the reader's functions simpler.

SIMPLIFIED PC/SC COMPONENT ORGANIZATION



Section 3
INSTALLATION AND TESTING

The Spectrum II reader with the standard bezel can be mounted via the front mounting flanges that are part of the bezel assembly. Side mounting studs are also provided.

The reader can be mounted in any orientation, but the preferred position is with the board on top (so that debris will not accumulate on the smart card connector). The reader accepts an 8-pin Molex 51004-0800 female connector. The I/O connector pin-outs are provided in Appendix F. Data is transmitted to the host in an ASCII data format.

To test the reader to ensure it is working properly, please follow these steps:

1. Connect the supplied serial cable between the reader and the host PC.

2. Plug the 5 volt “wall hugger” power supply to an external power source. Connect the power jack to the back of the DB9 connector.

3. Boot or reboot the PC. When prompted, insert the CD with the signed (certified) PC/SC driver in the CD disk player and and select the appropriate drive. (If the PC boots without recognizing that new hardware has been installed, your PC is probably not configured for Plug-and-Play on the serial connections. If the supplied reader has an LED, and that LED remains amber throughout the boot process, this is the case. Refer to the manufacturer’s instructions on configuring the BIOS for Plug-and-Play.)

4. Follow the instructions to install the driver to support the PC/SC connection to the reader. At this point, the hybrid reader’s LED should be green.

5. Install the demonstration software by double-clicking on \demo\setup.exe in the provided disk. Follow the installation instructions.

6. Once installed, select RUN/PROGRAM/SPEC II PCSC VC DEMO/IDT PCSC DEMO SP2 to run the program.

To establish that the reader is properly connected and functioning, select the CONNECT TO READER button. Then select GET READER VERSION. Then select DISCONNECT.

To ensure that the smart card reader works, fully insert the provided sample T=0 smart card. Then select CONNECT TO CARD. Demonstrate that the card is functional by entering “00 A4 00 00 02 31 40” in the command box, and then selecting SEND APDU. This will select file 3140. A “90 00” response should be returned from the card. To issue a second command to read the first 4 bytes of that file, enter “90 52 00 00 04”.

To demonstrate the magnetic stripe reader, fully insert a card with a magnetic stripe. Select GET MAGSTRIPE DATA. Within 60 seconds, pull the card in one clean motion from the reader. The contents of tracks 1, 2, and 3 will display in the response area.

Driver Installation Notes

The current ID TECH PC/SC drivers are designed to connect to an unused RS-232 port on a PC running Windows 2000 or XP. (Windows operating systems newer than Windows 98 SE may work.) There is no install program for the drivers. The normal way to install the drivers is to connect the hybrid reader to an available RS-232 port on a PC running an appropriate operating system and boot (or reboot) the PC.

Instead of rebooting, the user can also scan for new hardware. To scan for new hardware, select START, then SETTINGS, then CONTROL PANEL, then ADD/REMOVE HARDWARE, then NEXT, then ADD/TROUBLESHOOT A DEVICE, then NEXT. When prompted for the location of the drivers, select the location that the drivers are typically located at, such as a floppy in drive A.

Three files are necessary: its0020.inf, its0020.sys, and its0020.cat, which is the certification catalog. Drivers with catalog files are “signed” drivers. Unless the user is specifically instructed to install an unreleased driver, only signed drivers should be used.

The serial port must be configured for Plug-and-Play (PNP). If the LED on the reader does not switch from amber to green during the scan for new devices, it will be necessary to check the BIOS in the PC to verify that it is configured for PNP. Striking a specific key early during the boot process will normally enter the BIOS. The computer will normally display the proper key to press.

If the reader has been previously installed on the system, it will install without prompting from the PC for the driver installation disk. If it has not been previously installed, the prompt NEW HARDWARE INSTALLED will appear, and the user will be led through the installation of new hardware. If the LED is not green, go to CONTROL PANEL/ADMINISTRATIVE TOOLS/SERVICES/SMART CARD. Right click and activate. The LED should begin flashing.

To verify proper operation, install and run the ID TECH PC/SC demo program. Then a valid smart card must be inserted in the reader. The provided IDT T=0 smart card will do well. If you connect to the card, you must disconnect from the card anytime the card has been removed from the reader. Functions such as latch and unlatch should function properly.

If the inserted smart card does not work, it may not be a valid EMV smart card. Changing the default mode of the reader from EMV cards to ISO cards by selecting ISO CARD under SET CARD OPTION can test this. (Please see page 17 for details.) To restore the reader to the original state, re-select EMV CARD.

Section 4
DEFINITIONS

The following are definitions of common terms and abbreviations used throughout this manual.

AAMVA	American Association of Motor Vehicle Administrators
ASCII	American Standard Characters for Information Exchange
ATR	Answer To Reset: The POWER ON response from a ICC
Asynchronous card	A microprocessor ICC that follows ISO7816-3
BIOS	Basic Input Output System: Low-level hardware support entered before the operating systems loads
BPI	Bits Per Inch
C-APDU	Command-Application Protocol Data Unit
CDL	California Drivers License Format
EMV	Europay-MasterCard-Visa: Financial transaction certification
F/D Ratio	F=Frequency, D=Divisor. In ISO 7816-3, these terms are used as a ratio of oscillator frequency to determine speed of the ICC interface
Host	The Personal Computer (PC) to which the reader is attached.
IFD	Interface Device: This is how ISO 7816 refers to the reader
ICC	Integrated Circuit Card
IPS	Inches Per Second
ISO	International Standards Organization
Hex	Hexadecimal, Base 16
Hybrid	The combination of two technologies, such as smart card and magnetic stripe
LED	Light Emitting Diode
LRC	Longitudinal Redundancy Check
Microprocessor card	Either a T=0 or T=1 asynchronous card
Memory card	An ICC: Commands supported vary by manufacturer, but generally these cards are simpler and less expensive than microprocessor cards
MSDN	Microsoft Developers Network
MSR	Magnetic Stripe Reader
PnP or PNP	Plug and Play
PC/SC or PCSC	Personal Computer/Smart Card standard interface
PSC P	rogrammable Security Code
R-APDU	Response-Application Protocol Data Unit
RS232	Reference Standard serial communication
Smart Card	An ICC (includes both memory and microprocessor cards)
Synchronous card	See: Memory card
Transport code	A password used between the ICC manufacturer and the card customizer
WHQL	Windows Hardware Quality Control Labs

Section 5
RELATED DOCUMENTS AND LINKS

EMV 2000	Integrated Circuit Card Specifications for Payment Systems Version 4.0, December, 2000 (Amendment January 2003)
ISO 7810	Identification cards – Physical characteristics (1995)
ISO 7811	Identification cards –Recording technique (1995)
ISO/IEC 7816	Identification Cards - Integrated circuit(s) cards with contacts Part 2: Dimension and location of the contacts (1989, DIS 1998) Part 3: Electronic signals and transmission protocols (1997) Part 4: Inter-industry commands for interchange (1995/Amd.1:1997) Amd1: Impact of secure messaging on the structures of apdu messages.
AAMVA	Best Practices Guidelines for the Use of Magnetic Stripes

ID TECH DOCUMENTS

PC/SC Programming in Visual C++
Describes Function calls and parameters to access smart cards using Visual C++ through the PC/SC interface.

PC/SC Programming in Visual Basic
Describes Function calls and parameters to access smart cards using Visual Basic through the PC/SC interface.

RELATED LINKS

AAMVA	http://www.aamva.org/
EMV	http://www.emvco.com/
ID TECHÒ	http://www.idt-net.com/
MSDN	http://msdn.microsoft.com/library/
PC/SC	http://www.pcscworkgroup.com/
SDK Support Windows	SmartCardSDK@discuss.Microsoft.com
Catalog	http://www.windowscatalog.com

Section 6
PROGRAMMING PC/SC IN C++

Although no attempt has been made here to document all possible smart card calls, PC/SC functions to perform a simple transaction to a smart card are listed below in the order that they would be needed. The following would be a typical link to get the full context of these smart card functions from the Microsoft Developers Network (MSDN) library:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/security/scardestablishcontext.asp

Also included is a typical function call that could be used to perform the various PC/SC functions.

For programming functions that are vendor specific, like magnetic stripe reading and latch, see APPENDIX H: ID TECH PC/SC IDTLIB.DLL GUIDE.

For a sample program demonstrating the use of these functions see Sample PC/SC VC++ Demonstration Program.

SCARDESTABLISHCONTEXT

The **SCardEstablishContext** function opens a communication context with the resource manager context (the scope) within which database operations are performed.

```
SCardEstablishContext(SCARD_SCOPE_USER, NULL, NULL, &ContextHandle);
```

SCARDCONNECT

The **SCardConnect** function establishes a connection (using a specific resource manager context) between the calling application and a smart card contained by a specific reader. If no card exists in the specified reader, an error is returned.

```
SCardConnect(ContextHandle, ReaderName[ReaderCount],
SCARD_SHARE_SHARED, SCARD_PROTOCOL_T0 |
SCARD_PROTOCOL_T1, &CardHandle, &ActiveProtocol);
```

SCARDTRANSMIT

The **SCardTransmit** function sends a service request to the smart card, and expects to receive data back from the card.

```
SCardTransmit(CardHandle, &IO_Request, scardSendBuff, sendLength, NULL,
scardRecvBuff, &recvLength);
```

SCARDTRANSMIT

The **SCardTransmit** function sends a service request to the smart card, and expects to receive data back from the card.

```
SCardTransmit(CardHandle, &IO_Request, scardSendBuff, sendLength, NULL,
scardRecvBuff, &recvLength);
```

SCARDDISCONNECT

The **SCardDisconnect** function terminates a connection previously opened between the calling application and a smart card in the reader.

```
SCardDisconnect(CardHandle, SCARD_EJECT_CARD);
```

SCARDRELEASECONTEXT

The **SCardReleaseContext** function closes an established resource manager context, freeing any resources allocated under that context, including SCARDHANDLE objects and memory allocated using the SCARD_AUTOALLOCATE length designator.

```
SCardReleaseContext(ContextHandle);
```

Table 1 – SCARD Error to Value

Error Code	Hexadecimal Value
SCARD_F_INTERNAL_ERROR	80100001
SCARD_E_CANCELLED	80100002
SCARD_E_INVALID_HANDLE	80100003
SCARD_E_INVALID_PARAMETER	80100004
SCARD_E_INVALID_TARGET	80100005
SCARD_E_NO_MEMORY	80100006
SCARD_F_WAITED_TOO_LONG	80100007
SCARD_E_INSUFFICIENT_BUFFER	80100008
SCARD_E_UNKNOWN_READER	80100009
SCARD_E_TIMEOUT	8010000A
SCARD_E_SHARING_VIOLATION	8010000B
SCARD_E_NO_SMARTCARD	8010000C
SCARD_E_UNKNOWN_CARD	8010000D
SCARD_E_CANT_DISPOSE	8010000E
SCARD_E_PROTO_MISMATCH	8010000F
SCARD_E_NOT_READY	80100010
SCARD_E_INVALID_VALUE	80100011
SCARD_E_SYSTEM_CANCELLED	80100012
SCARD_F_COMM_ERROR	80100013

Table 1 – SCARD Error to Value	
Error Code	**Hexadecimal Value**
SCARD_F_UNKNOWN_ERROR	80100014
SCARD_E_INVALID_ATR	80100015
SCARD_E_NOT_TRANSACTED	80100016
SCARD_E_READER_UNAVAILABLE	80100017
SCARD_P_SHUTDOWN	80100018
SCARD_E_PCI_TOO_SMALL	80100019
SCARD_E_READER_UNSUPPORTED	8010001A
SCARD_E_DUPLICATE_READER	8010001B
SCARD_E_CARD_UNSUPPORTED	8010001C
SCARD_E_NO_SERVICE	8010001D
SCARD_E_SERVICE_STOPPED	8010001E
SCARD_E_UNEXPECTED	8010001F
SCARD_E_ICC_INSTALLATION	80100020
SCARD_E_ICC_CREATEORDER	80100021
SCARD_E_UNSUPPORTED_FEATURE	80100022
SCARD_E_DIR_NOT_FOUND	80100023
SCARD_E_FILE_NOT_FOUND	80100024
SCARD_E_NO_DIR	80100025
SCARD_E_NO_FILE	80100026
SCARD_E_NO_ACCESS	80100027
SCARD_E_WRITE_TOO_MANY	80100028
SCARD_E_BAD_SEEK	80100029
SCARD_E_INVALID_CHV	8010002A
SCARD_E_UNKNOWN_RES_MNG	8010002B
SCARD_E_NO_SUCH_CERTIFICATE	8010002C
SCARD_E_CERTIFICATE_UNAVAILABLE	8010002D
SCARD_E_NO_READERS_AVAILABLE	8010002E
SCARD_E_COMM_DATA_LOST	8010002F
SCARD_W_UNSUPPORTED_CARD	80100065
SCARD_W_UNRESPONSIVE_CARD	80100066
SCARD_W_UNPOWERED_CARD	80100067
SCARD_W_RESET_CARD	80100068
SCARD_W_REMOVED_CARD	80100069
SCARD_W_SECURITY_VIOLATION	8010006A
SCARD_W_WRONG_CHV	8010006B
SCARD_W_CHV_BLOCKED	8010006C
SCARD_W_EOF	8010006D
SCARD_W_CANCELLED_BY_USER	8010006E

Table 2 - SCARD Error Text Description	
	Text Description
SCARD_E_BAD_SEEK	An error occurred in setting the smart card file object pointer.
SCARD_E_CANCELLED	The action was canceled by a SCardCancel request.
SCARD_E_CANT_DISPOSE	The system could not dispose of the media in the requested manner.
SCARD_E_CARD_UNSUPPORTED	The smart card does not meet minimal requirements for support.
SCARD_E_CERTIFICATE_UNAVAILABLE	The requested certificate could not be obtained.
SCARD_E_COMM_DATA_LOST	A communications error with the smart card has been detected.
SCARD_E_DIR_NOT_FOUND	The specified directory does not exist in the smart card.
SCARD_E_DUPLICATE_READER	The reader driver didn't produce a unique reader name.
SCARD_E_FILE_NOT_FOUND	The specified file does not exist in the smart card.
SCARD_E_ICC_CREATEORDER	The requested order of object creation is not supported.
SCARD_E_ICC_INSTALLATION	No primary provider can be found for the smart card.
SCARD_E_INSUFFICIENT_BUFFER	The data buffer for returned data is too small for the returned data.
SCARD_E_INVALID_ATR	An ATR string obtained from the registry is not a valid ATR string.
SCARD_E_INVALID_CHV	The supplied PIN is incorrect.
SCARD_E_INVALID_HANDLE	The supplied handle was invalid.
SCARD_E_INVALID_PARAMETER	One or more of the supplied parameters could not be properly interpreted.
SCARD_E_INVALID_TARGET	Registry startup information is missing or invalid.
SCARD_E_INVALID_VALUE	One or more of the supplied parameter values could not be properly interpreted.
SCARD_E_NO_ACCESS	Access is denied to the file.
SCARD_E_NO_DIR	The supplied path does not represent a smart card directory.
SCARD_E_NO_FILE	The supplied path does not represent a smart card file.
SCARD_E_NO_MEMORY	Not enough memory available to complete this command.
SCARD_E_NO_READERS_AVAILABLE	No smart card reader is available.
SCARD_E_NO_SERVICE	The smart card resource manager is not running.
SCARD_E_NO_SMARTCARD	The operation requires a smart card but no smart card is currently in the device.
SCARD_E_NO_SUCH_CERTIFICATE	The requested certificate does not exist.,
SCARD_E_NOT_READY	The reader or card is not ready to accept commands.
SCARD_E_NOT_TRANSACTED	An attempt was made to end a non-existent transaction.
SCARD_E_PCI_TOO_SMALL	The PCI receive buffer was too small.

Table 2 - SCARD Error Text Description	
	Text Description
SCARD_E_PROTO_MISMATCH	The requested protocols are incompatible with the protocol currently in use with the card.
SCARD_E_READER_UNAVAILABLE	The specified reader is not currently available for use.
SCARD_E_READER_UNSUPPORTED	The reader driver does not meet minimal requirements for support.
SCARD_E_SERVICE_STOPPED	The smart card resource manager has shut down.
SCARD_E_SHARING_VIOLATION	The smart card cannot be accessed because of other outstanding connections.
SCARD_E_SYSTEM_CANCELLED	The action was canceled by the system, presumably to log off or shut down.
SCARD_E_TIMEOUT	The user-specified timeout value has expired.
SCARD_E_UNEXPECTED	An unexpected card error has occurred.
SCARD_E_UNKNOWN_CARD	The specified smart card name is not recognized.
SCARD_E_UNKNOWN_READER	The specified reader name is not recognized, reader maybe disconnected.
SCARD_E_UNKNOWN_RES_MNG	An unrecognized error code was returned from a layered component.
SCARD_E_UNSUPPORTED_FEATURE	This smart card does not support the requested feature.
SCARD_E_WRITE_TOO_MANY	An attempt was made to write more data than would fit in the target object.
SCARD_F_COMM_ERROR	An internal communications error has been detected.
SCARD_F_INTERNAL_ERROR	An internal consistency check failed.
SCARD_F_UNKNOWN_ERROR	An internal error has been detected, but the source is unknown.
SCARD_F_WAITED_TOO_LONG	An internal consistency timer has expired.
SCARD_P_SHUTDOWN	The operation has been aborted to allow the server application to exit.
SCARD_S_SUCCESS	No error was encountered.
SCARD_W_CANCELLED_BY_USER	The action was cancelled by the user.
SCARD_W_CHV_BLOCKED	The card cannot be accessed because the maximum number of PIN entry attempts has been reached.
SCARD_W_EOF	The end of the smart card file has been reached.
SCARD_W_REMOVED_CARD	The smart card has been removed, so that further communication is not possible.
SCARD_W_RESET_CARD	The smart card has been reset, so any shared state information is invalid.
SCARD_W_SECURITY_VIOLATION	Access was denied because of a security violation.
SCARD_W_UNPOWERED_CARD	Power has been removed from the smart card, so that further communication is not possible.
SCARD_W_UNRESPONSIVE_CARD	The smart card is not responding to a reset.
SCARD_W_UNSUPPORTED_CARD	The reader cannot communicate with the card, due to ATR string configuration conflicts.
SCARD_W_WRONG_CHV	The card cannot be accessed because the wrong PIN was presented.

Section 7
GENERAL COMMANDS

Following are explanations and examples of of the proper format and command content to send commands to the reader. All commands and characters are expressed in hex format and contained in brackets:

Command Structure

Every command follows the same basic structure:

HEADER	DATA	TRAILER

The HEADER consists of <60> followed by <Command Length>

The DATA consists of Function ID, Function Length, and Function Data

The TRAILER consists of <LRC> followed by <ETX>

Example of LRC

The Longitudinal Redundancy Check (LRC) is calculated by taking “Exclusive OR” (Modulus 2) of all characters preceding it. The total, with LRC, is equal to zero. For example, the following command means “read 16 bytes of data from <00 00>.”

<60><00><06><42><DA><B0><00><00><10><5E><03>

<5E> is the LRC character. It is derived from the following:

Characters	#1	#2	
<60>	0110	0000	
<00>	0000	0000	
<06>	0000	0110	
<42>	0100	0010	
<DA>	1101	1010	
<B0>	1011	0000	
<00>	0000	0000	
<00>	0000	0000	
<10>	0001	0000	
<5E>	0101	1110	<Result of Exclusive OR>

When sending a command:

```
                                  Func
                                  | ID |
<60><Command Length><53>[<xx><Len><FuncData>]<LRC><ETX>
          x x       x x                x x
```

The response confirming the command structure will be:

<60><00><02><90><00><LRC><ETX>

In this example:
<Command Length> is a two-byte counter from <53> to the end of <FuncData>.
<FuncID> is the total of contents, a respective command, and one byte that identifies the particular function affected.
<Len> is a one byte length count for the <FuncData> block.
<FuncData> is the data block for the function.
<ETX> = 03h

The overall <LRC> (Modulus 2 = Exclusive OR) checksum (from <60> to <LRC>) should be zero. See page 14 for an example of the LRC calculation. The following table is a summary of the general commands described in this section:

HEAD *<60><Command Length>*	**DATA** *<XX><XX>*	**NAME**	**USAGE**
60 00 04	53 41 01 xx	Get Baud Rate	To set the rate of serial communication
60 00 02	52 1F	Get Settings	To retrieve current settings
60 00 02	4C 01	Latch On	To close the latch
60 00 02	4C 00	Latch Off	To release the latch
60 00 01	39	Get Version	To get the version of the reader's firmware
60 00 01	24	Get Reader Status	To get reader status in the form of a single byte
60 00 04	53 10 01 XX	Set Card Option	To elect either a memory card or a microprocessor card, 5v or 3v
60 00 04	53 11 01 XX	Set Operation Mode	To set various operational characteristics
60 00 04	53 12 01 XX	Set Memory Card Type	To define which memory card to use
60 00 02	6C XX	Host LED Control	To change LED color or turn it off
60 00 01	49	Reset the Reader	Returns the reader to its default state

SET BAUD RATE

The rate of serial communication between the host application and the reader is set to 38,400 bps after power-up, but the host application can also select 9,600 bps if desired. The reader will go to the selected baud rate after sending back a response to the SET BAUD RATE command. The host applicaton should go to the new baud rate after receiving that response.

To set the desired baud rate, enter the command:
<60><00><06><53><41><01><Baud Rate Setting><LRC><ETX>

Where the baud rate setting is either:
<35> for 9,600 bps or
<37> for 38,400 bps

The response will be: <60><02><90><00><LRC><ETX>

GET SETTINGS

<60><00><02><52><1F><LRC><ETX>
|Func|
ID

This command retrieves all current settings. The reader sends back an acknowledgement <ACK> and a response that is a collection of many function-setting blocks. Each function-setting block has the following format:

<FuncID><Len><FuncData>
XX

LATCH-ON COMMAND

<60><00><02><4C><01><2F><03>

This command is used to close the latch.

The response will be: <60><00><02><90><00><F2><03>

Note: Latch works only when the reader is equipped with the latch option and card has been fully inserted.

LATCH-OFF COMMAND

<60><00><02><4C><00><2E><03>

This command is used to release the latch.

The response will be: <60><00><02><90><00><F2><03>
Note: The first LATCH OFF command does not work if someone has manually set the latch to ON.

GET FIRMWARE VERSION

<60><00><01><39><58><03>

The response will be: <60><00><Len><Version><LRC><03>

The response varies by the reader firmware version.

GET READER STATUS

<60><00><01><24><45><03>

A single byte reader status will be returned.

Bit Position	0	1
0	IC power not ready	IC power ready
1	Card not seated*	Card seated*
2	Latch released*	Latch closed*
3	Card not present	Card present
4	No magnetic data*	Magnetic data present*
5-7	Unused	

** Note: Flags are available only when optional features are supported by the reader. The flag will always be 0 if an option is not supported.*

The response will be: <60><00><01><Reader Status><LRC><03>

SET CARD OPTION

<60><00><04><53><10><01><Setting><LRC><03>

A single byte setting is defined as follows:

Bit Position	0	1
0-3	unused	
4	**EMV Card**	ISO Card
5	**3V Off**	3V On
6	**5V On**	5V Off
7	**Microprocessor Card**	Memory Card

The response will be: <60><00><02><Return Status><LRC><03>

If both the 3V and 5V options are set to ON, the reader will try 3V when powering up a microprocessor card, and then 5V if the card fails to power up at 3V. The reader will try only 3V when powering up a memory card.

Note: The Spectrum II Hybrid Insert Reader is set at the factory to read and write to smart cards conforming to Europay, MasterCard, and Visa (EMV) standards. However, not all smart cards are EMV-compliant. If the reader remains set to read and write to EMV-compliant cards only, the reader will only read EMV cards. If the reader, therefore, is expected to read a wider range of cards than just EMV-specific cards, it should be set to read cards conforming to ISO 7810 and 7816.

Please see Appendix B for Return Status Code Explanations.

SET OPERATION MODE

<60><00><04><53><11><01><Setting><LRC><ETX>

A single byte setting is defined as follows:

Bit Position	0	1
0	PC/SC Mode Off	**PC/SC Mode On***
1	Card Seated Change Off	**Card Seated Change On**
2	Card Present Change Off	**Card Present Change On**
3	MSR Data Envelope Off	**MSR Data Envelope On**
4	**LED Controlled by Reader**	LED Controlled by Host
5	**Magnetic Data Present Off**	Magnetic Data Present On
6	**Standard Decoder**	Raw Data Decoder
7	**Unused**	

The response will be: <60><00><02><Return Status><LRC><03>

* To ensure proper communication with the reader, always have the PC/SC Mode set to ON. The Operation Mode Setting cannot be reset by a default command.

The “Magnetic Data Present” option is only available when the unit has been set to buffered mode.

The “Raw Data Decoder” enables raw data to be sent to the host for further processing. Two ASCII characters represent each raw data byte: The first ASCII character is for the high digit of the hex code. The second ASCII character is for the low digit of the hex code. For example, the characters “4” and “B” represent raw data “4Bh” (01001011).

If “Raw Data Decoder” has been sent, all data will be treated as a bit string and will be sent out in hex format. Leading or trailing zeros (depending on whether the reader reads on insertion or withdrawal) will not be sent.

Note: A “Card switch change notification” (<60><00><02><B0><Card Status><LRC><03>) will be issued by the reader if “Card seated change” or “Card present change” has been set to ON and the card switch was changed.

After a good read, the magnetic stripe data will be sent out with an envelope (<60><Len_H><Len_L><Card data indication 1><Card indication 2><Magstripe data><LRC><03>), if “MSR Data Envelope” is ON. Otherwise, magnetic stripe data will be sent out without an envelope (<Magstripe data>).

<Card data indication 1> (<Cx>) is an ID to indicate magnetic data.

Bit Position	
0-3	Unused
4	0
5	0
6	1
7	1

<Card data indication 2> flags the current read.

Bit Position	0	1
0	Track 1 decode fail	Track 1 decode success
1	Track 2 decode fail	Track 2 deocde success
2	Track 3 decode fail	Track 3 decode success
3	No track 1 data	Track 1 data exists
4	No track 2 data	Track 2 data exists
5	No track 3 data	Track 3 data exists
6-7	Unused	

Note: Track X decode flag available only when track X data exists.

SET MEMORY CARD TYPE

<60><00><04><53><12><01><Setting><LRC><03>

A single byte defines the card type as follows:

Setting	Description
<00>	**Three-byte I²C memory card**
<01>	Four-byte I²C memory card
<02>	SLE4428 card
<03>	SLE4442 card
<04>	GPM276 card
<05>	GPM271 card
<06>	AT88SC101 card

The response will be: <60><00><02><Return Status><LRC><03>

HOST LED CONTROL

The LED can either be controlled by the reader or the host. (The choice between reader and host control can be made by setting a bit in the Notification Option.) This command sets the LED when it is to be controlled by the host. <60><00><02><6C><LED Status><LRC><03>

The LED status can be set as:
<30> Set LED to off
<31> Set LED to Green
<32> Set LED to Red
<33> Set LED to Amber

The response will be: <60><00><02><90><00><F2><03>

RESET THE READER

This command allows the host to return the reader to its default state (card not powered, not armed to read, latch should open, no magnetic or ICC data stored, etc.) It does not restore Operation Mode settings to their default settings; these must be done individually.

<60><00><01><49><28><03>

The response will be: <60><00><02><90><00><F2><03>

Section 8
SMART CARD COMMANDS

Communication Timing

The Spectrum II Insert Reader supports 3 and 5 VDC IC cards, and provides automatic hardware protection for card removal, ESD, supply voltage drop, short circuit, and overheating.

Following are the command/response protocols for communication between the hybrid reader (and ICC media) and the host. All commands and characters are presented in 'Hex' and (< >) brackets.

Successful Command

Host to Hybrid Reader:
<60> <XX XX> <nnnnnnnnnnnnnnnnnnnn><ZZ><03>
ACK Command Length Data (C-APDU) LRC ETX

Hybrid Reader to Host:
<60> <UU UU> <nnnnnnnnnnnnnnnnnnnn><SS><ZZ><03>
ACK Command Length Data (R-APDU) Status LRC 03

Unsuccessful Command

Host to Hybrid Reader:
<60> <XX XX> <nnnnnnnnnnnnnnnnnnnn><ZZ><03>
ACK Command Length Data (C-APDU) LRC 03

Hybrid Reader to Host:
<E0> <UU UU> <SS SS><ZZ><03>
NACK Command Length Status LRC 03

or:
<E0> <UU UU> <nnnnnnnnnnnnnnnnnnnn> <SS SS><TT><03>
NACK Command Length Data (R-APDU) Status LRC 03

The status either defines a successful transaction or gives an error code. Please see Appendix B for the Status Code table.

Command Structure

Every command follows the same basic structure:

HEADER	DATA	TRAILER

The **HEADER** consists of <60> followed by <Command Length>

The **DATA** consists of Function ID, Function Length, and Function Data.

The **TRAILER** consists of <LRC> followed by <ETX>

All IC reader commands follow the Turbo TLP-224 protocol and use the ISO 7816-4 Application Protocol Data Unit structure to communicate with a IC card.

IC Microprocessor Card Reader Output Command Structure:
<60><XX XX><41><Command><LRC><ETX>

This command is used to pass an APDU or TPDU to a microprocessor card where only an ISO status is expected from the card.

IC Microprocessor Card Reader Input Command Structure:

<60><XX XX><61><Command><LRC><ETX>

This command is used to pass an APDU or TPDU to a microprocessor card where both data and an ISO status are expected from the card.

IC Reader Microprocessor Card T=1 Command Structure:
<60><XX XX><46><NAD><PCB><LEN><INF><EDC><LRC><ETX>

This command is used to pass a T=1 data block to a microprocessor card.

IC Memory Card Reader Command Structure:
<60><XX XX><42><Command or Response><LRC><ETX>

This command is used to perform an operation on a memory card.

ISO 7816-4 Command Structure:
<CLA><INS><P1><P2>[Lc] [Data] [Le]

ISO 7816-3 Command Structure:
<CLA><INS><P1><P2><P3>(Lc or Le)<Data>(if P3 is Lc)

Response Structure:
<Data>(Optional)<SW1><SW2>

Definitions:

<Data> Response data (usually Le bytes) if operation was successful
<SW1> Status byte 1
<SW2> Status byte 2

Definitions of Commands:

<CLA> ISO Class byte
<INS> ISO instruction code
<P1> ISO Parameter 1, usage varies with commands
<P2> ISO Parameter 2, usage varies with commands
<P3> ISO Parameter 3, length of data (Lc) or maximum length of expected reply (Le)
<Data> Data to send, varies with commands.
<Lc> The number of bytes present in the data field of the command APDU
<Le> The maximum number of bytes expected in the data field of the response APDU
<Data> Data to send (varies with command)
<NAD> ISO Node Address always 0 for T=1 protocol
<PCB> ISO Protocol Control Byte for T=1 protocol
<LEN> Length byte for T=1 protocol
<INF> Information fields for T=1 protocol
<EDC> Error detection code of the block (only LRC is supported)

COMMANDS

The following table is a summary of the smart card reader commands described in this section:

DATA *<XX>*	**NAME**	**USAGE**
6E	ICC Power On	To apply power to a microprocessor or memory card and return the ATR
4D	ICC Power Off	To turn the power to a microprocessor or memory card off
41	Output to a T=0 microprocessor card	Sends request (may include data) to microprocessor card
61	Input to a T=0 microprocessor card	Sends request to microprocessor card and waits for response data
42	Input/Output to a memory card	Used to perform an operation on a memory card
46	Send T=1 data block to microprocessor card	Sends T=1 data block to microprocessor card
42 DA 20	Verify Programmable Security Code	To verify the programmable security code for SLE4428, SLE4442, and AT88SC101 memory cards
42 DA B0	Read Binary	To read data from a memory card
42 DA C2	Read Binary with Protection Bit	To read data and its status from a SLE4428 memory card
42 DA B1	Read Protection Bit	To read protection bits from an SLE4442 memory card
42 DA B2	Read Security Memory	To read the security memory of an SLE4442 memory card
42 DA D0	Write Binary	To write data to a memory card
42 DA C1	Write Binary with Protection Bit	To write data to SLE4428 and SLE4442 memory cards and then prohibit further writing
42 DA D1	Write Security Memory	To write to the security memory of an SLE4442 memory card
42 DA 0E	Erase Command	To erase data at a specific address
42 DA DC	Decrease Counter by One	To decrease the counter on a GPM 271 and 276 card
42 DA D4	Restore Counter	To restore the counter on a or GPM 271 and 276 card
42 DA C4	C4 Control Command	To control the state of the C4 line and blowing fuses on a AT88SC101 card
42 DA C4 FE	Fuse Command	To blow a fuse on a AT88SC101 card

POWER-ON COMMAND

<60><00><Length><6E>[<Option>] [<PPS>]<LRC><03>

This command is used to power up the selected microprocessor card. It follows the ISO power-up sequence and returns the ATR as the response. Memory cards do not have an explicit power-up command, as the first I/O operation sent to the card causes a power-up.

<Option> and <PPS> are optional. A single-byte defines the Option as follows:

Bit Position	"0"	"1"
0	**No IFS**	Send S(IFS) request if T=1 protocol
1	**No Explicit PPS**	Explicit PPS
2-5	Unused	
6	**Auto PPS**	No auto PPS
7	**IFS response check**	No check on response of S(IFS) request

The response will be: <60><00><ATR Length><ATR><LRC><03>

Note: The response will vary, depending upon the card used. Specific information should be available directly from the card manufacturer.

POWER-OFF COMMAND

<60><00><01><4D><LRC><03>

This command is used to power down the selected microprocessor card. It works for any type of card.

Response is as follows: <60><00><02><90><00><F2><03>

OUTPUT TO A T=0 MICROPROCESSOR CARD COMMAND

Two-byte
| length |
<60><XX><XX><41><Command><LRC><03>

This command is used to send a request (that could include data) to a microprocessor card.

Response is as follows : <60><00><02><90><00><F2><03>

INPUT TO A T=0 MICROPROCESSOR CARD COMMAND

```
        Two-byte
      |  length  |
<60><XX><XX><61><Command><LRC><ETX>
```

This command is used to send a request to a microprocessor card and wait for a response.

INPUT/OUTPUT TO A MEMORY CARD COMMAND

```
        Two-byte
      |  length  |
<60><XX><XX><42><Command><LRC><ETX>
```

This command is used to perform an operation on a memory card.

SEND A T=1 DATA BLOCK TO A MICROPROCESSOR CARD COMMAND

```
        Two-byte
      |  length  |
<60><XX><XX><46><Command><LRC><ETX>
```

This command is used to send a data block to a microprocessor card.

There is no fixed response.

Memory Card Commands

Memory card commands pertain specifically to SLE4428, SLE4442, GPM276, GPM271, and AT88SC101 cards only. These commands consist of the memory command identifier (42h) and an ISO APDU. The reader interprets the command based on the selected memory card type and performs the requested operation. The reader returns ISO 7816 status (SW1, SW2) after finishing the operation.

VERIFY PROGRAMMABLE SECURITY CODE

Note: This command is for SLE4428, SLE4442, and AT88SC101 cards only.

This command is used to verify the Programmable Security Code (PSC). Each failed attempt writes one more bit of the error counter to zero. When all bits of the error counter are at zero, the PSC will no longer be accessible.

```
                xx  xx
<60><Command Length><42><DA><20><P1><P2>
<PSC Data Length><PSC><LRC><03>
          xx
```

For SLE4428 and SLE4442 cards, P1 and P2 are <00>. For the AT88SC101 card, P1 is the count of consecutive unsuccessful attempts allowed before further attempts are denied. P2 is the byte address of the password on the card. For the SLE4428 card, the PSC is two bytes long. For the SLE4442 card, the PSC is three bytes long.

<Command Length> is a two-byte counter indicating the length of data from <42> to the end of the PSC. (The most significant byte comes first.) Many operations on the card are blocked until a correct PSC has been presented. This command is used to verify the programmable security code. Each failed attempt writes one more bit of the Error Counter to zero. When all bits are zero, the PSC will no longer be accessible.

<PSC Data Length> is the one-byte length of following data.

Response is as follows:
<60><00><02><Return Status><LRC><03>
x x x x

READ BINARY

<60><00><06><42><DA><B0><Addr_h><Addr_l><Length><LRC><03>
xx

The Read Binary command is used to read data from the card, where:
<Addr_h><Addr_l> indicates the address from which the data should be read

<Addr_h> indicates the high byte of the two-byte address

<Addr_l> indicates the low byte of the two-byte address

<Length> indicates the length of data to be read from the card
xx

READ BINARY WITH PROTECTION BIT
Note: This command is for the SLE4428 card only.

<60><00><06><42><DA><C2><Addr_h><Addr_l><Length><LRC><03>

The Read Binary with Protection Bit command is used to read data and its status from the card, where:
<Addr_h><Addr_l> indicates the address from which the data should be read

<Addr_h> indicates the high byte of the two-byte address

<Addr_l> indicates the low byte of the two-byte address

<Length> indicates the one-byte length of data to be read from the card
xx

This command returns even bytes of data (less than <Length>). A two byte set is returned for each byte in the card. The first byte is card data, the second byte is the protection bit. “0” in the protection bit means “prohibit further writing.” “1” in the protection bit means “allow further writing.”

READ PROTECTION BIT

Note: This command is for the SLE4442 card only.

<60><00><06><42><DA><B1><00><00><04><4B><03>
The Read Protection Bit command is used to read the 32 protection bits. “0” in the protection bit means “prohibit further writing.” “1” in the protection bit means “allow further writing.”

READ SECURITY MEMORY

Note: This command is for the SLE4442 card only.

<60><00><06><42><DA><B2><00><00><04><48><03>

The Read Security Memory command is used to read four bytes of security memory. The PSC will be returned as <00> before successful verification of the PSC.

WRITE BINARY

xx xx
<60><Command Length><42><DA><D0><Addr_h><Addr_l>
<Data Length><Data><LRC><03>
xx

The Write Binary command is used to write data to the card where:
<Addr_h><Addr_l> indicates the address to which the data should be written

<Addr_h> indicates the high byte of the two-byte address

<Addr_l> indicates the low byte of the two-byte address

<Data Length> indicates the length of the data to be written to the card
xx

<Data> indicates the data to be written to the card

<Command Length> indicates the two-byte length of the command from <42> to <Data>

WRITE BINARY WITH PROTECTION BIT

Note: This command is for the SLE4428 and SLE4442 cards only.

xx xx
<60><Command Length><42><DA><C1><Addr_h><Addr_l>
<Data Length><Data><LRC><03>
xx

The Write Binary with Protection Bit command is used to write data to the card and prohibit further writing to the specified addresses, where:

<Addr_h><Addr_l> indicates the address to which the data should be written

<Addr_h> indicates the high byte of the two-byte address
<Addr_l> indicates the low byte of the two-byte address

<Data Length> indicates the one-byte length of the data to be written to the card
xx

<Data> indicates the data to be written to the card

<Command Length> indicates the two-byte length of the command from <42> to <Data>

WRITE SECURITY MEMORY

Note: This command is for the SLE4442 card only.

xx xx
<60><Command Length><42><DA><D1><00><Addr>
<Data Length><Data><LRC><ETX>
xx

The Write Security Memory command is used to write to security memory where:
<Addr> indicates the address to which the data should be written

<Data Length> indicates the one-byte length of the data to be written to the card

<Data> indicates security data to be written to the card

<Command Length> indicates the two-byte length of the command from <42> to <Data>

ERASE COMMAND

Note: This command is for the GPM271, GPM276, and AT88SC101 cards only.

xx xx
<60><00><06><42><DA><0E><00><Addr><Data Length><LRC><03>

The erase command is used to erase data at the specified address, where:
<Addr> indicates the address where the data is to be erased
<Data Length> indicates the length of the data to be erased

Note: A successful (<90><00>) return status does not necessarily mean the data was erased. Please attempt to read the data on the card after the erase operation has been completed to determine if the erase was actually accomplished.

DECREASE COUNTER BY ONE COMMAND

The Decrease Counter by One command is used to decrease the counter on GPM271 and GPM276 cards only.

<60><00><06><42><DA><DC><00><00><00><22><03>

RESTORE COUNTER COMMAND

The Restore Counter command is used to restore the counter on GPM271 and GPM276 cards only.

<60><00><06><42><DA><D4><00><00><00><2A><03>

C4 CONTROL COMMAND

The C4 Control command is used to control the state of the C4 line and blowing fuses on the AT88SC101 card only.

<60><00><06><42><DA><C4><Value><00><00><LRC><03>

<Value> indicates the C4 status. <01> will set C4 to high, and <00> will set C4 to low.

FUSE COMMAND

The Fuse command is used to blow a fuse on the AT88SC101 card only.

<60><00><06><42><DA><C4><FE><Byte_Addr><Bit_Addr><LRC><03>

<Byte_Addr> and <Bit_Addr> indicate the byte and bit address of the fuse to be blown.

Section 9
SECURITY ACCESS MODULES

A Security Access Module (SAM) is like a smart card in a smaller form factor. It provides authentication and/or encryption support for a particular type of card. SAMs fit into the five SIM connectors on an expansion board that attaches to the reader. If a reader is set to support only one type of card, then the reader will probably need only one SAM. If a reader is configured to accept a variety of cards, such as Visa, MasterCard, and Discover, it may need a SAM for each type of card it supports.

In typical operation, the host will ask the smart card for a random number, which will be given to the SAM. The SAM will then generate an encoded message containing a secret key which is presented back to the card. The card will generate a response to the SAM which will establish the card's validity to the host.

The Spectrum II can support up to six connectors (five SAM connectors and one smart card connector). When addressing the connectors, the main (smart card) landing connector is assigned the number '0' and each of the SAMs is assigned a number from 1 to 5.

Card Selection Command

This command allows the host application to select a card with which to read or write. Only one card can communicate with the application, either via the main connector or the SAM. The main connector is selected after powering up the reader.

<60><00><02><43><Card Connector><LRC><03>

<Card Connector> is a single byte defined as follows:

<00>	Main Connector
<01>	SAM Connector 1
<02>	SAM Connector 2
<03>	SAM Connector 3
<04>	SAM Connector 4
<05>	SAM Connector 5

The response will be: <60><00><02><90><00><LRC><03>

SET SAM CARD COMMAND

This command configures each SAM connector to support cards with certain defined characteristics.

<60><00><05><53><SAM Setting><02><Card Option><Memory Card Type><LRC><03>

A single-byte <SAM Setting> is defined as follows:
<61> SAM Connector 1
<62> SAM Connector 2
<63> SAM Connector 3
<64> SAM Connector 4
<65> SAM Connector 5

Single byte <Card Option> and <Memory Card Type> settings are the same as for the main connector.

The response will be: <60><00><02><90><00><LRC><03>

A Typical SAM Operation Scenario

1. Set CARD OPTION and MEMORY CARD TYPE for cards that will read and write on both the main connector and each SAM (if necessary).
2. Select main card.
3. Power main card.
4. Select SAM card.
5. Power on SAM card.
6. Select card (main or SAM).
7. Smart card (main or SAM) read/write operation.
8. Repeat 6 and 7 as necessary.
9. Select main card.
10. Power off main card.
11. Select SAM card.
12. Power of SAM card.

Notes: The GET READER STATUS command is available on the main connector only. Be sure to power off the other card when starting to read and write to a memory card. Powering on a SAM card will power on all SAMs, but only the selected SAM will communicate with the host application.

Section 10
MAGNETIC STRIPE READER COMMANDS

The Magnetic Stripe Reader has a number of configuration and data editing options. The most common use of the data editing feature is to extract only those specific data fields (such as Name, Account Number, EXP Date, Address, Age, etc.) required by the application.

Default settings, which are programmed into the reader at the factory, are printed in **boldface**.

Command Structures

Every command follows the same basic structure:

HEADER	DATA	TRAILER

The **HEADER** consists of <60> followed by <Command Length>
The **DATA** consists of Function ID, Function Length, and Function Data

The **TRAILER** consists of <LRC> followed by <ETX>

A simple Turbo TLP-224 protocol with one byte “check sum” is used when sending setup commands to reader. When sending a command:

```
                                Func
                                | ID |
<60><Command Length><53>[<xx><Len><FuncData>]<LRC><ETX>
          x x       x x                 x x
```

The response confirming the command structure will be:

```
                                Func
                                | ID  |
<60><Command Length><52>[<xx><LRC><ETX>
```

In this example:
<Command Length> is a two-byte counter from <53> to the end of <FuncData>.
<FuncID> is the total of contents, a respective command, and one byte that identifies the particular function affected.

<Len> is a one byte length count for the <FuncData> block.
<FuncData> is the data block for the function.
<ETX> = 03h

The overall <LRC> (Modulus 2 = Exclusive OR) checksum (from <60> to <LRC>) should be zero.
See page X for an example of the LRC calculation.

Communication Timing

During command processing or the reading of a magnetic stripe, the reader will not respond to a new command. Caution must be taken to maintain a minimum delay (250 ms) between two commands. The maximum delay for the reader to respond to a setting command is 1 second. The typical delay is less than 150 ms. The response delay for an IC card depends on the card.

Before issuing a new command, always wait for a response to the last command. Make sure the inter-command delay is more than 250 ms.

Command-Sending Protocol

System		MSR
Setting Command	¾ ¾ ¾ ¾ ®	
	¬ ¾ ¾ ¾ ¾	Status

Receiving Command Protocol

System		MSR
Receiving Command	¾ ¾ ¾ ¾ ®	
	¬ ¾ ¾ ¾ ¾	Setting and Status

Commands

The following table is a summary of the magnetic stripe reader commands described in this section:

HEAD *<60><Command Length>*	**DATA** *<XX><XX>*	**NAME**	**USAGE**
60 00 02	53 18	Restore to Default	To return the reader to its default settings
60 00 04	53 1A 01 XX	MSR Reading	To turn the magnetic stripe reading function (and buffer mode) on or off
60 00 04	53 1D 01 XX	Decoding Method	To read a card in a selected direction
60 00 04	53 19 01 XX	Send Option	To enable or disable the sentinel or account number on Track 2 only
60 00 04	53 21 01 XX	Terminator Setting	To format the data read from the card
60 00 04	53 3X 01 XX	Track 1, 2, 3 ID Setting	To edit the data read from the card
60 xx xx	53 DX XX	Preamble and Postamble Settings	To edit the data read from the card
60 00 04	53 13 01 XX	Track Selection Setting	To select the tracks on the magnetic stripe to be read or decoded
60 00 04	53 17 01 XX	Track Separator Setting	To edit the data read from the card

The command responses given below indicate successfully-entered commands. If an error message is returned instead of an indicated response, please refer to Appendix B: Status Code Table for an explanation.

RESTORE TO DEFAULT

<60><00><02><53><18><29><03>

This command does not have any <FuncData>. It restores all settings to the default value. data stored, etc.) It does not, however, restore Operation Mode settings to their default settings; these must be done individually. Please see Appendix A: Default Settings for a chart of all default settings.

The response will be:
<60><00><02><90><00><F2><03>

MSR READING

Turns the magnetic stripe reading function on or off. If the reading function is disabled, no data will be sent to the host.

<60><00><04><53><1A><01><MSR Reading Setting><LRC><03>

MSR Reading Setting:
<30> MSR Reading Disable
<31> MSR Reading Enable (Auto Transmit Mode)
<32> MSR Reading Buffered Mode

The response will be:
<60><00><02><90><00><F2><03>

DECODING METHOD

To ensure optimal reading performance, the reader is set at the factory to read magnetic stripe cards on withdrawal. Card velocity must be consistent to yield the best read rate, and a consistent speed of the card across the magnetic head is easier to achieve on withdrawal rather than insertion. However, the reader can be configured to read on both insertion and withdrawal or on insertion alone,ifdesired.

<60><00><04><53><1D><01><Decoding Method Setting><LRC><03>

Decoding Method Setting:
<31> Two Directions
<32> Read on insertion only
<33> Read on withdraw only

The response will be: <60><00><02><90><00><F2><03>

SEND OPTION

This setting allows the application program to disable or enable the start/end sentinel, and to disable or enable the Account Number for Track 2 only.

<60><00><04><53><19><01><Send Option Setting><LRC><03>

Send Option Setting:
<30> Do not send Start/End sentinel, but do send all data on all tracks
<31> Send Start/End sentinel and all data on all tracks
<32> Do not send Start/End sentinel for any track, but do send account number on Track 2 only
<33> Send Start/End sentinel on Track 1 only, and account number on Track 2 only for a credit card, or Send Start/End sentinel on Tracks 1 and 3 for a standard card

The response will be: <60><00><02><90><00><F2><03>

TERMINATOR SETTING

If the Data Edit feature is disabled, simple message formatting can be accomplished by using the Terminator, the Preamble, and the Postamble.

<60><00><04><53><21><01><Terminator Settings><LRC><ETX>

Terminator Settings:
<30> CR/LF
<31> CR
<32> LF
<33> None

The response will be: <60><00><02><90><00><F2><03>

TRACK 1 ID SETTING

<60><00><04><53><31><01><Track 1 ID><LRC><03>

<Track 1 ID> ASCII code set as Track 1 ID, NULL as none

The Track 1 ID can be any single ASCII character desired. No Track 1 ID is expressed as NULL.

The response will be: <60><00><02><90><00><F2><03>

TRACK 2 ID SETTING

<60><00><04><53><32><01><Track 2 ID><LRC><03>

The Track 2 ID can be any single ASCII character desired. No Track 2 ID is expressed as NULL.

The response will be: <60><00><02><90><00><F2><03>

TRACK 3 ID SETTING

<60><00><04><53><33><01><Track 3 ID><LRC><03>

The Track 3 ID can be any single ASCII character desired. No Track 3 ID is expressed as NULL.

The response will be: <60><00><02><90><00><F2><03>

PREAMBLE SETTING

<60><Command Length><53><D2><Len><Preamble String><LRC><03>

In this example:
<Command Length> is the length of the two-byte command from <53> to <Preamble String>

<Len> is the number of bytes of Preamble String

<Preamble String> is {string length}{string} (String length is one byte, maximum 9.)

The response will be: <60><00><02><90><00><F2><03>

POSTAMBLE SETTING

<60><Command Length><53><D3><Len><Postamble String><LRC><03>

In this example:
<Command Length> is the length of the two-byte command from <53> to <Postamble String>
<Len> is the number of bytes of Postamble String
<Postamble String> is {string length}{string} (String length is one byte, maximum 9.)

The response will be: <60><00><02><90><00><F2><03>

TRACK SELECTION SETTING

There are up to three tracks of encoded data on a magnetic stripe. This setting selects the tracks to be read and decoded.

<60><00><04><53><13><01><Track_Selection Settings><LRC><03>

Track Selection Settings:
<30> Any Track
<31> Track 1 Only
<32> Track 2 Only
<33> Track 1 & Track 2
<34> Track 3 Only
<35> Track 1 & Track 3
<36> Track 2 & Track 3
<37> All Three Tracks

Note: The default setting, "Any Track," permits the reader to read any track on which data is persent. The setting "All Three Tracks" will instruct the reader to read data from all three tracks. If any of the three tracks fails to read for any reason, no data will be sent from any of the tracks.

TRACK SEPARATOR SETTING

This setting allows the user to select the character to be used to separate data decoded by a multiple track reader.

<60><00><04><53><17><01><Track_Separator><LRC><03>

The Track Separator can be any one ASCII Character. The default value is **CR**.

ENVELOPE FOR MAGNETIC STRIPE DATA

This command adds the ID TECH envelope to magnetic stripe data before it is sent to the host.

<60> <Len_H> <Len_L> <card data indication 1> <card data indication 2> [Track 1 data][Track 2 data][Track 3 data] <LRC><03>

<card data indication 1> (<Cx>) is an ID to indicate magnetic data.

Bit Position	
0~3	Unused (set to 0)
4	'0'
5	'0'
6	'1'
7	'1'

<card data indication 2>

Bit Position	'0'	'1'
0	Track 1 decode fail	Track 1 decode success
1	Track 2 decode fail	Track 2 decode success
2	Track 3 decode fail	Track 3 decode success
3	No Track 1 data	Track 1 data exists
4	No Track 2 data	Track 2 data exists
5	No Track 3 data	Track 3 data exists
6~7	Unused (set to 0)	

Note: The Track x decode flag will be 0 if Track x data does not exist.

The order of magnetic data and switch change notification depends on the order they come to the microcontroller. This is not fixed.

ARM TO READ IN BUFFER MODE

This command sets the reader to read magnetic stripe data and store it in memory.

<60><00><03><50><01><30><02><03>

The response will be: <60><00><02><90><00><F2><03>

If the reader controls the LED, the LED will turn green and the reader will send an ACK response to the host. Previously-read data will be erased, and the reader will wait for the next card insertion.

When a card is inserted and withdrawn, the decoded data will be saved in memory and not sent to the host. If the reader controls the LED, the LED will go off. (If there was no data to read, the LED will briefly turn red and then go off.) A notification will be sent to the host to indicate the presence of magnetic data. Data will be held until receiving the next Arm To Read Command or MSR Reset Command.

While in Buffer Mode, the reader will continue to allow non-ICC commands (e.g. unlatch, status, LED commands). If the reader receives an ICC command, it will respond but the magnetic data in memory will be erased.

MSR RESET IN BUFFER MODE

This command resets the reader to its MSR default settings when Buffer Mode is enabled.

<60><00><03><50><01><32><00><03>

The response will be: <60><00><02><90><00><F2><03>

If the reader is configured to automatically transmit magnetic data, the reader will respond that the command is not supported. Any stored magnetic data will be erased. The reader will send an ACK response to the host.

READ MSR DATA IN BUFFER MODE

There are up to three tracks of encoded data on a magnetic stripe. This setting selects the tracks to be read and decoded in Buffer Mode.

Command:
<60><00><03><51><01><Track Select Byte><LRC><03>

Track Selection Settings:
<30> Any Track
<31> Track 1
<32> Track 2
<33> Track 1 & Track 2
<34> Track 3
<35> Track 1 & Track 3
<36> Track 2 & Track 3
<37> All Three Tracks

The data on the selected track(s) will be sent to the host either in envelope format or not, according to the Card Notification Setting, or in RAW format. The data will not be erased after this command.

Section 11
COMMAND EXAMPLES

In the demonstration software there are examples of specific commands to the reader and the correct responses to those commands. For convenience, they are grouped into two sample command routines:

Sample T=0 Command to card

Use the ID TECH-provided sample card labeled @. Start the IDTECH PC/SC Sample Demo Program and insert the @ card in the reader. In the command window type the following commands:

1. Select File 30 40.
   Command: 60 00 08 **41 00 A4 00 00 02 30 40** LRC 03
   Response: 60 00 02 90 00 F2 03

2. Verify CHV (sample password).
   Command: 60 00 14 **41 00 20 00 80 0E 73 61 6D 70 6C 65 70 61 73 73 77 6F 72 64** LRC 03
   Response: 60 00 02 90 00 F2 03

3. Write 8 bytes 12345678 to file 3040.
   Command: 60 00 0E **41 00 D6 00 00 08 31 32 33 34 35 36 37 38** C1 03
   Response: 60 00 02 90 00 F2 03

4. Read 8 bytes.
   Command: 60 00 06 **61 00 B0 00 00 08** D8 03
   Response: 60 00 0A 31 32 33 34 35 36 37 38 90 00 F2 03

5. Write 8 bytes ABCDEFGH.
   Command: 60 00 0E **41 00 D6 00 00 08 41 42 43 44 45 46 47 48** F9 03
   Response: 60 00 02 90 00 F2 03

6. Read 8 bytes.
   Command: 60 00 06 **61 00 B0 00 00 08** BF 03
   Response: 60 00 0A 31 32 33 34 35 36 37 38 90 00 F2 03

Sample T=1 Read and Write Command to Card

This sample command uses the IBM PC/SC T=1 Test Card

1. Power on and request ATR.

2. Negotiate IFS (propose maximum block size of 254 bytes which card accepts). Must immediately follow the ATR.
   Command: 60 00 06 46 **00 C1 01 FE 3E** LRC 03
   Response: 60 00 05 00 E1 01 FE 1E 04

3. Select file 0007.
   Command: 60 00 0C 46 **00 00 07 A4 A4 00 00 02 00 07 02** LRC 03
   Response: 60 00 14 00 00 10 63 0C 03 E8 00 07 00 00 00 FF FF 11 01 00 90 00 13 LRC 03

4. Write 1 byte to file at address 0000.
   Command: 60 00 0B 46 **00 40 06 A4 D6 00 00 01 00 35** LRC 03
   Response: 60 00 06 00 40 02 90 00 D2 LRC 03

5. Read 1 byte from file starting at address 0000.
   Command: 60 00 0A 46 **00 00 05 A4 B0 00 00 01 10** LRC 03
   Response: 60 00 07 00 00 03 00 90 00 93 LRC 03

6. Write 25 bytes to address 0000.
   Command: 60 00 23 46 **00 40 1E A4 D6 00 00 19 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 2D** LRC 03
   Response: 60 00 06 00 40 02 90 00 D2 LRC 03

7. Read 25 bytes starting at address 0000.
   Command: 60 00 0A 46 **00 05 A4 B0 00 00 19 08** LRC 03
   Response: 60 00 1F 00 00 1B 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 90 00 93 LRC 03

Section 12
OPERATION

The ID TECH Hybrid Insert Reader is easy to operate. Make sure the reader is properly connected and receiving sufficient power. The green LED will indicate that it is ready to read.

LED INDICATION	**MEANING (LED controlled by reader)**
Amber	Reader is sending or receivbing data from the host.
Green	Reader is ready to power on a smart card, read a magnetic stripe, or is idle.
Red	Bad magnetic stripe read.
Off	Reader is reading magnetic stripe data, or a smart card has been powered on (but is not communicating with the host).

By default, the LED is under the control of the reader. The LED can also be under the control of the host application. (Please see page 11 for the specific HOST LED CONTROL commands.) If the LED is under the control of the host, the following settings are available:

· Turn the LED off
· Turn the LED green
· Turn the LED red
· Turn the LED amber

To read a **Magnetic Stripe Card**, just follow these simple steps:

1. Insert the card, magnetic stripe down, into the reader until it hits a hard stop. As soon as the magnetic stripe is detected by the reader, the green LED indicator will go off.

2. When the card has been inserted all the way, the green LED will light again.

3. Withdraw the card in one continuous motion. The green LED will go off again.
(The reader is capable of reading a magnetic stripe on both insertion and withdrawal, but a more reliable read is achieved on withdrawal.)

4. When the card has been fully withdrawn, the LED will turn amber to indicate processing.

5. The LED will turn red (to indicate a bad read) or green (to indicate a good read).

To read a Smart Card, follow these steps:

1. Insert the card, chip up, into the reader until it stops. Then push the card in another 1/8 inch until it clicks in place. The click indicates that the card is properly seated. When the card is properly seated and powered, the green LED indicator will go off.

2. If the “latch” option is present, and the LATCH ON command is sent by the host to the reader, the latch will engage, preventing the withdrawal of the card prior to the completion of processing.

3. The LED will turn amber to indicate processing. It will go off when processing is completed.

4. The LED will turn green again when power to the card is turned off. The latch, if present, will disengage when the LATCH OFF command is sent by the host to the reader. Withdraw the card.

For a list of ID TECH-compatible cards, please see Appendix C or visit our website at www.idt-net.com.

Note: There are two notifications which may be sent to the host by the reader during operation. One is a “Unit Error Notification” (in the event of overheating or unstable VCC power, of if the card is removed during processing). The other is the optional “Card Status Change” notification.

Buffer Mode

When the unit is armed to read in the buffer mode, decoded data will be retained in memory and an optional notice will be sent to the host to indicate its presence. Data will be held in memory until the reader receives the next ARM TO READ or MSR RESET command, at which point all data in memory will be erased. Please see pages 38 and 39 for the specific ARM TO READ IN BUFFER MODE, MSR RESET IN BUFFER MODE, and READ MSR DATA IN BUFFER MODE commands.

"CARD LATCH" Option

The optional “latch” mechanism holds an inserted card in place in the reader’s slot until processing has been completed.

The mechanism operates using LATCH ON and LATCH OFF commands issued by the host. To ensure full latching and unlatching, the host terminal should wait a minimum of 500ms between latching and unlatching commands to allow the motor control circuits to properly reset after each motor operation. The host can request the status of the latch mechanism by issuing a GET READER STATUS command. Please see pages 16 and 17 for the specific LATCH ON, LATCH OFF, and GET READER STATUS commands.

An optional feature available with the latch mechanism is “power failure latch release,” which automatically unlatches the card when power fails. The host terminal’s +5V power supply must not decay at a rate greater than 50V/s for this feature to perform reliably, because the power for driving the latch motor in the event of a power failure is derived from the host terminal’s power supply filter capacitor. (The latch circuit gets its power from the power supply’s decaying voltage for a period of 20ms minimum to ensure a reliable latch release.) In the event the host terminal’s power supply filter capacitor is too small for reliable unlatching, an external capacitor with a voltage rating of at least 10V and a capacitance of 2200 microfarads must be used. (The capacitor’s positive lead must be connected to Pin 1.) If required, a capacitor can be installed on the reader at the factory.

Plug-and Play Support

The reader will notify the operating system of its arrival by sending a “Device ID String” at 1200 baud (1200, 7, 1, Mark), after power-up or reconnection to the terminal. However, the reader will only send this string if requested to do so by the operating system. The operating system will request the Plug-and-Play string upon booting and when scanning for hardware changes.

Plug-and-Play Device ID String:

(<01 22>ITS8865\[Serial Number]\\\ID TECH 8865 Hybrid Mag&EMV Card Reader [Check Sum byte 1][Check Sum byte 2])

[Serial Number] is an 8-byte string, represents serial number of a device.

[Check Sum byte 1] and [Checks Sum byte 2] is a single byte check character.

Section 13
TROULBLESHOOTING

The Spectrum II Hybrid Insert Reader is easy to install and use. Most problems encountered can be attributed to:

· Incorrect Interface Cabling
· Incorrect Configuration Setup
· Bad Magnetic Stripe Quality

General Procedures

The troubleshooting process can be simplified by following these simple diagnostic procedures.

1. Once it has been confirmed that the unit is correctly powered, try inserting a credit card. The LED will turn amber while processing, then either green or red, as appropriate.

2. Once the unit has indicated a “good read,” then proceed to check the interface cabling connections.

With PC/SC

Many problems in the PC/SC environment can be attributed to:

· Operating system is not Windows2000 or XP (32 bits)
· Serial port is not set to Plug-and-Play
· Wrong card
· Wrong card orientation
· Card is not encoded

Questions to Ask

1. Is the desired track on the magnetic stripe enabled?

2. Does the output data format match the requirements of the application software?

3. Has the magnetic stripe been encoded in a standard format?

4. Is the smart card a valid EMV smart card? If not, change the default mode from EMV cards to ISO cards by configuring for ISO 7816.

Appendix A
DEFAULT SETTINGS

Default Settings for Magentic Stripes

The ID TECH Hybrid Insert Reader is shipped from the factory with the following default settings already programmed:

Magnetic Track Basic Data Format
Track 1: <SS1><T_1 Data><ES><CR>*
Track 2: <SS2><T_2 Data><ES><CR>*
Track 3: <SS3><T_3 Data><ES><CR>*

where:
SS1(start sentinel track 1) = %
SS2(start sentinel track 2) = ;
SS3(start sentinel track 3) = ; for ISO, ! for CDL, % for AAMVA
ES(end sentinel all tracks) = ?
CR = Carriage Return

DEFINITIONS FOR MAGNETIC STRIPES

Start or End Sentinel: Characters in encoding format which come before the first data character (start) and after the last data character (end), indicating the beginning and end, respectively, of data. The Start Sentinel always begins with a “1” (one) bit to signal the start of data.

Track Separator: A designated character which separates data tracks.

Terminator: A designated character which comes at the end of the last track of data, to separate card reads.

LRC: Check character, following end sentinel.

CDL: Old California Drivers License format.

** Note: The <CR> commands shown above for Tracks 1 & 2 and Tracks 2 & 3 denote the default character for this position, the Track Separator position. The <CR> command shown for Track 3 denotes the default character for this position, the Terminator position.*

Default Settings Table

Setting	Default
MSR Reading	Auto Mode
Magnetic Decoding Method	Withdraw Only
Send Option	Send Start/End Sentinel
Terminator Setting	CR (Hex OD)
Preamble Setting	None
Postamble Setting	None
Track Selected Setting	Any Track
Track Separator Setting	CR (Hex OD)
Baud Rate Setting	38400 bps
Track 1 ID	None
Track 2 ID	None
Track 3 ID	None
IC Card Type	5V Microprocessor Card
Memory IC Card Type	3 Byte I²C Memory Card

Appendix B
STATUS CODE TABLE

Return Status and Explanation

Code	Definition
<C0> XX	Magnetic Card Data with Envelope
<B0> XX	Card status change notification
<90><00>	Operation completed successfully (All Operations)
<8C><00>	TCK error
<8B><00>	Unsupported TAx, TBx, TCx, TDx
<89><00>	ATR too long
<88><00>	Power not ready for T=0 microprocessor card
<87><00>	Protocol not supported by the reader
<86><00>	Unsupported Fi or Di in PPS
<85><00>	PPS confirmation error
<84><00>	Parity error in reception
<83><00>	Parity error in transmission
<82><00>	Unknown TS
<81><00>	Time out
<6E><00>	CLA not supported
<6D><00>	INS not supported
<69><00>	Command not supported
<67><00>	Warning, value read is different from value written
<66><88>	Invalid PSC presented
<66><87>	No more retries
<66><86>	No more counter to decrease
<2F><00>	Fault alarm received
<2D><00>	Memory card not supported
<2C><00>	Card not present
<2B><00>	Address not supported
<2A><00>	Command received correctly, but could not be completed

A fault alarm will be reported if a smart card is removed with the power still on, or a supply voltage drop, short circuit, or overheating has been detected.

Appendix C
COMPATIBLE MEMORY CARDS

ID TECH - Compatible Memory Cards

Card	Spec.	Typical Applications	Note (Card Type)
GemPlus GFM 4K	No	Small records storage, Loyalty, Conventions, Digital receipts	I^2C Memory Card (0)
Schlumberger PrimeFlex Open 2/4K	Yes	Small records storage, Loyalty, Conventions, Digital receipts	I^2C Memory Card (0)
CardLogix CLXSA001KA1	Yes	Small records storage, Loyalty, Conventions, Digital receipts	128 bytes I^2C MemoryCard (0)
CardLogix CLXSA002KA2	No	Small records storage, Loyalty, Conventions, Digital receipts	256 bytes I^2C Memory Card (0)
CardLogix CLXSA004KA6	Yes	Small records storage, Loyalty, Conventions, Digital receipts	512 bytes I^2C Memory Card (0)
CardLogix CLXSA008KA7	Yes	Small records storage, Loyalty, Conventions, Digital receipts	1K bytes I^2C Memory Card (0)
CardLogix CLXSA016KA8	Yes	Records storage, Health informatics, Loyalty, Conventions, Digital receipts	2K bytes Memory Card (0)
CardLogix CLXSA032KA9	Yes	Records storage, Health informatics, Loyalty, Conventions, Digital receipts	4K bytes I^2C Memory Card (2)
CardLogix CLXSA064KA3	Yes	Records storage, Health informatics, Loyalty, Conventions, Digital receipts	8K bytes I^2C Memory Card (1)
CardLogix CLXSA128KA4	Yes	Records storage, Health informatics, Loyalty, Conventions, Digital receipts	16K bytes I^2C Memory Card (1)
CardLogix CLXSA256KA5	Yes	Records storage, Health informatics, Loyalty, Conventions, Digital receipts	32K bytes I^2C Memory Card (1)
Gemplus GPM2K	Yes	Small records storage, Loyalty, Conventions, Digital receipts	256 bytes (SLE4442) 7816 Synchronous (Secured Memory Card) (2)
Schlumberger PrimeFlex Store 2K	Yes	Small records storage, Loyalty, Conventions, Digital receipts	256 bytes (SLE4442) 7816 Synchronous (Secured Memory Card) (2)
Gemplus GPM8K	Yes	Small records storage, Loyalty, Conventions, Digital receipts	1K bytes (SLE4428) 7816 Synchronous (Secured Memory Card) (3)
CardLogix CLXSA008KB3	Yes	Small records storage, Loyalty, Conventions, Digital receipts	1K bytes (SLE4428) 7816 Synchronous (Secured Memory Card) (3)
CardLogix CLXSA008KB4	Yes	Small records storage, Loyalty, Conventions, Digital receipts	1K bytes (SLE4428) 7816 Synchronous (Secured Memory Card) (3)
Gemplus GPM271	Yes	Stored value, Laundromats, Telephones, Prepaid systems, Tokens	7816 Synchronous (Token Card) (5)
Gemplus GPM276	Yes	Stored value, Laundromats, Telephones, Prepaid Systems, Tokens	7816 Synchronous (Token Card) (4)

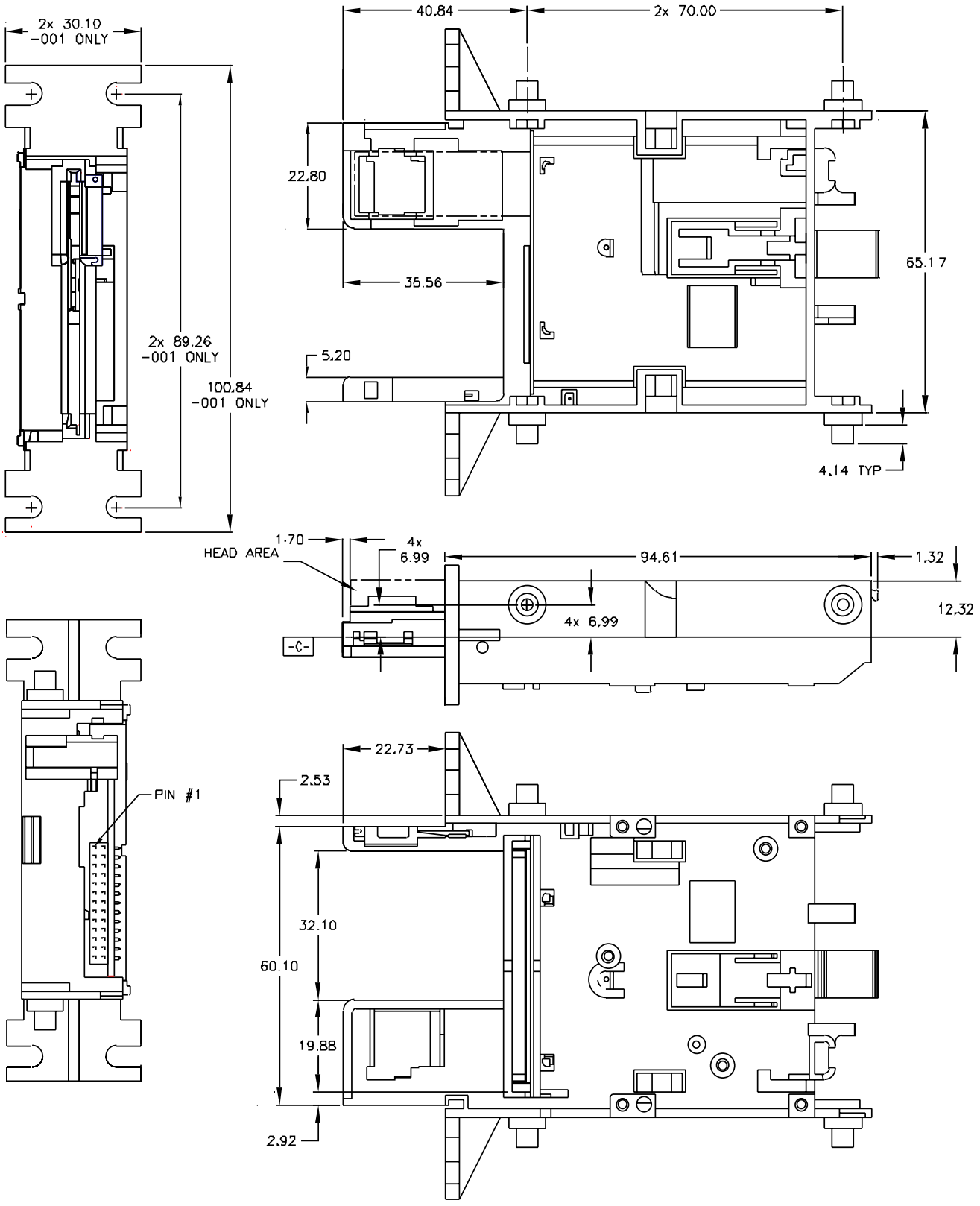
Note: For the latest additions to this list, please visit our website at www.idt-net.com.

Appendix D
FUNCTION ID TABLE

Function Description	Function ID	Function Value
MSR Reading	1A	0 – 2
Magnetic Decoding Method	1D	1 – 3
Send Option	19	0 – 3
Terminator Settings	21	CR/LF, CR, LF, None
Preamble Setting	D2	String
Postamble Setting	D3	String
Track Selected Settings	13	0 – 7
Track Separator Settings	17	Any ASCII Code
Track 1 ID	31	Any ASCII Code
Track 2 ID	32	Any ASCII Code
Track 3 ID	33	Any ASCII Code
IC Card Type	10	Any Value
Memory IC Card Type	12	0 – 6
Set Operation Mode	11	Any Value
Data Editing	1B	Not Used
SAM 1 Setting	61	Any Value
SAM 2 Setting	62	Any Value
SAM 3 Setting	63	Any Value
SAM 4 Setting	64	Any Value
SAM 5 Setting	65	Any Value

Appendix E

OUTLINE DRAWING OF READER



Appendix F
CONNECTOR

Connector Pin-Outs

PIN DESIGNATIONS FOR THE 8-PIN MOLEX HEADER

POSITION	SIGNAL
PIN 1	CASE GND
PIN 2	TXD
PIN 3	RXD
PIN 4	VCC/DTR
PIN 5	RTS
PIN 6	CTS
PIN 7	GND
PIN 8	DSR

8-pin Molex header 53015-0810 accepts Molex 51004-0800 housing.

PIN DESIGNATIONS FOR DB9 FEMALE CONNECTOR

POSITION	SIGNAL
PIN 2	TXD
PIN 3	RXD
PIN 4	DSR
PIN 5	GND
PIN 6	DTR
PIN 7	CTS
PIN 8	RTS

POWER ADAPTER

5V, 500 ma
Polarity: Inside (+)

Warning: Using any power supply (VCC) higher than 5.5 may damage the reader.

Appendix G
SPECIAL INSTALLATION INSTRUCTIONS

Installing an Unsigned Driver Over Signed Drivers

An unsigned driver should only be installed when it has been approved by ID TECH. To install a driver that has not been signed, the computer must be configured to allow unsigned drivers to be installed. Do this in the control panel, and answer “yes” to the prompt: “This driver is not signed do you wish to continue anyway?” Be attentive to the installation process, however, because Windows will prefer a signed driver over an unsigned one, and if a signed driver is already in the system, Windows will likely try to install that driver over the unsigned one.

1. Unzip the driver in an appropriate directory and remember its location.

2. Go to SETTINGS/CONTROL PANEL/SYSTEM/HARDWARE/DEVICE MANAGER.

3. Select SMART CARD READERS.

4. Right click on ID TECH Spectrum II Magstripe and EMV Card Reader.

5. Select PROPERTIES/DRIVER/UPDATE DRIVER/NEXT.

6. Select DISPLAY LIST OF KNOWN DRIVERS... /NEXT.
7. Select HAVE DISK/BROWSE.

8. Browse to and select the unzipped driver’s location.

9. Select OK/NEXT/NEXT and YES to the prompt that the digital signature was not found.

10. Select FINISH/CLOSE.

Configuring to Display System Files

Because its0020.sys is a system file, it will not be visible by default with Windows Explorer. To display system files in a Windows 2000 or Windows XT system, go to EXPLORER/TOOLS/ FOLDER OPTIONS... /VIEW and uncheck the boxes HIDE FILE EXTENSIONS OF KNOWN TYPES and HIDE PROTECTED OPERATING SYSTEM FILES.

Configuring the Smart Card Service

If your application reports that it cannot find the smart card reader in Windows 2000, go to SETTINGS/CONTROL PANEL/ADMINISTRATIVE TOOLS/SERVICES and check to see if the status on the SERVICE SMART CARD is started and the STARTUP TYPE is set to ACTIVE. (If in Windows XP, go to START/CONTROL PANEL/ADMINISTRATIVE TOOLS/SERVICES.)

If not, right click on the line that says SMART CARD and under the general tab go to STARTUP TYPE and change the setting to AUTOMATIC. Then under SERVICE STATUS select START.

Verify that the application can find the reader.

Reconnectiong a Disconnected Reader

If the reader has been disconnected or powered-down while operating, it will stop functioning. To correct this situation, reconnect the reader and/or power the reader and then either reboot the PC or do the following:

In Windows 2000, go to SETTINGS/CONTROL PANEL/SYSTEM/HARDWARE/DEVICE MANAGER/SMART CARD READERS. Select the ID TECH Spectrum II Magstripe and EMV Card Reader entry. Select DISABLE, then select ENABLE.

In Windows XP, go to START/CONTROL PANEL/SYSTEM/HARDWARE/DEVICE MANAGER/ SMART CARD READERS. Select the ID TECH Spectrum II Magstripe and EMV Card Reader entry. Select DISABLE, then select ENABLE.

Appendix H
ID TECH PC/SC IDTLIB.DLL GUIDE

Purpose of this library is to simplify access to the reader when using vendor- specific functions like magnetic stripe reading or latch control. This library is not required: all functions can be done by the application without requiring this DLL. However, this DLL is provided to simplify the programming of an application to fully utilize the reader's functions.

The error return codes are either SCARD_S_SUCCESS for success, or "–1" if the function failed entirely. For the standard smart card error codes, see "Smart Card Error Codes" on page XX.

Function prototypes are for the callable functions in the IDTLIB.DLL for used for handling vendor-specific IOCTL calls to the Spectrum reader in a Windows PC/SC environment.

Note: hCard is the card handle returned by Windows from SCardConnect().

IDTLIB_API LONG IdtGetVersion (OUT BYTE *versionBuffer, IN OUT DWORD *versionBufferLength);
BYTE *versionBuffer is a pointer to an unsigned character buffer to hold the DLL version string. This buffer must be at least 7 bytes long.
DWORD versionBufferLength is an unsigned long integer (32 bits) that holds the length of the buffer in bytes to hold the version string when called and on return to hold the actual size in bytes of the version string.

IDTLIB_API LONG IdtCardOff (IN ULONG hCard);
ULONG hCard is unsigned long (32 bits) card handle.

This API is used to send a "Power Off" command to the reader.

LONG IDTLIB_API IdtCardSelect (IN ULONG hCard, IN BYTE cardSelect);
ULONG hCard is unsigned long integer (32 bits) card handle.
BYTE cardSelect is a byte to select card module.

This API is used to select a card module to operate with.

Card selections are defined as follows :

```
#define    CARD_SEL_MAIN    0
#define    CARD_SEL_SAM1    1
#define    CARD_SEL_SAM2    2
#define    CARD_SEL_SAM3    3
#define    CARD_SEL_SAM4    4
#define    CARD_SEL_SAM5    5
```

IDTLIB_API LONG IdtGetMagstripeTimeout (IN ULONG hCard, OUT UCHAR *timeout);
ULONG hCard is unsigned long integer (32 bits) card handle
UCHAR timeout is an unsigned character (8 bits) returning the timeout value in seconds.

This API is used to get the timeout (in Second) for magnetic stripe data.

IDTLIB_API LONG IdtGetDriverVersion (IN ULONG hCard, OUT UCHAR *drvrVersion);
ULONG hCard is unsigned long integer (32 bits) card handle.

This API is used to get version of driver.

IDTLIB_API LONG IdtGetFirmwareVersion (IN ULONG hCard, OUT UCHAR *firmwareVersion);
ULONG hCard is unsigned long integer (32 bits) card handle
UCHAR *firmwareVersion a pointer to an unsigned character buffer to receive the firmware version.

This API is used to get version of firmware on the reader.

LONG IDTLIB_API IdtGetReaderState(IN ULONG hCard, OUT UCHAR *readerState);

ULONG hCard is unsigned long integer (32 bits) card handle
UCHAR *readerState is a pointer to an unsigned character buffer to receive the reader status.

This API is used to get the current state of the reader and readerState is defined as follows:
bit 0: power, 0 => not ready, 1 => ready
bit 1: card, 0 => not seated, 1 => seated
bit 2: latch, 0 => released, 1 => closed
bit 1: card, 0 => absent, 1 => present

IDTLIB_API LONG IdtGetVersion (OUT BYTE *versionBuffer, IN OUT DWORD *versionBufferLength);
BYTE *versionBuffer is a pointer to an unsigned character buffer to hold the DLL version string. This buffer must be at least seven bytes long.
DWORD versionBufferLength is an unsigned long integer (32 bits) that holds the length of the buffer in bytes to hold the version string when called and on return to hold the actual size in bytes of the version string.

This API is used to get the version of DLL.

LONG IDTLIB_API IdtPollMagstripe (IN ULONG hCard, IN BYTE TrackSelect, // track selection OUT BYTE *outputBuffer, // OUT DWORD *outputBufferLength);
ULONG hCard is unsigned long integer (32 bits) card handle.
BYTE TrackSelect is a byte to select tracks of magnetic stripe.
BYTE outputBuffer is a pointer to a buffer containing the magnetic data.
DWORD outputBufferLength is a pointer to an unsigned long integer that holds the length of magnetic data.

This API is used to poll the reader after the user has issued the IdtStartMagstripe() call. If the MSR data is available, it will be returned according the tracks selected.

Return code:
—SCARD_S_SUCCESS: MSR data available.
—SCARD_E_NOT_READY: MSR data not available yet.
—SCARD_E_TIMEOUT: Attempt to read has timed out.

Track selections are defined as follows (note that they are in ASCII):

#define	MSR_RD_ANY_TRK	'0'
#define	MSR_RD_TRK_1	'1'
#define	MSR_RD_TRK_2	'2'
#define	MSR_RD_TRK_12	'3'
#define	MSR_RD_TRK_3	'4'
#define	MSR_RD_TRK_13	'5'
#define	MSR_RD_TRK_23	'6'
#define	MSR_RD_ALL_TRK	'7'

LONG IDTLIB_API IdtReadMagstripe (IN ULONG hCard, OUT BYTE *outputBuffer, // OUT DWORD *outputBufferLength IN BOOLEAN TurnCardOff);
ULONG hCard is unsigned long integer (32 bits) card handle.
BYTE outputBuffer is a pointer to a buffer containing the magnetic data.
DWORD outputBufferLength is a pointer to an unsigned long integer that holds the length of magnetic data.
BOOLEAN TurnCardOff is a Boolean to indicate whether a "Power Off " command should be send before getting magnetic card data.

LONG IDTLIB_API IdtResetMSRBuffer (IN ULONG hCard);
ULONG hCard is unsigned long integer (32 bits) card handle

This API is used to reset the MSR buffer (clear the buffer) and put the reader into disarmed state.

IDTLIB_API LONG IdtSendCommand (IN ULONG hCard, IN BYTE *commandBuffer, IN DWORD commandLength, OUT BYTE *responseBuffer, IN OUT DWORD *responseBufferLength);

ULONG hCard is unsigned long integer (32 bits) card handle
BYTE *commandBuffer is a pointer to a buffer containing the command to send to the reader.
DWORD commandLength is an unsigned long integer (32 bits) that holds the length of the command in the commandBuffer
BYTE *responseBuffer is a pointer to a byte buffer into which the response from the reader will be placed.
DWORD *responseLength is an unsigned long integer (32 bits) that holds the length of the response in bytes in the responseBuffer.

This API is used to send a generic command to the reader. The user is advised NOT to use this API liberally as this API will change the reader state without driver registering the changes.

LONG IDTLIB_API IdtSetLEDColor (IN ULONG hCard, IN BYTE LEDColor);

ULONG hCard is unsigned long integer (32 bits) card handle
BYTE LEDColor is a color reader will set to.

This API is used to set the LED color, provided the reader is set into the host controlled mode.

The LED colors are defined as (note that they are in ASCII):

```
#define     LED_OFF     '0'
#define     LED_GREEN   '1'
#define     LED_RED     '2'
#define     LED_AMBER   '3'
```

LONG IDTLIB_API IdtSetMSRReadMode (IN ULONG hCard, IN BYTE readMode);

ULONG hCard is unsigned long integer (32 bits) card handle
BYTE readMode is the MSR mode reader will set to.

This API is used to set the MSR read mode and read modes are defined as (note that they are in ASCII):

```
#define     MSR_MODE_DISABLE    '0'
#define     MSR_MODE_AUTO_TX    '1'     // auto-tranismit mode
#define     MSR_MODE_BUFFER     '2'     // buffered mode
```

IDTLIB_API LONG IdtSetMagstripeTimeout (IN ULONG hCard, IN UCHAR timeout);

ULONG hCard is unsigned long integer (32 bits) card handle
UCHAR timeout is an unsigned character holding the new timeout value in seconds.

This API is used to set the timeout (in second) for waiting for magnetic stripe data.

LONG IDTLIB_API IdtStartMagstripe (IN ULONG hCard, IN DWORD outputBufferLength, IN BOOLEAN TurnCardOff);

ULONG hCard is unsigned long integer (32 bits) card handle.
DWORD outputBufferLength is a pointer to an unsigned long integer that holds the length of magnetic data.
BOOLEAN TurnCardOff is a Boolean to indicate whether a “Power Off ” command should be send before getting magnetic card data.

This API is used to initiate the reading of MSR data. The user is expected to follow up this call by polling the reader for MSR data using IdtPollMagstripe().

IDTLIB_API LONG IdtUnlatch (IN ULONG hCard);
ULONG hCard is unsigned long integer (32 bits) card handle.

This API is used to send the “Latch Off” command to the reader.

WRITE YOUR OWN FUNCTIONS

If you wish to write your own functions and eliminate the use of the DLL, all these commands use the ScardControl() to communicate with the driver.

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/Security/scardcontrol.asp

LONG SCardControl(
SCARDHANDLE *hCard*
DWORD *dwControlCode*
LPCVOID *lpInBuffer*
DWORD *nInBufferSize*
LPVOID *lpOutBuffer*
DWORD *nOutBufferSize*
LPDWORD *lpBytesReturned*

Parameters
hCard [in] Reference value returned from SCardConnect.
dwControlCode [in] Control code for the operation. This value identifies the specific operation to be performed.
lpInBuffer [in] Pointer to a buffer that contains the data required to perform the operation. This parameter can be NULL if the *dwControlCode* parameter specifies an operation that does not require input data.
nInBufferSize [in] Size, in bytes, of the buffer pointed to by *lpInBuffer*.
lpOutBuffer [out] Pointer to a buffer that receives the operation’s output data. This parameter can be NULL if the *dwControlCode* parameter specifies an operation that does not produce output data.
nOutBufferSize [in] Size, in bytes, of the buffer pointed to by *lpOutBuffer*.
lpBytesReturned [out] Pointer to a DWORD that receives the size, in bytes, of the data stored into the buffer pointed to by *lpOutBuffer*.

Return Values
This function returns different values depending on whether it succeeds or fails.
Success: SCARD_S_SUCCESS.
Failure: An error code (see “Smart Card Error Codes”).

Example:

SMARTCARD_MACRO	SCARD_CTL_CODE
SMARTCARD_VS_GENERIC_COMMAND	2100
SMARTCARD_VS_LOCK_LATCH	2101
SMARTCARD_VS_UNLOCK_LATCH	2102
SMARTCARD_VS_GET_MAGSTRIPE	2103
SMARTCARD_VS_SET_MAGSTRIPE_TIMEOUT_COMMAND	2104
SMARTCARD_VS_GET_MAGSTRIPE_TIMEOUT_COMMAND	2105
SMARTCARD_VS_GET_DRIVER_VERSION_COMMAND	2106
SMARTCARD_VS_GET_FIRMWARE_VERSION_COMMAND	2107

```
#define SMARTCARD_VS_GET_MAGSTRIPE SCARD_CTL_CODE (2103)
// wait for the card to be removed and read the magstripe
LONG IDTLIB_API IdtReadMagstripe(IN ULONG hCard, OUT BYTE *outputBuffer,
IN OUT DWORD *outputBufferLength,
IN BOOLEAN TurnCardOff) {DWORD bytesBack; LONG ret;

if(TurnCardOff) IdtCardOff (hCard);
ret = SCardControl(hCard, SMARTCARD_VS_GET_MAGSTRIPE, NULL, 0, outputBuffer,
*outputBufferLength, &bytesBack);
*outputBufferLength = bytesBack;
return ret;}
```

Appendix I
DEMONSTRATION SOFTWARE AND CARD

For evaluation and test purposes, the Spectrum II Hybrid Insert Reader ships with demonstration software and a test card. These can be used to vary the reader's operational parameters and ensure that it is working correctly.

You will need the following:
1. An ID TECH Hybrid Insert Reader with a DB-9 cable and a 5-volt AC/DC power adapter.
2. A host PC running Windows 2000 or XP (32 bits only) with a serial port.
3. The smart card test card shipped with the unit.
4. The ID TECH demonstration software.

There are two separate sample programs, depending on whether the sample code is in Visual C++ or in Visual Basic.

PC/SC VISUAL C++ DEMONSTRATION PROGRAM

This PC/SC demonstration and sample program was written in Microsoft Visual C++ 6.0. The demonstration program is only meant to run when there is a PC/SC smart card reader attached to the system. The program will respond differently if the Spectrum reader is connected and selected than if another manufacturer's PC/SC reader is selected. The purpose of this program is to demonstrate how to find, connect to, and use the Spectrum PC/SC reader in a Windows environment.

On start up, this demonstration program will react differently if there is more than one PC/SC reader found to be attached to the system than if there is only one reader attached to the system. Buttons are grayed out when a function is not available. The demo program is configured to support a maximum of four connected PC/SC readers; any beyond this limit will not be displayed.

For any function that accesses the reader (except test hardware), the first operation is to either connect to the reader or to the card. This is done by selecting CONNECT TO RDR or CONNECT TO CARD.

If a microprocessor card is seated in the reader—and will remain in the reader until the smart card access portion of the connection is done—then the user should select CONNECT TO CARD. If smart card accesses are going to be made, the user should also select CONNECT TO CARD. For all other functions, select CONNECT TO READER. If it is desired to connect to the card when connected to the reader, or vice-versa, select DISCONNECT and then the appropriate button.

If the reader is not a Spectrum reader, the demo program will not give the option of connecting to the reader, because the reader's PC/SC driver will not support this vendor-specific function.

This demo program requires the use of idtlib.dll ID TECH's PC/SC hybrid reader support DLL. An application can be written to do all reader functions without the DLL, but the use of this DLL simplifies the use of reader functions.

The ID TECH PC/SC driver works on the WindowsXP and Windows2000 platforms only. This demo sample program should only be used with these operating systems as well.

INSTALLATION

Installation can be accomplished by clicking on the setup.exe program on the installation disk and following the instructions.

The program is composed of two parts: IDT PCSC Demo.exe and IDTLIB.DLL.

FUNCTION BUTTONS

SET MAGSTRIPE TIMEOUT establishes how long the reader/driver should wait for an expected event before aborting. The default timeout is 60 seconds. To change the default timeout, type a decimal number between 0 and 255 into the message field and select SET MAGSTRIPE TIMEOUT. To check the current value of the timeout, select GET MAGSTRIPE TIMEOUT.

READER VERSION, DLL VERSION, and DRIVER VERSION will get the respective version of each.

LATCH ON and LATCH OFF commands are normally used to latch a card in the reader while the ICC is being accessed. Pulling the card out of the reader during this period is called "tear." The latch command can also be used to prevent a user from inserting a card into the reader before the appropriate time.

SEND is used to send user commands through the PC/SC driver to the reader. This can be used to issue commands to the reader or card as if it was not in PC/SC mode. For instance, to get the reader's version, enter "39" in the message field and select SEND. Or, to get the reader's status, enter "24" in the message field and select SEND.

SEND APDU is used to send a particular APDU to a microprocessor (asynchronous) card. APDUs are always at least 4 bytes, and usually at least 5 bytes. An example of an APDU for the Sample T=0 smart card provided by ID TECH is to type 00 A4 00 00 02 31 40 in the message field and then select SEND APDU. This is the command to select file 3140, which can be accessed without a password. To read the first 4 bytes of the content of that file, enter 90 52 00 00 04 in the message field and select SEND APDU. The response should look like 00 06 90 48 F5 00 90 00. The 90 00 at the end of a smart card response is the normal response code for success.

GET MAGSTRIPE DATA powers the card off and, if a magstripe card is withdrawn before the time-out period, will display the magstripe data in the response window. If the reader remains at its default configuration, then the tracks of data will be displayed on separate lines with track 1 first and track 3 last.

TEST HARDWARE is designed to use a Sample ID TECH T=0 test card and verify that the reader card communication, seated switch, latch, smart card access, and magstripe access are functional. If the reader passes this test, it is functioning properly.

READ CONFIGURATION, CONFIGURE FOR EMV CARD, and CONFIGURE FOR ISO7816 are a group. EMV is a somewhat tighter specification than ISO7816. All cards, however, should be able to connect with the reader set to ISO7816, as this is a more general specification, but not all communication will function properly if the reader is to be used in an EMV environment.

CARD OFF will power down the current card in the reader.

EXIT will close the demo program and release the reader so that it can be attached by another application.

There are four card states: UNKNOWN, REMOVED, SEATED, and POWERED and four connection states: UNKNOWN, DISCONNECT, TOCARD, and TOREADER.

SAMPLE SOURCE CODE

This demonstration program is provided with source code (compiled with MS Visual C++ 6.0) to aid the developer in quickly making an application that properly performs the intended functions through the PC/SC interface and utilizing the supplied DLL.

PC/SC VISUAL BASIC DEMO PROGRAM

ID TECH has developed a PC/SC VB Demo Program named ID TECH PC/SC VB Demo. To test this program, the Spectrum PC/SC reader must be connected to a Windows 2000 or a Windows XP machine that has installed the ID TECH PC/SC driver. The ID TECH Standard CPU Smart Card is needed to perform the operation. The software supports multiple readers, but a PC with multiple serial ports is necessary to install PC/SC drivers properly for different kinds of PC/SC reader.

INSTALLATION

Installation can be accomplished by clicking on the setup.exe program on the installation CD and following the directions prompted by the windows.

LAUNCHING

Before launching the program, make sure that the Spectrum PC/SC reader is connected to the PC and the ID TECH standard CPU card is seated in the reader. The LED should blink quickly in an amber color. If the LED does not blink, it means the reader and the driver is not communicating. In this case, please consult Appendix X “Special Installation Instructions” to solve the problem.

OPERATION

In the program GUI, there are three groups of command buttons. Each group performs a different function test:

In the ID TECH CPU CARD group, select READ to read 15 characters from the smart card. Select WRITE to write a string to the card.

In the GENERAL CPU CARD group, select CONNECT TO CARD to establish the connection to the card. Then type an APDU command in the APDU test box and select SEND T=0 APDU. The data should be in ASCII format and nonprintable characters should be represented by \XX. For example, "\00\D6\00\00\07ID TECH" is a correct APDU. (A sample APDU will show up when you put the cursor in the APDU STRING box but do not click the mouse.) Repeat this step to perform all the operations desired, then press DISCONNECT to finish the operation.

It is recommended that the user first write all the APDU commands on paper, checking them carefully, before typing in these commands. Improper commands could permanently damage the test card.

The SPECIAL COMMANDS group controls latch and MSR functions, and enables the user to get reader, driver, and DLL versions. Connect to the card or reader by selecting either CONNECT TO READER or CONNECT TO CARD before performing the rest of the operation. After finishing all operations, select DISCONNECT in the GENERAL CPU CARD group to disconnect.

Selecting LATCH ON will lock the latch in the reader. A "click" sound will signal that the card has been locked in the reader. Selecting LATCH OFF will unlock the latch in the reader. Another "click" will indicate that the card can be removed.

Selecting READ VERSION will display the PC/SC driver version, the IdtLib.dll (please see "Program Source Code" section for description) version, and the PC/SC reader version.

To read a magnetic stripe card, insert the card into the reader. Select READ MSR and withdraw the card within the MSR timeout period. If successful, the magnetic stripe data should be displayed in the RESPONSE text box. Each time the PC is booted, the MSR timeout is set to a default value at about 60 seconds. Select MSR TIMEOUT to change the PC/SC driver MSR timeout delay.

PROGRAM SOURCE CODE

The ID TECH PC/SC VB Demo program is provided with source code to help the user in developing an application program. The code is written in Microsoft Visual Basic 6. Following is a list of files:

FILE	DESCRIPTION
IdtPcscDemo.frm	The main project form
PcscInterface.bas	A utility module to declare the Windows Smart Card API. The variable names, such as those in typedefs and function parameters, have been retained from thE C++ API definition.
PscsError.bas	A utility module to define error codes in Windows Smart Card API
IdtLib.dll	A utility dynamic linked library to provide support for ID TECH PC/SC readeR special function, such as Latch Function, MSR function…
IdtLib.bas	A utility module to declare functions in the IdtLib.dll
IdtPcscVbDemo.vbp	The project file

The user may develop an application program by modifying the main project form. All utility modules and the DLL can be included in the project with very little modification.