

USB (UNIVERSAL SERIAL BUS) KEYBOARD EMULATION FULL SIZE SWIPE READER TECHNICAL REFERENCE MANUAL

Manual Part Number 99875207 Rev 7

JANUARY 2004

PRELIMINARY

MAGTEK[®]

REGISTERED TO ISO 9001:2000

1710 Apollo Court
Seal Beach, CA 90740
Phone: (562) 546-6400
FAX: (562) 546-6301
Technical Support: (651) 415-6800
www.magtek.com

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of MagTek, Inc.

MagTek is a registered trademark of MagTek, Inc.

USB (Universal Serial Bus) Specification is Copyright© 1998 by Compaq Computer Corporation, Intel Corporation, Microsoft Corporation, NEC Corporation.

REVISIONS

Rev Number	Date	Notes
1	28 Aug 01	Initial Release
2	05 Oct 01	Title Page: Removed HID
3	12 Dec 02	Section 4, Command Number: Corrected GET and SET PROPERTY descriptions
4	28 Jan 03	Changed copyright symbol so .pdf copies would print on all printers
5	03 Jun 03	Front Matter: added ISO line to logo, changed Tech Support phone number, added new warranty statement.
6	09 Oct 03	Sec 5: replaced third paragraph with "Demo programs, version 1.2.0 and newer..."
7	22 Jan 04	Sec 1: Replaced third paragraph, p. 1, with current entry.

LIMITED WARRANTY

MagTek warrants that the products sold to Reseller pursuant to this Agreement will perform in accordance with MagTek's published specifications. This warranty shall be provided only for a period of one year from the date of the shipment of the product from MagTek (the "Warranty Period"). This warranty shall apply only to the original purchaser unless the buyer is authorized by MagTek to resell the products, in which event, this warranty shall apply only to the first repurchase.

During the Warranty Period, should this product fail to conform to MagTek's specifications, MagTek will, at its option, repair or replace this product at no additional charge except as set forth below. Repair parts and replacement products will be furnished on an exchange basis and will be either reconditioned or new. All replaced parts and products become the property of MagTek. This limited warranty does not include service to repair damage to the product resulting from accident, disaster, unreasonable use, misuse, abuse, customer's negligence, Reseller's negligence, or non-MagTek modification of the product. MagTek reserves the right to examine the alleged defective goods to determine whether the warranty is applicable.

Without limiting the generality of the foregoing, MagTek specifically disclaims any liability or warranty for goods resold in other than MagTek's original packages, and for goods modified, altered, or treated by customers.

Service may be obtained by delivering the product during the warranty period to MagTek (1710 Apollo Court, Seal Beach, CA 90740). If this product is delivered by mail or by an equivalent shipping carrier, the customer agrees to insure the product or assume the risk of loss or damage in transit, to prepay shipping charges to the warranty service location and to use the original shipping container or equivalent. MagTek will return the product, prepaid, via a three (3) day shipping service. A Return Material Authorization (RMA) number must accompany all returns.

MAGTEK MAKES NO OTHER WARRANTY, EXPRESS OR IMPLIED, AND MAGTEK DISCLAIMS ANY WARRANTY OF ANY OTHER KIND, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

EACH PURCHASER UNDERSTANDS THAT THE MAGTEK PRODUCT IS OFFERED AS IS. IF THIS PRODUCT DOES NOT CONFORM TO MAGTEK'S SPECIFICATIONS, THE SOLE REMEDY SHALL BE REPAIR OR REPLACEMENT AS PROVIDED ABOVE. MAGTEK'S LIABILITY, IF ANY, TO RESELLER OR TO RESELLER'S CUSTOMERS, SHALL IN NO EVENT EXCEED THE TOTAL AMOUNT PAID TO MAGTEK BY RESELLER UNDER THIS AGREEMENT. IN NO EVENT WILL MAGTEK BE LIABLE TO THE RESELLER OR THE RESELLER'S CUSTOMER FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE SUCH PRODUCT, EVEN IF MAGTEK HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

LIMITATION ON LIABILITY

EXCEPT AS PROVIDED IN THE SECTIONS RELATING TO MAGTEK'S LIMITED WARRANTY, MAGTEK'S LIABILITY UNDER THIS AGREEMENT IS LIMITED TO THE CONTRACT PRICE OF THE PRODUCTS.

MAGTEK MAKES NO OTHER WARRANTIES WITH RESPECT TO THE PRODUCTS, EXPRESSED OR IMPLIED, EXCEPT AS MAY BE STATED IN THIS AGREEMENT, AND MAGTEK DISCLAIMS ANY IMPLIED WARRANTY, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

MAGTEK SHALL NOT BE LIABLE FOR CONTINGENT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES TO PERSONS OR PROPERTY. MAGTEK FURTHER LIMITS ITS LIABILITY OF ANY KIND WITH RESPECT TO THE PRODUCTS, INCLUDING ANY NEGLIGENCE ON ITS PART, TO THE CONTRACT PRICE FOR THE GOODS.

MAGTEK'S SOLE LIABILITY AND BUYER'S EXCLUSIVE REMEDIES ARE STATED IN THIS SECTION AND IN THE SECTION RELATING TO MAGTEK'S LIMITED WARRANTY.

FCC WARNING STATEMENT

This equipment has been tested and found to comply with the limits for Class B digital device, pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a residential environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

FCC COMPLIANCE STATEMENT

This device complies with Part 15 of the FCC Rules. Operation of this device is subject to the following two conditions: (1) This device may not cause harmful interference; and (2) this device must accept any interference received, including interference that may cause undesired operation.

CANADIAN DOC STATEMENT

This digital apparatus does not exceed the Class B limits for radio noise for digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe B prescrites dans le Règlement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

CE STANDARDS

Testing for compliance to CE requirements was performed by an independent laboratory. The unit under test was found compliant to Class B.

UL/CSA

This product is recognized per Underwriter Laboratories and Canadian Underwriter Laboratories 1950.

TABLE OF CONTENTS

SECTION 1. FEATURES AND SPECIFICATIONS	1
FEATURES.....	1
HARDWARE CONFIGURATIONS	2
ACCESSORIES	2
REFERENCE DOCUMENTS.....	2
SPECIFICATIONS	2
SECTION 2. INSTALLATION	5
USB CONNECTION	5
WINDOWS PLUG AND PLAY SETUP	6
MOUNTING.....	6
SECTION 3. OPERATION	7
LED INDICATOR	7
CARD READ	7
SECTION 4. USB COMMUNICATIONS	9
HOST APPLICATIONS.....	9
CARD DATA	9
PROGRAMMABLE CONFIGURATION OPTIONS.....	10
LOW LEVEL COMMUNICATIONS	11
HID USAGES.....	11
REPORT DESCRIPTOR	12
COMMANDS.....	13
COMMAND NUMBER	13
DATA LENGTH.....	13
DATA.....	13
RESULT CODE.....	14
GET AND SET PROPERTY COMMANDS.....	14
SOFTWARE_ID PROPERTY	15
SERIAL_NUM PROPERTY	16
POLLING_INTERVAL PROPERTY	16
TRACK_ID_ENABLE PROPERTY	17
TRACK_DATA_SEND_FLAGS PROPERTY	18
TERMINATION_CHAR PROPERTY	18
SS_TK2_7BITS PROPERTY.....	19
SS_TK3_CADL PROPERTY	19
SS_TK3_ISO_ABA PROPERTY	20
SS_TK3_AAMVA PROPERTY	20
SS_TK3_7BITS PROPERTY.....	20
PRE_CARD_CHAR PROPERTY	21
POST_CARD_CHAR PROPERTY	21
PRE_TK_CHAR PROPERTY	22
POST_TK_CHAR PROPERTY.....	22
SECTION 5. DEMO PROGRAM	23
INSTALLATION.....	23
OPERATION.....	23
SOURCE CODE	24

FIGURES

Figure 1-1. USB Full Size Keyboard Emulation Swipe Reader-----	viii
Figure 1-2. Dimensions-----	4
Figure 2-1. Reader Cable and Connector-----	5

TABLES

Table 1-2. Specifications-----	3
Table 2-1. 4-Pin Connector-----	5

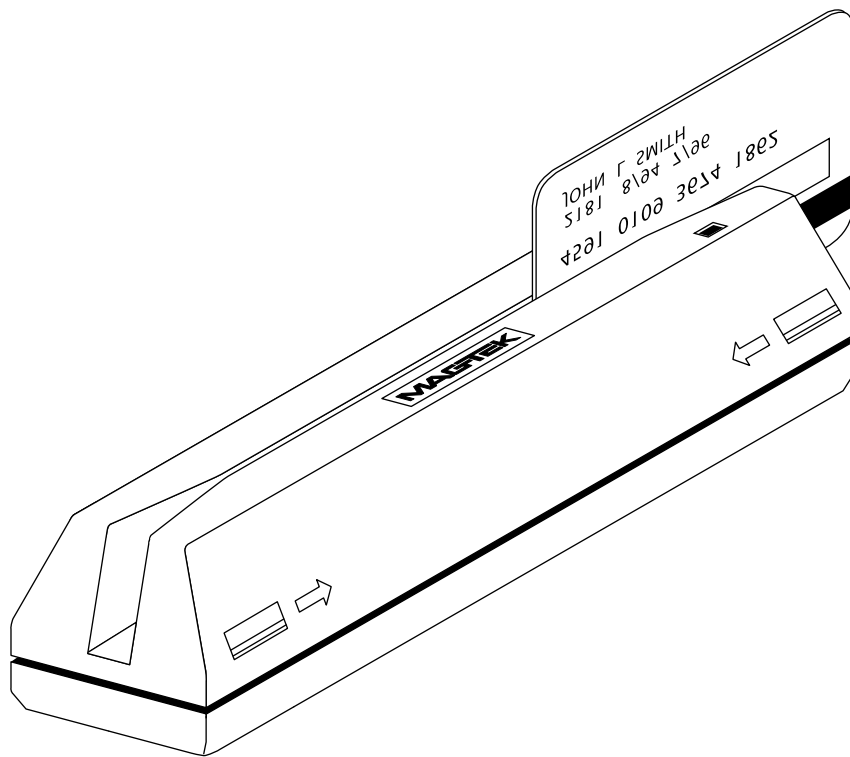


Figure 1-1. USB Full Size Keyboard Emulation Swipe Reader

SECTION 1. FEATURES AND SPECIFICATIONS

The USB (Universal Serial Bus), HID Keyboard Emulation, Full Size Swipe Reader is a compact magnetic stripe card reader, which conforms to ISO standards. The Reader is compatible with the PC series of personal computers or any device with a USB interface. A card is read by sliding it, stripe down and facing the LED side, through the slot either forward or backward.

A LED (Light Emitting Diode) indicator on the Reader panel provides the operator with continuous status of the Reader operations.

The reader emulates a USB Human Interface Device (HID) United States keyboard. This allows host applications designed to acquire card data from keyboard input to seamlessly acquire the card data from the USB swipe reader. Note that since this reader only emulates United States keyboards, it may not work with systems configured to use keyboards that are not United States keyboards.

Caution

If another keyboard is connected to the same host as this device and a key is pressed on the other keyboard while this device is transmitting, then the data transmitted by this device may get corrupted.

Because of potential "data interleave" issues associated with the USB Keyboard interface, MagTek recommends that the USB Keyboard Emulation MSR product should only be used by customers who have previously used MagTek's Keyboard Wedge MSR, or who are interfacing with an existing PC software application which gathers card data from the keyboard port. If previous applications were based upon RS-232 serial interface MSR's, or if this is a brand new development effort, it is strongly recommended that you use the MagTek's "standard version" of the USB MSR (Non-Keyboard Emulation Version). Please refer to Technical Manual 99875204 for further information regarding the "standard version" USB MSR.

FEATURES

Major features of the Swipe Reader are as follows:

- Powered through the USB – no external power supply required
- Hardware Compatible with PC or any computer or terminal with a USB interface
- Bidirectional card reading
- Reads encoded data that meets ANSI/ISO/CDL/AAMVA standards and others such as ISO track 1 format on track 2 or 3.
- Reads up to three tracks of card data
- LED for status
- Compatible with USB specification Revision 1.1
- Compatible with HID specification Version 1.1

USB HID Keyboard Emulation Swipe Reader

- Can use standard Windows drivers for communications. No third part device driver is required.
- Many programmable configuration options
- Non-volatile flash EEPROM memory for configuration storage
- Built-in 6 foot USB cable

HARDWARE CONFIGURATIONS

The hardware configuration is as follows:

Part Number	Tracks	Color
21088066	TK 1,2,3	Pearl White

ACCESSORIES

The accessories are as follows:

Part Number	Description
21042806	USB MSR Demo Program with Source Code (Diskette)
99510026	USB MSR Demo Program with Source Code (WEB)

REFERENCE DOCUMENTS

Axelson, Jan. *USB Complete, Everything You Need to Develop Custom USB Peripherals*, 1999. Lakeview Research, 2209 Winnebago St., Madison WI 53704, 396pp., <http://www.lvr.com>.

USB Human Interface Device (HID) Class Specification Version 1.1.

USB (Universal Serial Bus) Specification, Version 1.1, Copyright[©] 1998 by Compaq Computer Corporation, Intel Corporation, Microsoft Corporation, NEC Corporation.

USB Implementers Forum, Inc., www.usb.org.

SPECIFICATIONS

Table 1-2 lists the specifications for the USB Swipe Reader. Figure 1-2 shows the dimensions for the standard product. Other sizes are available by special order.

Table 1-2. Specifications

Reference Standards	ISO 7810 and ISO 7811/CDL/ AAMVA*
Power Input	5V From USB port
Recording Method	Two-frequency coherent phase (F2F)
Message Format	ASCII
Card Speed	3 to 50 IPS
MTBF	Electronics: 125,000 hours. Head: 1,000,000 passes

ELECTRICAL

Current Normal Mode Suspend Mode	30mA 300uA
--	---------------

MECHANICAL (STANDARD PRODUCT)

Dimensions	Length: 6.5" (165.1 mm), Width: 1.75" (44.45 mm) Height: 1.625" (41.28 mm)
Weight	6.60 oz. (187.6 gr.)
Cable length	6ft.
Connector	USB Type A plug

ENVIRONMENTAL

Temperature	
Operating	32°F to 131°F (0°C to 55°C)
Storage	-22°F to 158°F (-30°C to 70°C)
Humidity	
Operating	10% to 90% noncondensing
Storage	Up to 100% noncondensing
Altitude	
Operating	0-10,000 ft. (0-3048 m.)
Storage	0-50,000 ft. (0-15240 m.)

* ISO (International Standards Organization), CDL (California Drivers License), and AAMVA (American Association of Motor Vehicle Administrators).

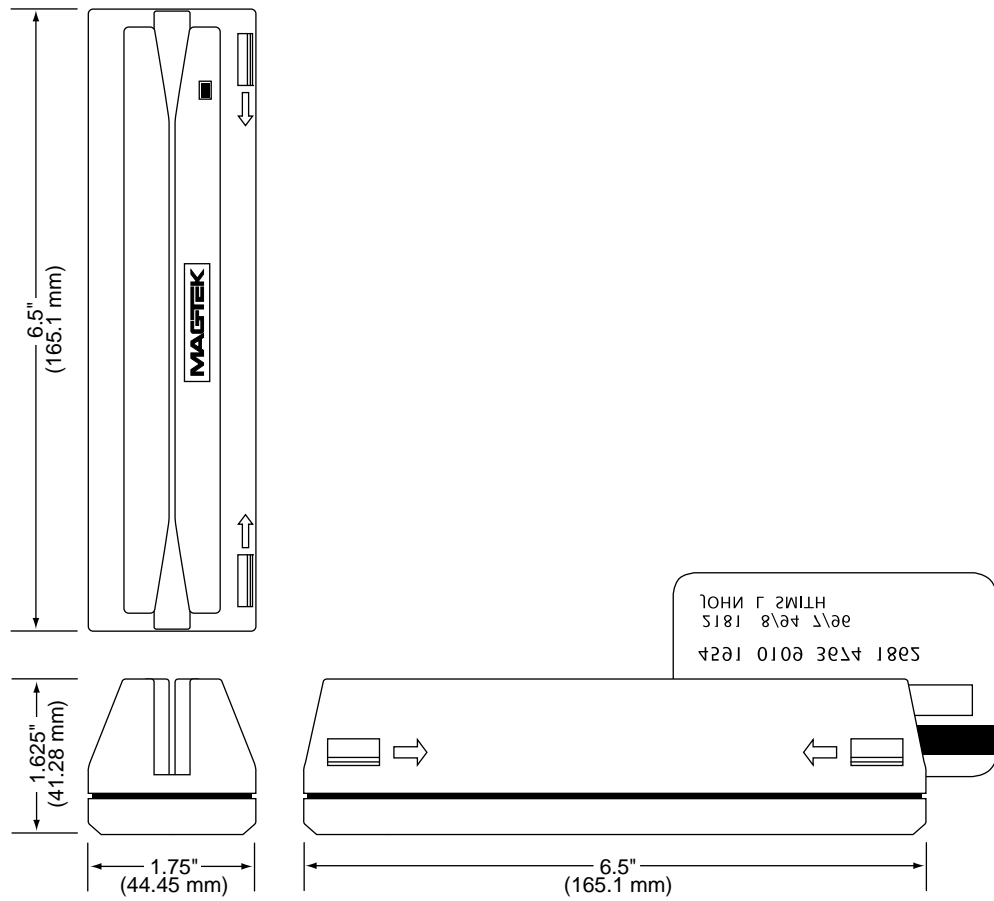


Figure 1-2. Dimensions

SECTION 2. INSTALLATION

This section describes the cable connection, the Windows Plug and Play Setup, and the physical mounting of the unit.

USB CONNECTION

Connect the USB cable to a USB port on the host. The Reader, LED Indicator, and pin numbers for the 4-pin connector are shown in Figure 2-1.

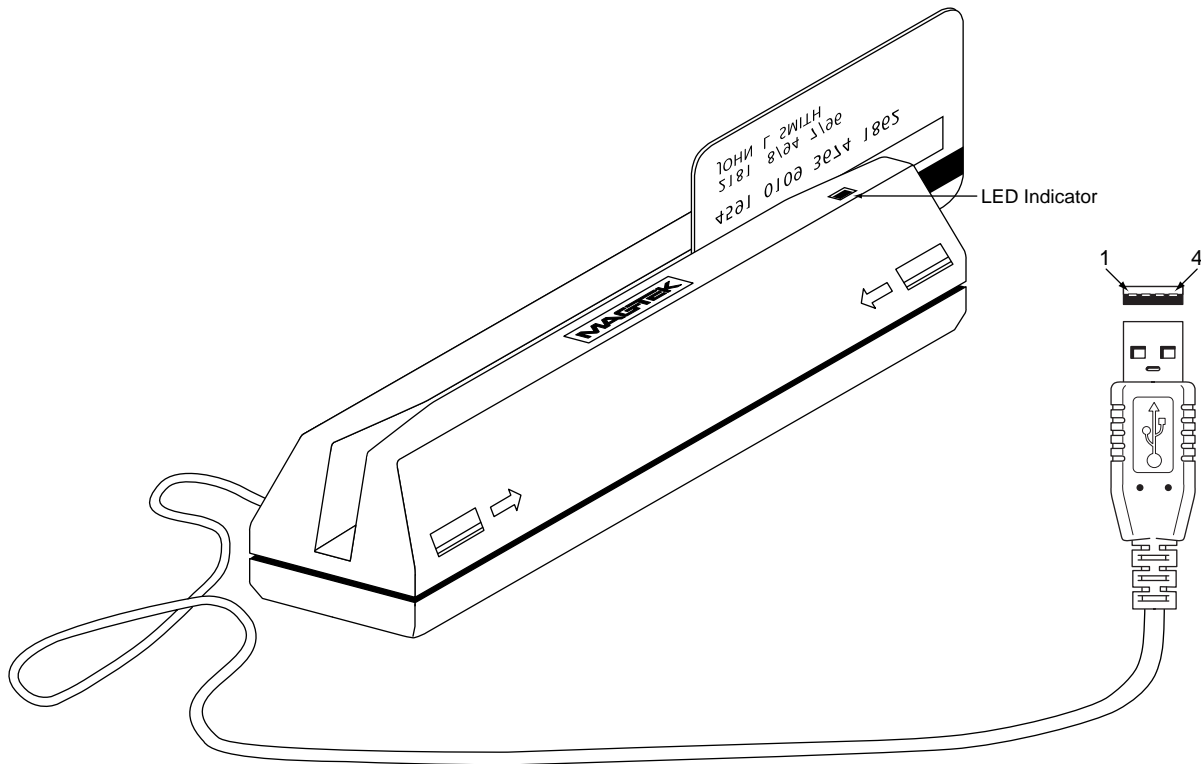


Figure 2-1. Reader Cable and Connector

Pin numbers and signal descriptions for the cable shown in the illustration are listed in Table 1-1.

Table 2-1. 4-Pin Connector

Pin Number	Signal	Cable Color
1	V _{CC}	Red
2	- Data	White
3	+Data	Green
4	Ground	Black

WINDOWS PLUG AND PLAY SETUP

On hosts with the Windows operating system, the first time the device is plugged into a specific USB port, Windows will pop up a dialog box, which will guide you through the process of installing a device driver for the device. After this process is completed once, Windows will no longer request this process as long as the device is plugged into the same USB port. The device driver that Windows will install for this device is the driver used for HID keyboard devices and it is part of the Windows operating system. When the dialog box pops up, follow the instructions given to you in the dialog box. Sometimes Windows will find all the files it needs on its own without giving you any prompts. Other times Windows will need to know the location of the files it needs. If Windows prompts you for the file locations, insert the CD that was used to install Windows on your PC and point Windows to the root directory of the CD. Windows should find all the files it needs there.

MOUNTING

1. Ensure the Reader is positioned on a flat, accessible surface with at least 4 inches clearance on either end for room to swipe a card. Orient the Reader so the side with the LED is facing the direction of intended use.

If fastening tape is to be used, clean the area that the Reader will be mounted on with isopropyl alcohol. Remove the adhesive protective cover on the fastening tape, and position the Reader and push down firmly.

2. Mount the Reader.

SECTION 3. OPERATION

This section describes the LED Indicator and Card Read.

LED INDICATOR

The LED indicator will be either off, red, or green. When the device is not powered, the LED will be off. When the device is first plugged in, the LED will be red. As soon as the device is plugged in, the host will try to enumerate the device. Once the device is enumerated the LED will turn green indicating that the device is ready for use. When a card is being swiped, the LED will turn off temporarily until the swipe is completed. If there are no errors decoding the card data then the LED will turn green. If there are any errors decoding the card data, the LED will turn red for approximately two seconds to indicate that an error occurred and then turn green. Anytime the host puts the device into suspend mode, the LED will turn off. Once the host takes the device out of suspend mode, the LED will return to the state it was in prior to entering suspend mode.

CARD READ

A card may be swiped through the Reader slot when the LED is green. The magnetic stripe must face toward the front (the side with the LED) and may be swiped in either direction. If there is data encoded on the card, the device will attempt to decode the data and then send the results to the host as if the data was being typed on a keyboard. After the results are sent to the host, the device will be ready to read the next card.

SECTION 4. USB COMMUNICATIONS

This device conforms to the USB specification revision 1.1. This device also conforms with the Human Interface Device (HID) class specification version 1.1. The device communicates to the host as a HID keyboard device. The latest versions of the Windows operating systems, Windows 98, Me, and 2000, all come with a standard Windows USB HID keyboard driver.

This is a full speed USB device. This device is powered from the USB bus. Its vendor ID is 0x0801 and its product ID is 0x0001. The device will go into suspend mode when directed to do so by the host. The device will wakeup from suspend mode when directed to do so by the host. The device does not support remote wakeup.

HOST APPLICATIONS

This device can be used with existing applications that acquire card data via keyboard input. Also, applications that communicate to this device can be easily developed. These applications can be easily developed using compilers such as Microsoft's Visual Basic or Visual C++. To demonstrate this device's card reading capabilities any application that accepts keyboard input such as Window's Notepad can be used.

CARD DATA

The card data is converted to ASCII and transmitted to the host as if it had been typed on a keyboard. Any data with ASCII values 0 – 31 or 127 will be transmitted as their equivalent control code combination. For example a carriage return value 13 (0D hex) will be sent as (^M) where ^ represents the Ctrl key on the keyboard.

Caution

If another keyboard is connected to the same host as this device and a key is pressed on the other keyboard while this device is transmitting, then the data transmitted by this device may get corrupted.

Because of potential "data interleave" issues associated with the USB Keyboard interface, MagTek recommends that the USB Keyboard Emulation MSR product should only be used by customers who have previously used MagTek's Keyboard Wedge MSR, or who are interfacing with an existing PC software application which gathers card data from the keyboard port. If previous applications were based upon RS-232 serial interface MSR's, or if this is a brand new development effort, it is strongly recommended that you use the MagTek's "standard version" of the USB MSR (Non-Keyboard Emulation Version). Please refer to Technical Manual 99875204 for further information regarding the "standard version" USB MSR.

The device's programmable configuration options affect the format of the card data.

The card data format for the default configuration is as follows:

```
[Tk1 SS][Tk1 Data][ES][Tk2 SS][Tk2 Data][ES][Tk3 SS][Tk3 Data][ES][CR]
```

where:

USB HID Keyboard Emulation Swipe Reader

Tk1 SS	=	% (7-bit start sentinel)
Tk2 SS	=	; (ISO/ABA 5-bit start sentinel) @ (7-bit start sentinel)
Tk3 SS	=	+ (ISO/ABA start sentinel) ! (CA drivers licence start sentinel) # (AAMVA start sentinel) & (7-bit start sentinel)
ES	=	? (end sentinel)
CR	=	(carriage return) (0D hex)

All data will be sent in upper case regardless of the state of the caps lock key on the keyboard. If no data is detected on a track then nothing will be transmitted for that track. If an error is detected on a track the ASCII character E will be sent in place of the track data to indicate an error.

The card data format for all programmable configuration options is as follows:

```
[P11][P13][Tk1 SS][Tk1 Data][ES][LRC][P14][P5][P13][Tk2 SS][Tk2  
Data][ES][LRC][P14][P5][P13][Tk3 SS][Tk3 Data][ES][LRC][P14][P5][P12]
```

where:

ES	=	? (end sentinel)
LRC	=	Longitudinal redundancy check character
P5	=	Terminating character
P11	=	Pre card character
P12	=	Post card character
P13	=	Pre track character
P14	=	Post track character
Tk1 SS	=	% (7-bit start sentinel)
Tk2 SS	=	; (ISO/ABA 5-bit start sentinel) P6 (7-bit start sentinel)
Tk3 SS	=	P8 (ISO/ABA start sentinel) P7 (CA drivers licence start sentinel) P9 (AAMVA start sentinel) P10 (7-bit start sentinel)

All fields with the format P# are programmable configuration property numbers. They are described in detail later in this document.

PROGRAMMABLE CONFIGURATION OPTIONS

This device has a number of programmable configuration properties. These properties are stored in non-volatile EEPROM memory. These properties can be configured at the factory or by the end user using a program supplied by MagTek. Programming these parameters requires low level communications with the device. During normal device operation, the device acts like a USB HID keyboard so the host operating system takes care of all low level communications with

the device so that the application developer is not burdened with these low level details. Details on how to communicate with the device to change programmable configuration properties follows in the next few sections. These details are included as a reference only. Most users will not need to know these details because the device will be configured at the factory or by a program supplied by MagTek. Most users may want to skip over the next few sections on low level communications and continue with the details of the configuration properties.

LOW LEVEL COMMUNICATIONS

It is strongly recommended that application software developers become familiar with the HID specification the USB specification before attempting to communicate directly with this device. This document assumes that the reader is familiar with these specifications. These specifications can be downloaded free from www.usb.org.

HID USAGES

HID devices send data in reports. Elements of data in a report are identified by unique identifiers called usages. The structure of the device's reports and the device's capabilities are reported to the host in a report descriptor. The host usually gets the report descriptor only once, right after the device is plugged in. The report descriptor usages identify the devices capabilities and report structures. For example, a device could be identified as a keyboard by analyzing the device's report descriptor. Usages are four byte integers. The most significant two bytes are called the usage page and the least significant two bytes are called usage IDs. Usages that are related can share a common usage page. Usages can be standardized or they can be vendor defined. Standardized usages such as usages for mice and keyboards can be found in the HID Usage Tables document and can be downloaded free at www.usb.org. Vendor defined usages must have a usage page in the range 0xff00 – 0xffff. All usages for this device use the standard HID keyboard usages or vendor defined magnetic stripe reader usage page 0xff00. The vendor defined usage IDs for this device are defined in the following table. The usage types are also listed. These usage types are defined in the HID Usage Tables document.

Magnetic Stripe Reader usage page 0xff00:

Usage ID (Hex)	Usage Name	Usage Type	Report Type
20	Command message	Data	Feature

REPORT DESCRIPTOR

The HID report descriptor is structured as follows:

Item	Value(Hex)
Usage Page (Generic Desktop)	05 01
Usage (Keyboard)	09 06
Collection (Application)	A1 01
Usage Page (Key Codes)	05 07
Usage Minimum (224)	19 E0
Usage Maximum (231)	29 E7
Logical Minimum (0)	15 00
Logical Maximum (1)	25 01
Report Size (1)	75 01
Report Count (8)	95 08
Input (Data, Variable, Absolute)	81 02
Report Count (1)	95 01
Report Size (8)	75 08
Input (Constant)	81 03
Report Count (5)	95 05
Report Size (1)	75 01
Usage Page (LEDs)	05 08
Usage Minimum (1)	19 01
Usage Maximum (5)	29 05
Output (Data, Variable, Absolute)	91 02
Report Count (1)	95 01
Report Size (3)	75 03
Output (Constant)	91 03
Report Count (6)	95 06
Report Size (8)	75 08
Logical Minimum (0)	15 00
Logical Maximum (101)	25 66
Usage Page (Key Codes)	05 07
Usage Minimum (0)	19 00
Usage Maximum (101)	29 66
Input (Data, Array)	81 00
Logical Maximum (255)	26 FF 00
Usage Page (vendor defined (MSR))	06 00 FF
Usage (command data)	09 20
Report Count	95 18
Feature (Data, Variable, Absolute, Buffered Bytes)	B2 02 01
End Collection	C0

COMMANDS

Command requests and responses are sent to and received from the device using feature reports. Command requests are sent to the device using the HID class specific request Set_Report. The response to a command is retrieved from the device using the HID class specific request Get_Report. These requests are sent over the default control pipe. When a command request is sent, the device will Nak the Status stage of the Set_Report request until the command is completed. This insures that as soon as the Set_Report request is completed, the Get_Report request can be sent to get the command response. The usage ID for the command message was shown previously in the Usage Table.

The following table shows how the feature report is structured for command requests:

Offset	Field Name
0	Command Number
1	Data Length
2 – 23	Data

The following table shows how the feature report is structured for command responses.

Offset	Field Name
0	Result Code
1	Data Length
2 – 23	Data

COMMAND NUMBER

This one byte field contains the value of the requested command number. The following table lists all the existing commands.

Value	Command Number	Description
0	GET_PROPERTY	Gets a property from the device
1	SET_PROPERTY	Sets a property in the device

DATA LENGTH

This one byte field contains the length of the valid data contained in the Data field.

DATA

This multi-byte field contains command data if any. Note that the length of this field is fixed at 22 bytes. Valid data should be placed in the field starting at offset 2. Any remaining data after the valid data should be set to zero. This entire field must always be set even if there is no valid data. The HID specification requires that Reports be fixed in length. Command data may vary in length. Therefore, the Report should be filled with zeros after the valid data.

RESULT CODE

This one byte field contains the value of the result code. There are two types of result codes: generic result codes and command specific result codes. Generic result codes always have the most significant bit set to zero. Generic result codes have the same meaning for all commands and can be used by any command. Command specific result codes always have the most significant bit set to one. Command specific result codes are defined by the command that uses them. The same code can have different meanings for different commands. Command specific result codes are defined in the documentation for the command that uses them. Generic result codes are defined in the following table.

Value	Result Code	Description
0	SUCCESS	The command completed successfully.
1	FAILURE	The command failed.
2	BAD_PARAMETER	The command failed due to a bad parameter or command syntax error.

GET AND SET PROPERTY COMMANDS

The Get Property command gets a property from the device. The Get Property command number is 0.

The Set Property command sets a property in the device. The Set Property command number is 1.

The Get and Set Property command data fields for the requests and responses are structured as follows:

Get Property Request Data:

Data Offset	Value
0	Property ID

Get Property Response Data:

Data Offset	Value
0 – n	Property Value

Set Property Request Data:

Data Offset	Value
0	Property ID
1 – n	Property Value

Set Property Response Data:

None

The result codes for the Get and Set Property commands can be any of the codes list in the generic result code table.

Property ID is a one byte field that contains a value that identifies the property. The following table lists all the current property ID values:

Value	Property ID	Description
0	SOFTWARE_ID	The device's software identifier
1	SERIAL_NUM	The device's serial number
2	POLLING_INTERVAL	The interrupt pipe's polling interval
3	TRACK_ID_ENABLE	Track enable / ID enable
4	TRACK_DATA_SEND_FLAGS	Track data send flags
5	TERMINATION_CHAR	Terminating char / per track or card flag
6	SS_TK2_7BITS	Start sentinel char for track 2 – 7 bit data
7	SS_TK3_CADL	Start sentinel char for track 3 – CA drivers license
8	SS_TK3_ISO_ABA	Start sentinel char for track 3 – ISO/ABA
9	SS_TK3_AAMVA	Start sentinel char for track 3 - AAMVA
10	SS_TK3_7BITS	Start sentinel char for track 3 – 7 bit data
11	PRE_CARD_CHAR	Pre card char
12	POST_CARD_CHAR	Post card char
13	PRE_TK_CHAR	Pre track char
14	POST_TK_CHAR	Post track char

The Property Value is a multiple byte field that contains the value of the property. The number of bytes in this field depends on the type of property and the length of the property. The following table lists all of the property types and describes them.

Property Type	Description
Byte	This is a one byte value. The valid values depend on the property.
String	This is a multiple byte ASCII string. Its length can be zero to a maximum length that depends on the property. The value and length of the string does not include a terminating NUL character.

SOFTWARE_ID PROPERTY

Property ID: 0
 Property Type: String
 Length: Fixed at 11 bytes
 Get Property: Yes
 Set Property: No
 Description: This is an 11 byte read only property that identifies the software part number and version for the device. The first 8 bytes represent the part number and the last 3 bytes represent the version. For example this string might be “21042802A05”. Examples follow:

Example Get SOFTWARE_ID property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	00

Example Get SOFTWARE_ID property Response (Hex):

Result Code	Data Len	Prp Value
00	01	32 31 30 34 32 38 30 32 41 30 35

SERIAL_NUM PROPERTY

Property ID: 1
 Property Type: String
 Length: 0 – 15 bytes
 Get Property: Yes
 Set Property: Yes
 Default Value: The default value is no string with a length of zero.
 Description: The value is an ASCII string that represents the device’s serial number. This string can be 0 – 15 bytes long. This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. The value of this property, if any, will be sent to the host when the host requests the USB string descriptor. When this property is changed, the unit must be power cycled to have these changes take effect for the USB descriptor. If a value other than the default value is desired, it can be set by the factory upon request. Examples follow.

Example Set SERIAL_NUM property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	04	01	31 32 33

Example Set SERIAL_NUM property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get SERIAL_NUM property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	01

Example Get SERIAL_NUM property Response (Hex):

Result Code	Data Len	Prp Value
00	03	31 32 33

POLLING_INTERVAL PROPERTY

Property ID: 2
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 1
 Description: The value is a byte that represents the devices polling interval for the Interrupt In Endpoint. The value can be set in the range of 1 – 255 and has units of milliseconds. The polling interval tells the host how often to poll the device for card data packets. For example, if the polling interval is set to 10, the host will poll the device for card data packets every 10ms. This property can be used to speed up or slow down the time it takes to send card data to the host. The trade-off is that speeding up the card data transfer rate increases the USB bus bandwidth used by the device, and slowing down the card data transfer rate decreases the USB bus bandwidth used by the device. This property is stored in non-volatile EEPROM

memory so it will not change when the unit is power cycled. The value of this property, if any, will be sent to the host when the host requests the device’s USB endpoint descriptor. When this property is changed, the unit must be power cycled to have these changes take effect for the USB descriptor. If a value other than the default value is desired, it can be set by the factory upon request. Examples follow:

Example Set POLLING_INTERVAL property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	02	02	0A

Example Set POLLING_INTERVAL property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get POLLING_INTERVAL property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	02

Example Get POLLING_INTERVAL property Response (Hex):

Result Code	Data Len	Prp Value
00	01	0A

TRACK_ID_ENABLE PROPERTY

Property ID: 3
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 95 (hex)
 Description: This property is defined as follows:

id	0	T ₃	T ₃	T ₂	T ₂	T ₁	T ₁
----	---	----------------	----------------	----------------	----------------	----------------	----------------

Id 0 – Decodes standard ISO/ABA cards only
 1 – Decodes AAMVA, CA DL/ID and 7-bit cards also

T_# 00 – Track Disabled
 01 – Track Enabled
 10 – Track Enabled/Required (Error if blank)

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

TRACK_DATA_SEND_FLAGS PROPERTY

Property ID: 4
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 63 (hex)
Description: This property is defined as follows:

0	SS	ES	LRC	0	LC	Er	Er
---	----	----	-----	---	----	----	----

SS 0 – Don't send Start Sentinel for each track
1 – Send Start Sentinel for each track

ES 0 – Don't send End Sentinel for each track
1 – Send End Sentinel for each track

LRC 0 – Don't send LRC for each track
1 – Send LRC for each track

Note that the LRC is the unmodified LRC from the track data. To verify the LRC the track data needs to be converted back from ASCII to card data format and the start sentinels that were modified to indicate the card encode type need to be converted back to their original values.

LC 0 – Send card data as upper case
1 – Send card data as lower case

Note that the state of the Caps Lock key on the host keyboard has no effect on what case the card data is transmitted in.

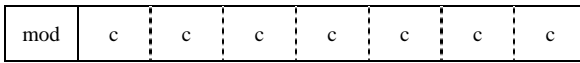
Er 00 – Don't send any card data if error
01 – Don't send track data if error
11 – Send 'E' for each track error

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

TERMINATION_CHAR PROPERTY

Property ID: 5
Property Type: Byte
Length: 1 byte
Get Property: Yes

Set Property: Yes
 Default Value: 0D (hex) (carriage return)
 Description: This property is defined as follows:



mod 0 – Send c after card data
 1 – Send c after each track

c 1-127 – 7 bit ASCII char code
 0 – send nothing

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

SS_TK2_7BITS PROPERTY

Property ID: 6
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 40 (hex) '@'
 Description: This character is sent as the track 2 start sentinel for cards that have track 2 encoded in 7 bits per character format. If the value is 0 no character is sent. If the value is in the range 1 – 127 then the equivalent ASCII character will be sent.

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

SS_TK3_CADL PROPERTY

Property ID: 7
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 21 (hex) '!'
 Description: This character is sent as the track 3 start sentinel for cards that have track 3 encoded in California drivers license format. If the value is 0 no character is sent. If the value is in the range 1 – 127 then the equivalent ASCII character will be sent.

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

SS_TK3_ISO_ABA PROPERTY

Property ID: 8
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 2B (hex) '+'
Description: This character is sent as the track 3 start sentinel for cards that have track 3 encoded in ISO/ABA format. If the value is 0 no character is sent. If the value is in the range 1 – 127 then the equivalent ASCII character will be sent.

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

SS_TK3_AAMVA PROPERTY

Property ID: 9
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 23 (hex) '#'
Description: This character is sent as the track 3 start sentinel for cards that have track 3 encoded in AAMVA format. If the value is 0 no character is sent. If the value is in the range 1 – 127 then the equivalent ASCII character will be sent.

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

SS_TK3_7BITS PROPERTY

Property ID: 10 (0A hex)
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 26 (hex) '&'

Description: This character is sent as the track 3 start sentinel for cards that have track 3 encoded in 7 bits per character format. If the value is 0 no character is sent. If the value is in the range 1 – 127 then the equivalent ASCII character will be sent.

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

PRE_CARD_CHAR PROPERTY

Property ID: 11 (0B hex)
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0

Description: This character is sent prior to all other card data. If the value is 0 no character is sent. If the value is in the range 1 – 127 then the equivalent ASCII character will be sent.

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

POST_CARD_CHAR PROPERTY

Property ID: 12 (0C hex)
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0

Description: This character is sent after all other card data. If the value is 0 no character is sent. If the value is in the range 1 – 127 then the equivalent ASCII character will be sent.

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

PRE_TK_CHAR PROPERTY

Property ID: 13 (0D hex)
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0

Description: This character is sent prior to the data for each track. If the value is 0 no character is sent. If the value is in the range 1 – 127 then the equivalent ASCII character will be sent.

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

POST_TK_CHAR PROPERTY

Property ID: 14 (0E hex)
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0

Description: This character is sent after the data for each track. If the value is 0 no character is sent. If the value is in the range 1 – 127 then the equivalent ASCII character be sent.

This property is stored in non-volatile EEPROM memory so it will not change when the unit is power cycled. When this property is changed, the unit must be power cycled to have these changes take effect. If a value other than the default value is desired, it can be set by the factory upon request.

SECTION 5. DEMO PROGRAM

The purpose of this demo program is not to demonstrate card reading with this HID keyboard emulation device. Use a text editor application such as Windows Notepad to demonstrate card reading for this HID keyboard emulation device. Any application that allows user input from a keyboard should be sufficient to demonstrate card reading for this device.

The primary purpose of the demo program, when used with this HID keyboard emulation device, is to allow users to change the device's programmable configuration properties. This is accomplished by sending commands to the device with the demo program. The demo program also comes with source code that can be used as a guide for application developers who want to change the device's programmable configuration properties in an application. However, it is unlikely that application developers will want to change these properties in an application since these properties only need to be set once and can be set at the factory. This program is written in Visual Basic.

Demo programs, version 1.2.0 and newer work on Windows 98, Me, 2000 and XP. Older versions do not support the HID keyboard emulation device on Windows 2000 or XP. These older versions only work on Windows 98 and Me.

When the demo program is run, a button for reading cards is displayed along with a button for sending commands. The card reading option is not supported for this HID keyboard emulation device. Use a text editor application such as Windows Notepad to demonstrate card reading for this HID keyboard emulation device.

The part numbers for the demo program can be found in this document in Section 1 under Accessories.

INSTALLATION

To install the demo program, run the setup.exe file and follow the instructions given on the screen.

OPERATION

To operate the demo program perform the following steps:

- Plug the device into a USB port on the host
- If this is the first time the device has been plugged into the host, then follow the instructions on the screen for installing the Windows HID device driver. This is explained in more detail in the installation section of this document.
- Run the demo program.
- To read cards and view the card data do not use the demo program. Use a text editor program such as Windows Notepad.
- To send commands to the device, click on the send commands button.
- Enter a command in the Message edit box. All data entered should be in hexadecimal bytes with a space between each byte. Enter the command number followed by the command data

if there is any. The application will automatically calculate and send the command data length for you. For example, to send the GET_PROPERTY command for property SOFTWARE_ID enter 00 00.

- Press Enter or click on Send message to send the command and receive the result.
- The command request and the command result will be displayed in the Communications Dialog edit box.
- The Clear Dialog button clears the Communication Dialog edit box.

SOURCE CODE

Source code is included with the demo program. It can be used as a guide for application development. It is described in detail, with comments, to assist developers. The book *USB Complete* by Jan Axelson is also a good guide for application developers, especially the chapter on Human Interface Device Host Applications (see “Reference Documents” in Section 1).