USER MANUAL

MiniMag Readers
Models IDMB

MSR OPOS

80066511-001
Rev. A 9-20-06

ID TECH SOFTWARE LICENSE AGREEMENT

ID TECH ("LICENSOR") IS WILLING TO LICENSE THIS SOFTWARE TO YOU ONLY IF YOU ACCEPT ALL OF THE TERMS IN THIS LICENSE AGREEMENT. PLEASE READ THE TERMS CAREFULLY BEFORE YOU AGREE BECAUSE YOU WILL BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS, LICENSOR WILL NOT LICENSE THIS SOFTWARE TO YOU.

Ownership of the Software

1. The Licensor software program ("Software") and any accompanying written materials are owned by Licensor [or its suppliers] and are protected by United States copyright laws, by laws of other nations, and by international treaties.

Grant of License

2. Licensor grants the right to use the Software in conjunction with an ID TECH product. You may load one copy into permanent memory of one computer and may use that copy only on that same computer.

Restrictions on Use and Transfer

3. The Software may not be copied, except that (1) one copy of the Software may be made solely for backup or archival purposes, and (2) the Software may be transfer to a single hard disk provided the original is kept solely for backup or archival purposes. The written materials may not be copied.

4. The Software may be permanently transferred and any accompanying written materials (including the most recent update and all prior versions) if no copies are retained and the transferee agrees to be bound by the terms of this Agreement. Such a transfer terminates your license. The software may not be rented or leased or otherwise transferred or assigned the right to use the Software, except as stated in this paragraph.

5. The software may not be reverse engineered, decompiled, or disassembled.

Limited Warranty

6. If used in conjunction with an ID TECH product, Licensor warrants that the Software will perform substantially in accordance with the accompanying written materials for a period of 90 days from the date of your receipt of the Software. Any implied warranties on the Software are limited to 90 days. Some states and territories do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

7. LICENSOR DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, WITH RESPECT TO THE SOFTWARE AND ANY ACCOMPANYING WRITTEN MATERIALS. This limited warranty gives you specific legal rights. You may have others, which vary from state to state.

8. LICENSOR'S ENTIRE LIABILITY AND YOUR EXCLUSIVE REMEDY SHALL BE REPLACEMENT OF THE SOFTWARE THAT DOES NOT MEET LICENSOR'S LIMITED WARRANTY. Any replacement Software will be warranted for the remainder of the original warranty period or 30 days, whichever is longer.

9. This Limited Warranty is void if failure of the Software has resulted from modification, accident, abuse, or misapplication.

10. IN NO EVENT WILL LICENSOR BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY LOSS OF PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF YOUR USE OR INABILITY TO USE THE SOFTWARE. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply you.

11. This Agreement is governed by the laws of the state of California.

12. For any questions concerning this Agreement or to contact Licensor for any reason, please write: International Technologies & Systems Corporation, 10721 Walker Street, Cypress, CA 90630 or call (714) 761-6368.

13. U.S. Government Restricted Rights. The Software and documentation are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c)(1) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1)(ii) and (2) of Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable. Supplier is ID TECH, 10721 Walker Street, Cypress, CA 90630.

Revision History

Revision Date	Description
6/15/2005	First draft release for internal review
11/11/2005	Modify for OPOS 2.00.000
12/14/2005	V2.2, add version compatibility to firmware v1.30, v1.66, v1.69
1/24/2006	V2.31 support USB HID KB connector
2/21/2006	V2.42 add supporting USB HID connector
3/13/06	V2.43 add supporting USB RS232 connector
5/19/06	Modify for v2.52
9/20/06	Release

Content

Description:

The documentation describes the properties, methods, and events by our MiniMag II MSR OPOS component. The component includes two parts: a Control Object running on the upper level, which is an ActiveX control, and a Service Control running on the lower level, which is an automation server. The properties, methods, and events are exposed by the Control Object. For example, when the Control Object is imported into your project as an ActiveX control, you will see all the properties, methods, and events.

Important Note:

For PS/2 keyboard or USB HID Keyboard interfaces, the standard keyboard should not be pressed when swiping cards, otherwise the card data will be wrong , MSR OPOS Driver will display a warning dialog and the data will be discarded .

Target Device:

1. MiniMag II PS/2 Key Board Interface
2. MiniMag II USB HID KB Interface
3. MiniMag II USB HID Interface
4. MiniMag II RS232/USBCDC interface

Platform:

Microsoft Windows XP, 2000, 98

Service Object and Control Object:

Service Object Version: 1.9.52 Dll Version: 2.52

Control Object Version:1.9.0

Methods, Properties and Events Of MSR

The following sections describe the methods, properties, and events.

Properties of MSR:

Please see UPOS Spec if the detailed information is wanted.

NOTE: CO --- Control Object
SO --- Service Object
AP or App --- the abbreviation of Application.

Property Group1---Description

Name	*Type*	*Mutability*	*Use After*	*Description*	*Support?*
DeviceControlDescription	String	read-only	--	Identify the Control Object and the company that produced it	Yes
DeviceControlVersion	*int32*	read-only	--	hold the Control Object version number.	Yes
DeviceServiceDescription	*String*	read-only	open	identify the Service Object supporting the device and the company that produced it	Yes
DeviceServiceVersion	*int32*	read-only	open	hold the Service Object version number.	Yes
PhysicalDeviceDescription	*string*	read-only	open	identify the device and any pertinent information about it.	Yes
PhysicalDeviceName	*string*	read-only	open	identify the device and any pertinent information about it.	Yes

Property Group2---Control

Name	*Type*	*Mutability*	*Use After*	*Description*	*Support?*
Claimed	*Boolean*	read-only	open	MiniMag must be claimed for exclusive use before access its methods and properties, and before any events to be fired. It is initialized to FALSE by the **Open** method. It is set to TRUE after the method **Claim** is successfully called.	Yes
AutoDisable	*Boolean*	read-write	open	When TRUE, as soon as an event **DataEvent** is received, then **DeviceEnabled** is automatically to FALSE. It is initialized to FALSE by the **Open** method.	Yes
DeviceEnabled	*Boolean*	read-write	open & claim	When FALSE, MiniMag has been disabled and any subsequent input will be discarded (No DataEvent could be received even if the card is swiped). It is initialized to FALSE by the **Open** method.	Yes

FreezeEvents	*boolean*	read-write	open	When TRUE, events are not required to be delivered and will be held by SO until events are unfrozen. It is initialized to FALSE by the **Open** method.	Yes
DataEventEnabled	*boolean*	read-write	open	When TRUE, a **DataEvent** or **ErrorEvent** will be delivered immediately when had. (Of course , ***FreezeEvents=FALSE and DeviceEnabled=TRUE*** is a prerequisit). It is initialized to FALSE by the **Open** method.	Yes
CapPowerReporting	*int32*	read-only	open	Identifies the reporting capabilities of the device about Power. It seems that MiniMag doesn't support in the hardware.	No
PowerNotify	*int32*	read-write	open	Contains the type power notification selection made by the Application. is initialized to OPOS_PN_DISABLED by the **Open** method.	No
PowerState	*int32*	read-only	open	Contains the current power condition. It seems that MiniMag doesn't support in the hardware.	No
State	*int32*	Read-only	--	Contains the current state of the Control. It can be set to one of the four values: Closed, Idle, Busy, or Error.	Yes
DataCount	*int32*	Read-only	open	Holds the number of enqueued **DataEvent**s remained in the queue.	Yes
CheckHealthText	*string*	read-only	open	Holds the results of the most recent call to the **CheckHealth** method. Before the first **CheckHealth** method call, its value is uninitialized.	Yes

Property Group3---Track Control

Name	***Type***	***Mutability***	***Use After***	***Description***	***Support?***
CapISO	*boolean*	read-only	open	If TRUE, MiniMag supports ISO cards.	Yes
CapJISOne	*boolean*	read-only	open	If TRUE, MiniMag supports JIS Type-I cards. JIS-I cards are a superset of ISO cards. Therefore, if **CapJISOne** is true, it is implied that **CapISO** is also TRUE.	Yes
CapJISTwo	*boolean*	read-only	open	If TRUE, MiniMag supports JIS type-II cards.	Yes
CapTransmitSentinels	*boolean*	read-only	open	If TRUE, MiniMag is able to transmit the start and end sentinels. e.g. start sentinel could be '%' or ';', and stop sentinel could be '?'.	Yes
DecodeData	*boolean*	read-write	open	If TRUE, each byte of track data properties is mapped from its original encoded bit sequence (as it exists on the magnetic card) to its corresponding decoded ASCII bit sequence.	Yes
ParseDecodeData	*boolean*	read-write	open	When TRUE, the decoded data contained within the **Track1Data** and **Track2Data** properties is further separated into fields for access via various other properties. If ***DecodeData=FALSE***, ***ParseDecodeData must be false.***	Yes
TransmitSentinels	*boolean*	read-write	open	If TRUE, the **Track1Data**, **Track2Data**, **Track3Data**, and **Track4Data** properties contain start and end sentinel values. Otherwise only the track data between these sentinels.	Yes
TracksToRead	*int32*	read-write	open	Indicate which track data that the App wishes to get following a card sweep**.**	Yes
ErrorReportingType	*int32*	Read-write	open	Holds the type of errors to report via **ErrorEvent**s. This property has one of the following values: MSR_ERT_CARD or MSF_ERT_TRACK	Yes

Property Group4---TrackData

Name	*Type*	*Mutability*	*Use After*	*Description*	*Support?*
Track1Data	*binary*	read-only	open	Holds the track 1 data obtained from the most recently swept card. If **DecodeData** is true, then it has been decoded from the "raw" format. it may also be parsed into other properties when the **ParseDecodeData** property is set.	Yes
Track1DiscretionaryData	*binary*	read-only	open	Holds the track 1 discretionary data obtained from the most recently swept card. It may be NULL when: **1)** The field was not included in the track data obtained, or, **2)** The track data format was not supported, **3)** **ParseDecodeData** is false.	Yes
Track2Data	*binary*	read-only	open	Holds the track 2 data obtained from the most recently swept card. If **DecodeData** is true, then it has been decoded from the "raw" format. it may also be parsed into other properties when the **ParseDecodeData** property is set.	Yes
Track2DiscretionaryData	*binary*	read-only	open	Holds the track 2 discretionary data obtained from the most recently swept card. It may be NULL when: **1)** The field was not included in the track data obtained, or, **2)** The track data format was not supported, **3)** **ParseDecodeData** is false.	Yes
Track3Data:	*binary*	read-only	open	Holds the track 3 data obtained from the most recently swept card.	Yes
Track4Data	*binary*	read-only	open	Holds the track 4 data (JIS-II) obtained from the most recently swept card.	Yes

Property Group5---ParsedData

Name	***Type***	***Mutability***	***Use After***	***Description***	***Support?***
AccountNumber	*string*	read-only	Open	Holds the account number obtained from the most recently swept card. it is initialized to NULL if: **1)** The field was not included in the track data obtained, or, **2)** The track data format was not suported, or, **3)** **ParseDecodeData** is false.	Yes
ExpirationData	*string*	read-only	Open	Holds the expiration date obtained from the most recently swept card. Others are same as ***AccountNumber.***	Yes
FirstName	*string*	read-only	Open	Holds the first name obtained from the most recently swept card. Others are same as ***AccountNumber.***	Yes
MiddleInitial	*string*	read-only	Open	Holds the middle initial obtained from the most recently swept card. Others are same as ***AccountNumber.***	Yes
Surname	*string*	read-only	Open	Holds the surname obtained from the most recently swept card. Others are same as ***AccountNumber.***	Yes
Title	*string*	read-only	Open	Holds the title obtained from the most recently swept card.. Others are same as ***AccountNumber.***	Yes
Suffix	*string*	read-only	Open	Holds the suffix obtained from the most recently swept card.. Others are same as ***AccountNumber.***	Yes
ServicCode	*string*	read-only	Open	Holds the service code obtained from the most recently swept card. Others are same as ***AccountNumber.***	Yes

Property Group6--- Statistic

Name	***Type***	***Mutability***	***Use After***	***Expected Result***	***Test Result***
CapStatisticsReporting	*binary*	read-write	Open	If true ,the SO can get device information to a XML statistics	No
CapUpdateStatistics	*binary*	read-write	Open	If true ,the SO can update the XML statistics	No

Property Group7---Firmware

Name	***Type***	***Mutability***	***Use After***	***Expected Result***	***Test Result***
CapCompareFirmwareVersion	*binary*	read-write	Open	If true ,the SO can compare the Firmware version	No
CapUpdateFirmware	*binary*	read-write	Open	If true ,the SO can update the firmware of the device	No

Methods of MSR:

These function declarations may be different when the Control Object(OPOSMSR.OCX) is imported into your application project. Please see UPOS Spec if the detailed information is wanted.

1)Open

Syntax **LONG Open (BSTR** *DeviceName***);**

Remarks Call to open a device for subsequent I/O.

Device Name:

PS2 Keyboard Interfae:	"IDTECH_MMII_PS/2KB"
USB HID KB interface:	"IDTECH_MMII_USBKB" or "IDTechMiniMagII"
USB HID interface	"IDTECH_MMII_USBHID"
RS232 interface	"IDTECH_MMII_RS232"
USBCDC interface	“IDTECH_MMII_USBCDC” or “IDTECH_MMII_RS232

Support? Yes

This method finds more parameters in the Windows Register Tables on key or subkeys:
HKEY_LOCAL_MACHINE\Software\OLEforRetail\ServiceOPOS\MSR\xxxInfterface

RS232 interface and USBCDC interface:

Subkey: Connector
Key value name: CONNECTOR
Key value: "COM1/baud=9600/parity=N/data=8/stop=1"

COM1 specify the serial port name, if the device is plugged in the second port, it should modified as "COM2". The rest settings should be right defined according to the settings of the reader.

Key value Name: COM1, this name should be same the first filed of CONNECTOR key value. The field is separated by "/". So, if the reader is changed to other port, this key value is needed modified also.

USB HID interface:

Subkey: Connector:
Key value name: CONNECTOR
Key value: USBHID/0acd/0500

First field USBHID specify the type of the connector. 0acd is the USB device vendor ID, 0500 is the reader product ID for USB HID connector device.

USB HID KB interface:

Subkey: Connector:

Key value name: CONNECTOR

Key value: USBHID/0acd/0520

First field USBHID specify the type of the connector. 0acd is the USB device vendor ID, 0520 is the reader product ID.

PS2 keyboard interface:

Subkey: Connector:

Key value name: CONNECTOR

Key value: PS2

2)ClaimDevice ***Added in Release 1.5***

Syntax **LONG ClaimDevice (LONG** *Timeout***);**

Remarks Call this method to request exclusive access to the device. Many devices require an application to claim them before they can be used. ***Release 1.0 – 1.4*** In releases prior to 1.5, this method is named Claim.

Support? Yes

3)CheckHealth

Syntax **LONG CheckHealth (LONG** *Level***);**

Remarks Called to test the state of a device.

Support? Yes

Description When select CH_INTERNAL, SO will return the firmwareversion of the Msr device, if not it tells that there is something wrong with the device . CheckHealthText property will be "Internal HCheck: Successful" ,if reading the firmware version is successful .

When select CH_EXTERNAL,SO will display a swiping card dialog ,it will display the "Real data " of the card , include Start Sentinel and End Sentinel. CheckHealth Text property will show "External HCheck:: HCheck: Complete" , after close the dialog .

When select CH_INTERACTIVE , SO will display a dialog , which include firmware version and swiping card .CheckHealthText property will show "External HCheck:: HCheck: Complete" , after close the dialog .

4)ClearInput

Syntax **LONG ClearInput ();**

Remarks Called to clear all device input that has been buffered.

Support? Yes

5)DirectIO

Syntax **LONG DirectIO (LONG** *Command*, **LONG*** *pData*, **BSTR*** *pString***);**

Remarks Call to communicate directly with the Service Object.

Support? Yes

Description In the current, it implemented incompletely**.** We will improve it in the next release.

6)ReleaseDevice ***Added in Release 1.5***

Syntax **LONG ReleaseDevice ();**

Remarks Call this method to release exclusive access to the device.

Release 1.0 – 1.4

In releases prior to 1.5, this method is named **Release**.

Support? Yes

7)Close

Syntax **LONG Close ();**

Remarks Called to release the device and its resources.

Support? Yes

8) ResetStatistics ***Added in Release 1.8***

Syntax **LONG ResetStatistics(BSTR m_StatisticsBuffer);**

Remarks Called to Resets the defined resettable statistics in a device to zero.

Support? No

9) retrieveStatistics ***Added in Release 1.8***

Syntax **LONG RetrieveStatistics(BSTR* m_pStatisticsBuffer);**

Remarks Called to Retrieves the requested statistics from a device.

Support? No

10) UpdateStatistics ***Added in Release 1.8***

Syntax **LONG UpdateStatistics(BSTR m_StatisticsBuffer);**

Remarks Called to Updates the defined resettable statistics in a device.

Support? No

11) CompareFirmwareVersion

Syntax **LONG CompareFirmwareVersion(BSTR m_FirmwareFileName, long* m_pResult);**

Remarks Called to compare the firmware version with current firmware version of the device

Support? No

7) UpdateFirmware

Syntax **LONG UpdateFirmware(BSTR m_FirmwareFileName);**

Remarks Called to update current firmware.

Support? No

Events of MSR:

These events are fired by the Service Object when it is necessary. The following functions are, in fact, the event-handlers that can be added into the applications. Then the applications can receive these events and do some processing accordingly. Please see UPOS Spec if the detailed information is wanted.

1)DataEvent

Syntax **void DataEvent (LONG** *Status***);**

The *Status* parameter contains the input status. Its value is Control-dependent, and may describe the type or qualities of the input.

Remarks Fired to present input data from the device to the application.

Description a **DataEvent** can be received when a magnetic card is swiped if the three conditions are all met:

1) **DeviceEnabled** = TRUE
2) **FreezeEvents** = FALSE
3) **DataEventEnabled** = TRUE.

The track data can be obtained , and the parsed data can also be obtained if **ParseDecodeData** is TRUE.

Support? Yes

2)DirectIO Event

Syntax **void DirectIOEvent (LONG** *EventNumber*, **LONG*** *pData*, **BSTR*** *pString***);**

Parameter Description

EventNumber Event number. Specific values are assigned by the Service Object.

pData Pointer to additional numeric data. Specific values vary by *EventNumber* and the Service Object.

pString Pointer to additional string data. Specific values vary by *EventNumber* and the Service Object.

Remarks Fired by a Service Object to communicate directly with the application.

Support? No

Description The event **DirectIOEvent** is used for some special communication between one SO and an application. In the current, it is implemented incompletely.

3)Error Event

Syntax **void ErrorEvent (LONG** *ResultCode*, **LONG** *ResultCodeExtended*,
LONG *ErrorLocus*, **LONG*** *pErrorResponse*);

Parameter Description

ResultCode	Result code causing the error event. See **ResultCode** for values.
ResultCodeExtended	Extended result code causing the error event. See **ResultCodeExtended** for values.
ErrorLocus	Location of the error. See values below.
PErrorResponse	Pointer to the error event response. See values below.

when **ErrorReportingType** property is MSR_ERT_TRACK, and *ErrorCode* is E_EXTENDED, then *ErrorCodeExtended* contains Track-level status, broken down as follows:

Byte3	Byte2	Byte1	Byte0
Track 4	Track 3	Track 2	Track 1

Remarks Fired when an error is detected and the Control's **State** transitions into the error state.

Support? Yes

4)StatusUpdate Event

Syntax **void StatusUpdateEvent (LONG** *Status*);

The *Status* parameter is for device class-specific data, describing the type of status change.

Remarks Fired when a Control needs to alert the application of a device status change.

Note The MiniMag hardware cannot support the notification of power status change.

Support? No

Description It is not implemented by the SO for the power status cannot be inquired from the MiniMag.