IDentity MaX

Installation and Software User's Guide

Installation and Software User's Guide

SIRIT Corporation
1321 Valwood Pkwy Suite 620
Carrollton, TX 75006

Notices

SIRIT INC.

1321 VALWOOD PKWY, CARROLLTON, TX 75006
372 BAY ST, SUITE 1100, TORONTO, ONTARIO, CANADA M5H 2W9

Trademarks: Product names mentioned in this manual may be trademarks, and they are used for identification only.

Installation and Maintenance Manual
Part Number: IDENTITY MAX Guide-Version P.1.0
Release Date: May 15, 2006
IDentity MaX Installation Manual.doc

⚠ NOTICE

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment to an outlet or circuit different to that which the receiver is connected.
- Consult SIRIT.

⚠ NOTICE

Note: This equipment complies with FCC Part 15.247 and Industry Canada RSS 210 rules. Any changes or modifications not expressly approved by SIRIT could void the user's authority to operate the equipment. To maintain compliance, the IDentity Flex reader must be used with the power supply that was supplied with the reader.

⚠ RF Exposure Warning

To comply with the FCC radiofrequency (RF) Exposure requirements, the antenna(s) used with this device must be installed to provide a minimum separation distance of 1 meter from all persons.

⚠ NOTICE

For PLUGGABLE EQUIPMENT, the socket/outlet shall be installed near the equipment and shall be easily accessible. For PERMANENTLY CONNECTED EQUIPMENT, a readily accessible disconnect device shall be incorporated into the fix

⚠ TRANSPONDER NOTICE

WARNING! Danger of explosion if battery is incorrectly replaced. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.
ed wiring.

Table of Contents

1.0 Overview

The ID*entity* MaX reader is an RFID device capable of reading EPC Class 0, Class1, and Class 1 Gen 2 RFID tags, and processing the data for a gated access type application. Generally, the reader is configured, upon installation, to work with specific inputs and outputs to interface with customer systems including access panels, gates and visual indicators. All system configurations are accomplished using either a PC based application or a web based user interface.

2.0 Communications

The ID*entity* MaX reader supports a variety of communication interfaces and protocols.

2.1 Ethernet

The reader Ethernet port is fully 802.3 compatible with Auto 10/100 and Auto MDIX functionality. It also supports Power over Ethernet and is classified as a Class 0 powered device. The Ethernet is the primary interface for the web based user interface.

2.2 USB

The USB port of the ID*entity* MaX is USB 2.0 compliant and is accessed via a type B connector on the bottom of the unit. The USB port is the primary communication method for the PC based user interface.

2.3 Host Serial

The Host serial port supports a variety of protocols and signaling, including RS-232, RS-422, and RS-485. It is primarily used to transfer RFID tag information to a host or maintenance PC.

2.4 Wiegand Port

The Wiegand port is a simplex type communication from the reader to a host or access panel compliant to the SIA 26-bit standard but also supports extended formats.

3.0 Installation and Connections

3.1 Installation Overview

This section lists the steps required to install and test an ID*entity* MaX system comprised of a single reader. The installer should have a successful installation by completing each step before moving on to the next.

The Suggested installation steps are:

1. Gather all necessary tools and materials. See section 3.2.
2. Install reader in desired location using provided hardware. See section 3.3.
3. Complete power and communication connections. See section 3.4.
4. Perform system verification and configuration using either the PC based interface or the web based interface. See sections 4.0 or 5.0.

3.2 Tools and Materials

The installation uses the following materials provided by Sirit.

1. ID*entity* MaX reader
2. Interface cable (Available from Sirit, but not standard)
3. 48 VDC power supply
4. IEC to three prong power cable
5. Universal mounting bracket

The only tools required will be standard hand tools.

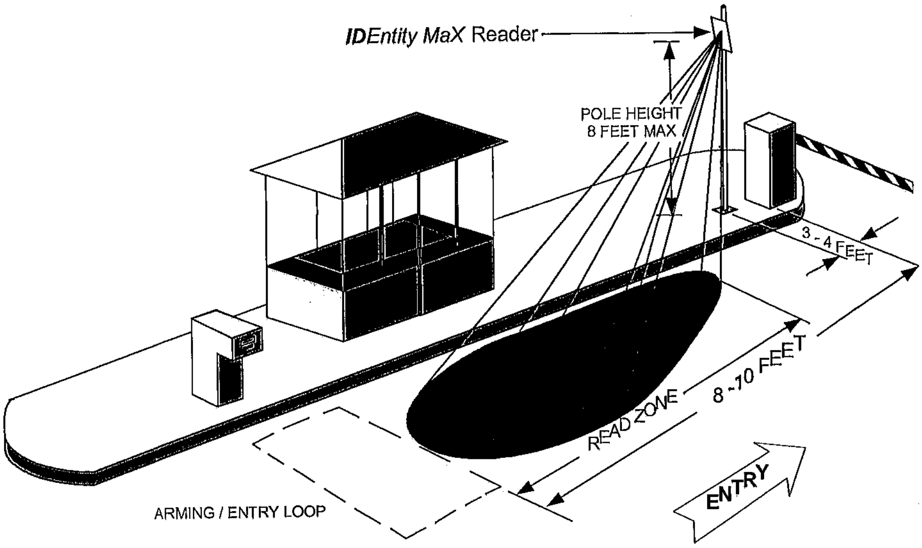
However, the interface cable will require additional tools and material to install Ethernet, RS-232, I/O, and power connectors.

3.3 Reader Installation

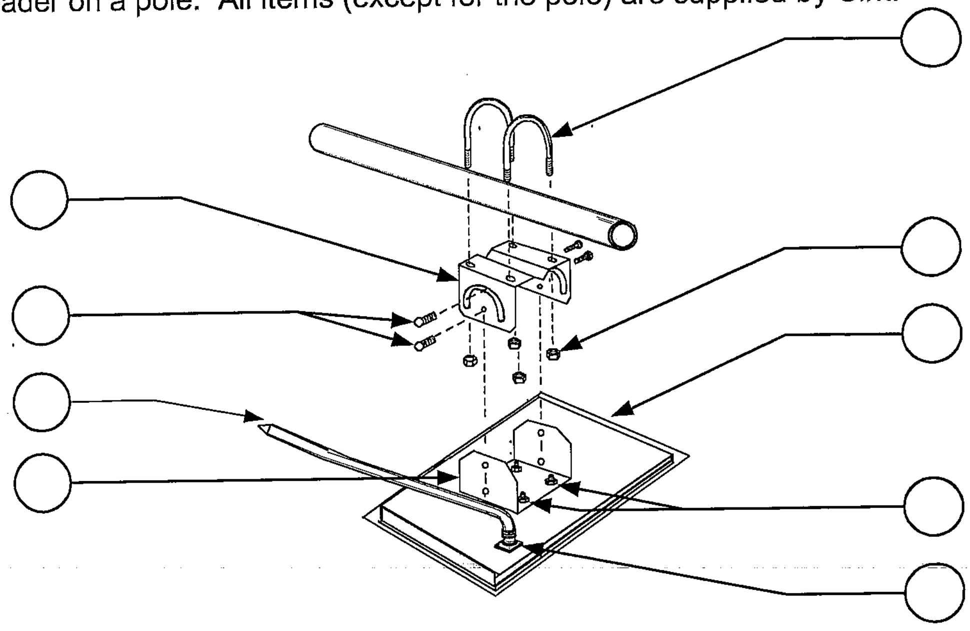
The ID*entity* MaX reader may be mounted on vertical poles or horizontal pipes utilizing the supplied universal mounting bracket (UMB). Note that the U-bolts provided with the UMB are designed to be used with a 1.5-inch pole. The reader requires the supplied interface cable be attached to the Amphenol connector on the rear of the reader. The cable may be clamped to walls with clips or run through appropriate conduits.

The location and angle of the reader define its read zone. Establish the location and mounting method in conjunction with the Site Planning Guide prior to installation. It is essential that the reader be installed in the correct orientation and location for accurate transponder reads.

The figure below illustrates a typical parking access installation. This is an example only and should not be substituted for the installation setup.



The Figure below shows the method and hardware required to install the ID*entity* MaX reader on a pole. All items (except for the pole) are supplied by Sirit.



Item	Description
1	U-bolts (for 1.5 inch pole), reader mounting bracket, 2 per bracket
2	UMB bracket half, conduit mount side
3	UMB bracket half, reader mount side
4	Interface cable
5	Bracket pinning bolts, 2 required
6	Nuts for U-bolts, 4 per bracket
7	ID*entity* MaX reader
8	Bracket mounting nuts, ¼-20, 4 per bracket
9	Amphenol connector

3.4 Communication and Power Connections

The interface cable is comprised of approximately 20 feet of a 30 conductor (15 pair) twisted pair cable with an Amphenol connector on one end, the mate of the connector on the rear of the reader. The opposite end will have all communications, power, and I/O connections as loose wire ends. The appropriate communication connectors will be required for configuration and remote communication. See Appendix B for proper cable pin outs.

To attach the cable to the reader, carefully align the single connector on the interface cable with the Amphenol connector on the reader. With minimal pressure, screw the outer housing of the cable connector onto the reader connector. The connectors should automatically lock in place when they are properly attached.

Next, connect the communications connectors to their mates on the user systems. The Ethernet connection is required to use the web based interface, the USB interface is required to use the PC based interface, and the serial connection is required for host attachment.

Finally, plug the IEC power cable into the power supply, and the DC side of the power supply into the connector on the interface cable. Plug the three prong connector into a 120V, 60Hz AC power socket.

4.0 PC based User Interface

The PC based application maxgui.exe is the primary user interface for system configuration and monitoring. It is designed to use the USB connection for communication to the maintenance PC. Microsoft .NET framework version 2.0 needs to be installed on the maintenance PC in order to run the GUI. The .NET framework installation file is also included as part of the documentation package for the ID*entity* MaX.

To install the .NET framework, copy dotnetfx.exe to the maintenance PC and run the application. Follow all prompts to install.

Because this GUI is based on xml formatted commands, there is a folder called "allxmlcmd" that needs to be copied to the c:\ directory. All xml commands are in this folder.

Connect the USB port of the interface cable to the USB port of the PC. Verify that the reader has power and is operational. At the initial installation, a dialog box may appear asking for the USB driver. A copy of this driver has been included with the IDentity MaX software package. Run the interface program maxgui.exe.

Looking at the main screen of the interface, disregard the "Test Hardware" and "Flash Memory Operation" tabs. These were used by the developer for test and debug and will be removed at a later revision. All sub menus will have a "Confirm" button. This button needs to be clicked after making any changes to the configuration in that particular menu. Otherwise, the changes will not be written into the Random Access Memory on the reader.

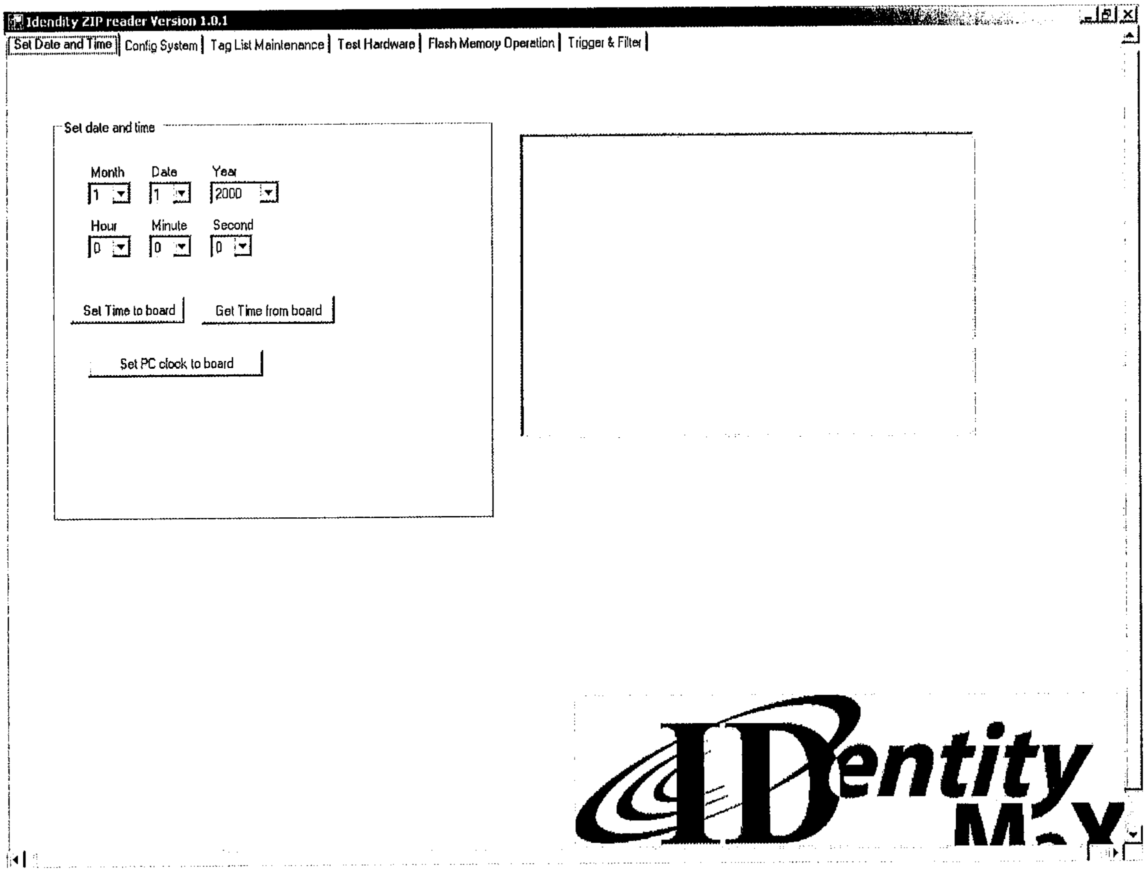
4.1 Setting reader date and time

The ID*entity* MaX reader will be configured during the manufacturing process for the date and time based upon US central time zone (GMT-6). The "Set Date and Time" tab allows the user to change the reader time and date.

"Get Time and Date" is used to view current reader date and time.

To change the time and date, simply use the drop down options provided for each parameter. Hours are in 24 hour format. Press "Set Time to board" to overwrite values currently in the reader.

Though not necessary, it is sometimes desirable to synchronize date and time on the reader and maintenance PC. "Set PC clock to board" will change the reader clock to the same time and date as the PC clock.

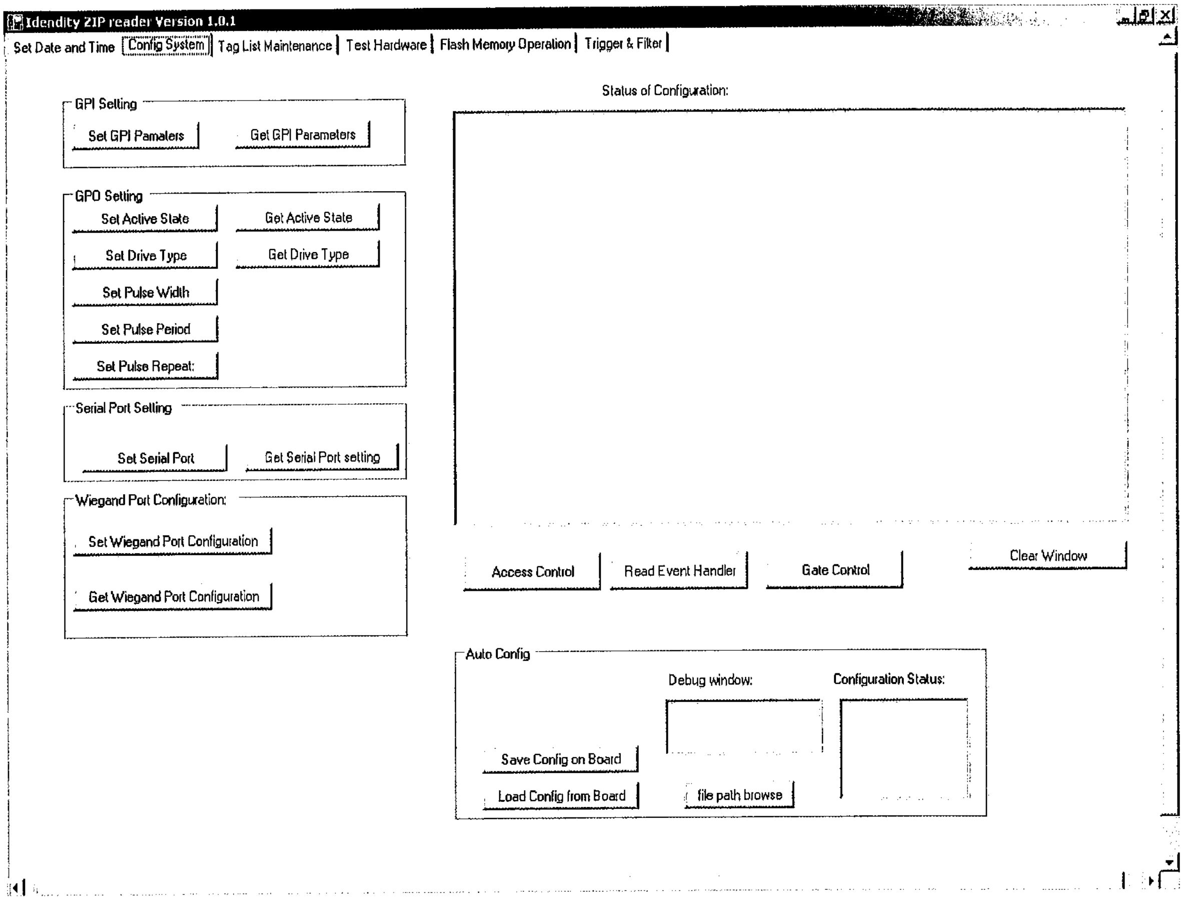


Month 1 Date 1 Year 2000

Hour 0 Minute 0 Second 0

4.2 Configuring the System

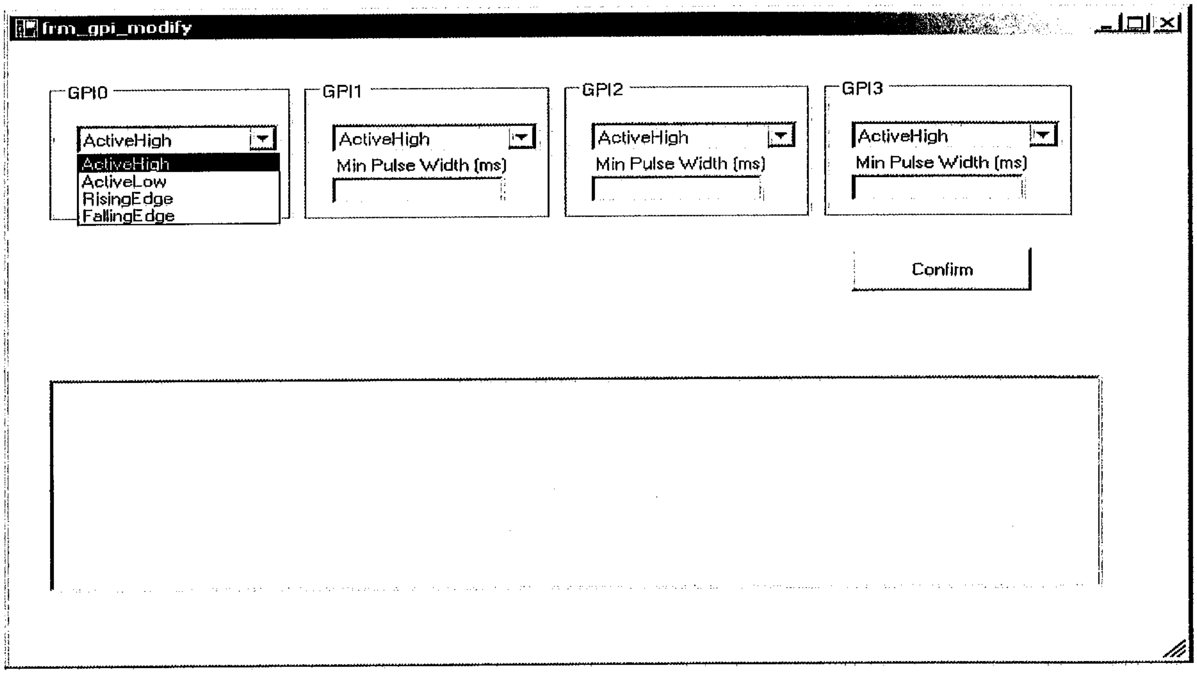
The Configuration System menu is used to verify and change all parameters related to system interface. This includes the general purpose inputs and outputs, the serial port, Wiegand port, gate control, and translation table exception handling.



4.2.1 General Purpose Inputs (GPI)

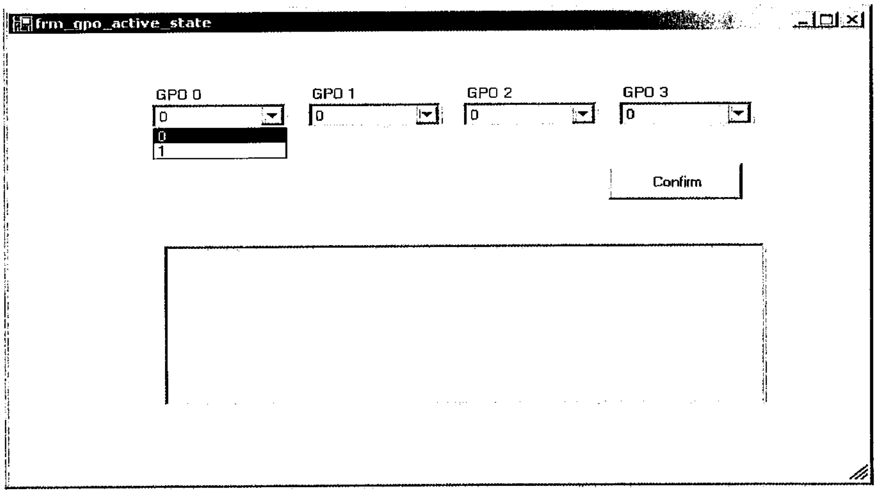
To view current GPI settings use "Get GPI Parameters."

To change GPI settings, use "Set GPI Parameters"
Each input can be set to an active high, active low, or edge triggered event.

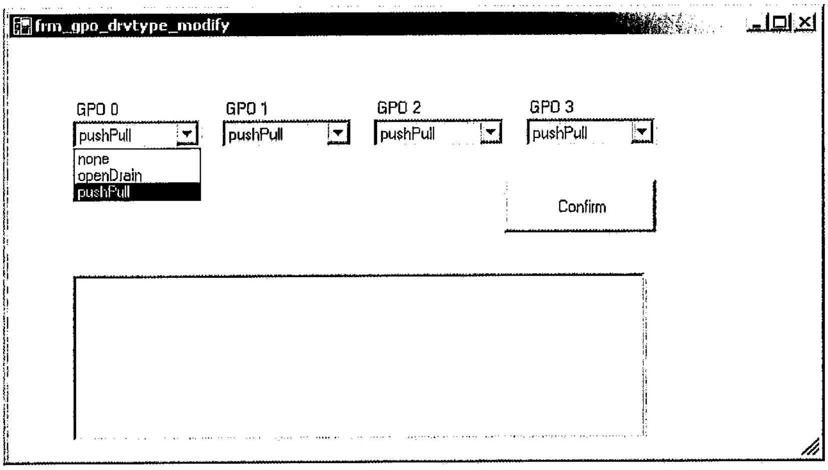


4.2.2 General Purpose Outputs (GPO)

The four general purpose outputs have several parameters that can be configured for a variety of applications.
To view the current values of Active state and Drive type, use the respective button.
Active state can be set to active low (0) or active high (1).



There are three options for output drive state. Open drain, where the user will need to supply pull up resisters, Push pull, and none for a tri-stated output.



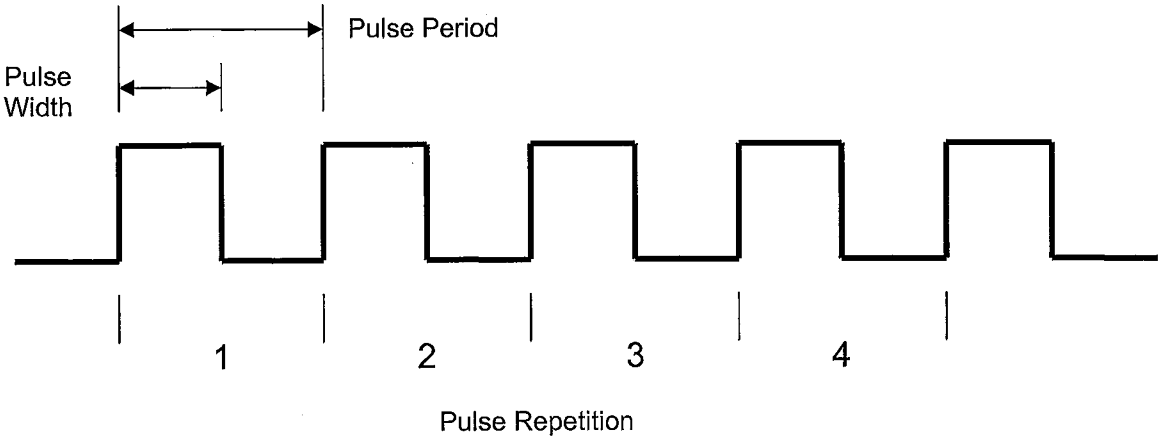
GPO 0 pushPull

GPO 1 pushPull

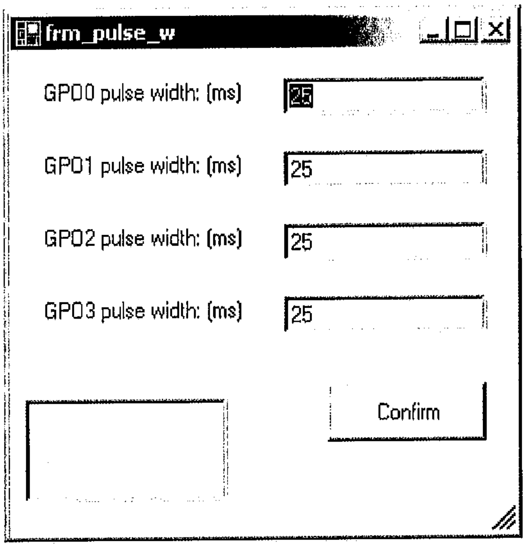
GPO 2 pushPull

GPO 3 pushPull

Each output waveform is specified with three parameters: Pulse width, pulse period and pulse repetition. The width is the time that the signal is active. Period is the time from rising edge to rising edge (Active high state) or falling edge to falling edge (Active low state). Pulse repeat is the number of times the active signal is repeated. The relationship between the three is shown below:



All four outputs can be created using the three menus: "Set Pulse Width", "Set Pulse Period", and "Set Pulse Repeat".

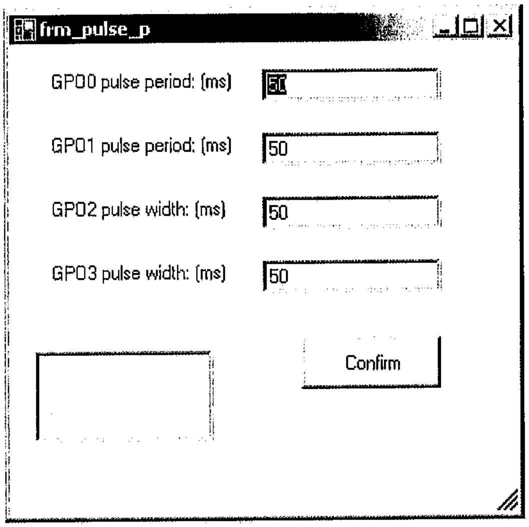

frm_pulse_w

GPO0 pulse width: (ms) 25

GPO1 pulse width: (ms) 25

GPO2 pulse width: (ms) 25

GPO3 pulse width: (ms) 25

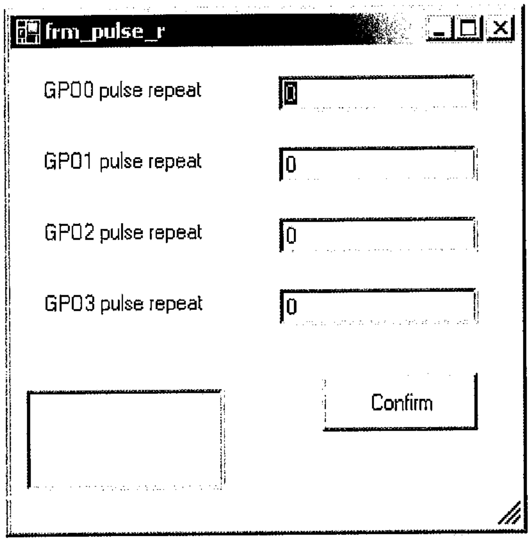

frm_pulse_p

GPO0 pulse period: (ms) 50

GPO1 pulse period: (ms) 50

GPO2 pulse width: (ms) 50

GPO3 pulse width: (ms) 50


frm_pulse_r

GPO0 pulse repeat 0

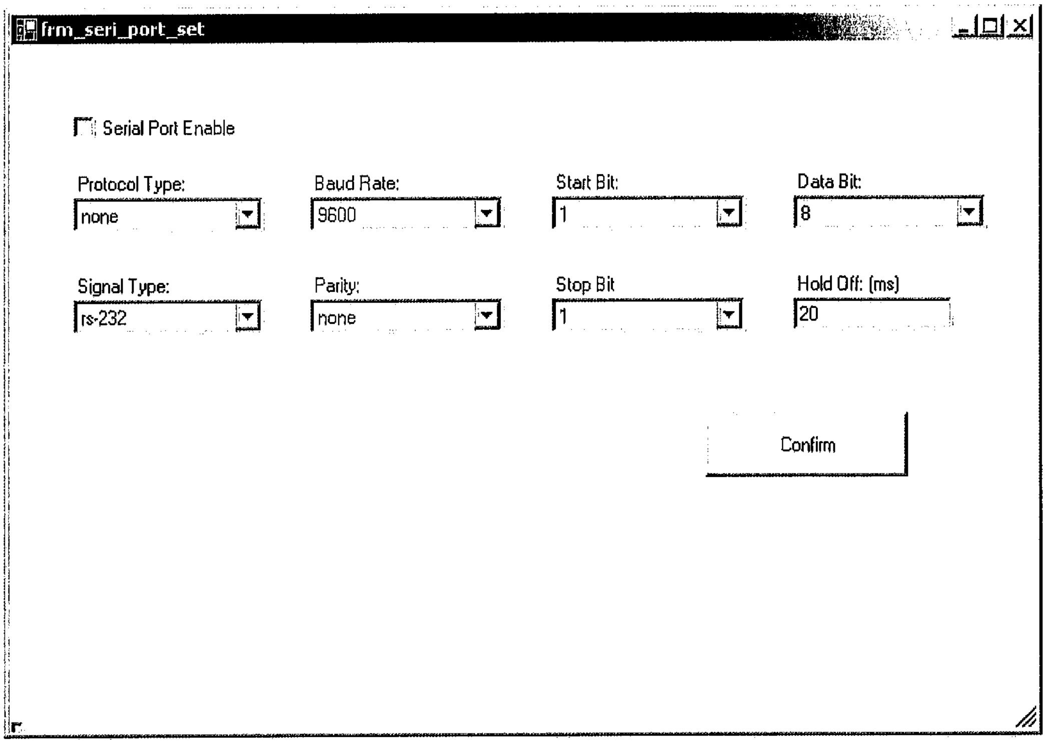
GPO1 pulse repeat 0

GPO2 pulse repeat 0

GPO3 pulse repeat 0

4.2.3 Serial Port Configuration

The serial port supports a variety of protocols and signaling. To view the current serial port set up, use "Get Serial Port setting". To change any of the parameters, use "Set Serial Port",



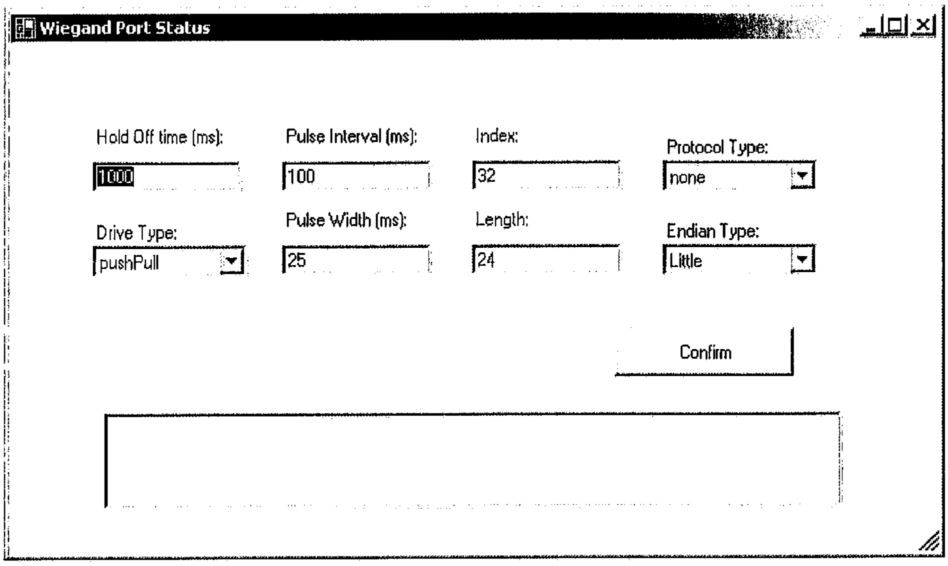
☐ Serial Port Enable

Protocol Type: none
Baud Rate: 9600
Start Bit: 1
Data Bit: 8
Signal Type: rs-232
Parity: none
Stop Bit 1
Hold Off: (ms) 20

To enable the host port, check the enable box. This serial port supports WPS and ACOM protocols, as well as RS-232, RS-422, and RS-485 signaling. To change any of the parameters, simply choose the value from the pull down menus. Hold off time refers to the time that is inserted between output data.

4.2.4 Wiegand Port Configuration

The Wiegand port is a simplex type communication compliant to the SIA 26-bit standard as well as extended formats.
Use "Set Wiegand Port Configuration" to specify Wiegand parameters.



Hold Off time (ms): 1000

Pulse Interval (ms): 100

Index: 32

Protocol Type: none

Drive Type: pushPull

Pulse Width (ms): 25

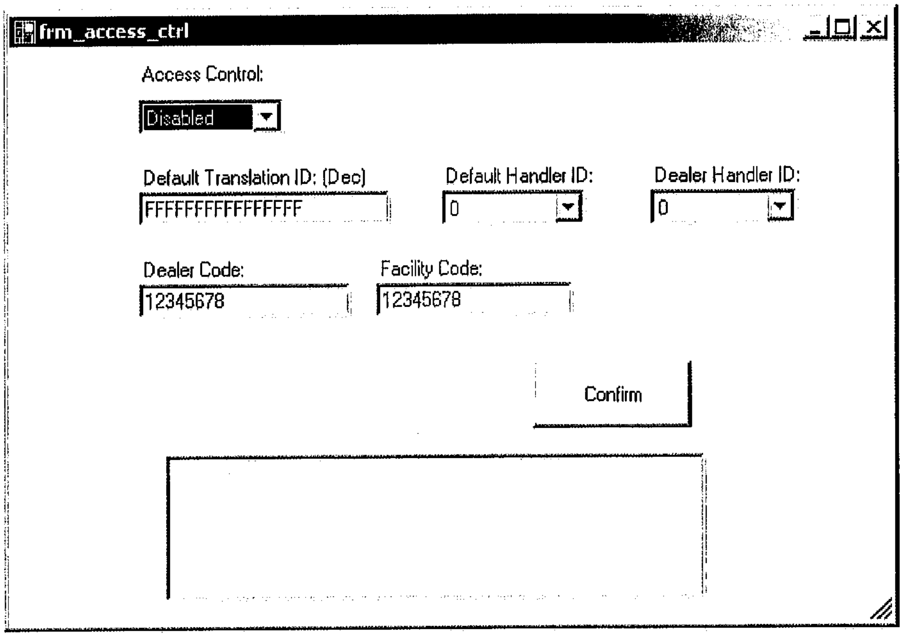
Length: 24

Endian Type: Little

Transmission on the Wiegand interface can only occur when a corresponding Host Upload Event has been configured and satisfied. As before, the pulse width is the time that a data pulse is active for valid data and the pulse interval is the time between those pulses. Index is the offset (in bits) within the translated tag data where the serialization information will begin. Length specifies the length of the data packet, not including any protocol overhead. Both Parity and Checksum protocols are supported in addition to raw data format. Endian type denotes whether most significant (Big) or least significant (Little) will be transmitted first.

4.2.5 Access Control Configuration

The ID*entity* Max reader uses a translation table to reference all actions and ID associated with specific tags. Access control is used to determine the action(s) taken for any tag read. Use the "Access Control" button to start configuring these parameters.



Access Control: Disabled

Default Translation ID: (Dec) FFFFFFFFFFFFFFFF

Default Handler ID: 0

Dealer Handler ID: 0

Dealer Code: 12345678

Facility Code: 12345678

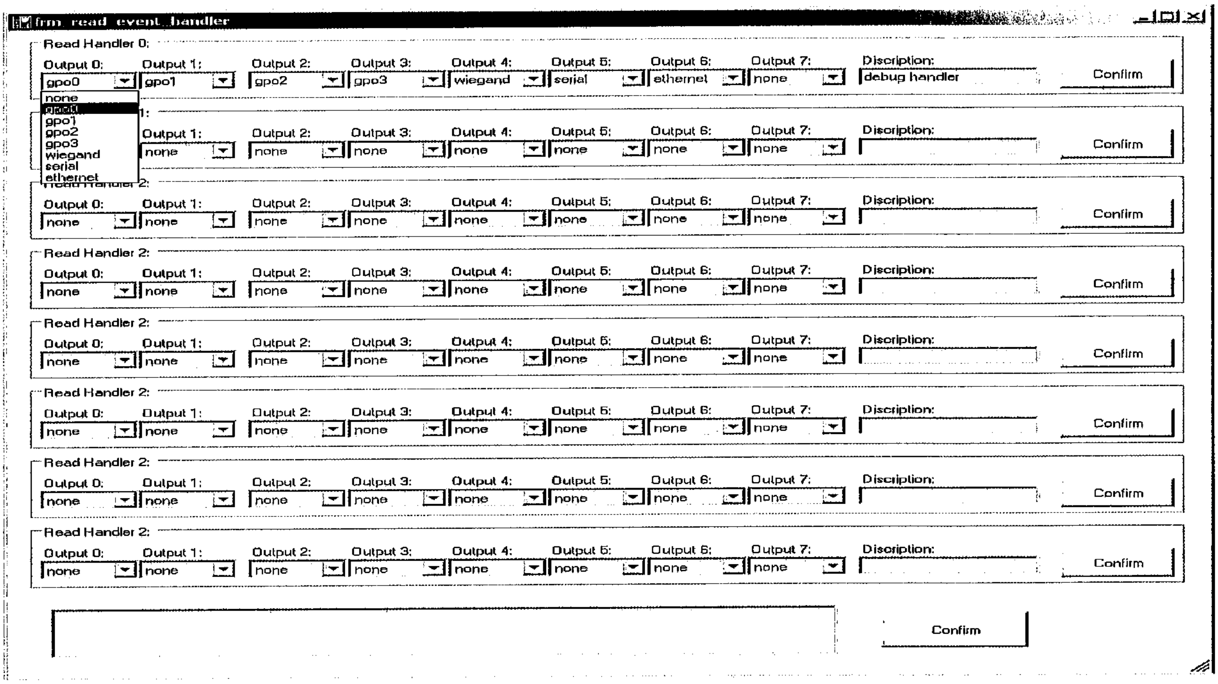
To enable access control features, select "enabled" from the pull down menu. The default translation ID is the value the reader will enter in the log and is set by the user to any 16 digit or less value. The default handler indicates the reporting action(s) the system will take if an unknown tag is read. See section 4.2.6 for a full list of options available.

The dealer and facility codes are programmed by Sirit during the manufacturing process and are unique to every Sirit customer. However, the Dealer code and the Dealer handler ID are not implemented in this version of the ID*entity* MaX reader.

4.2.6 Read Event Handler

The read event handler allows the user to specify a variety of actions to be taken when a tag is read and looked up in the translation table. In the event that an unknown tag is read, the default event handler specified in the Access Control menu will be used. With this menu, the user will be able to turn on a warning lamp, send a warning using the host serial port, raise the gate, or all of these.

To configure the handler, select the desired action from the pull down menu.

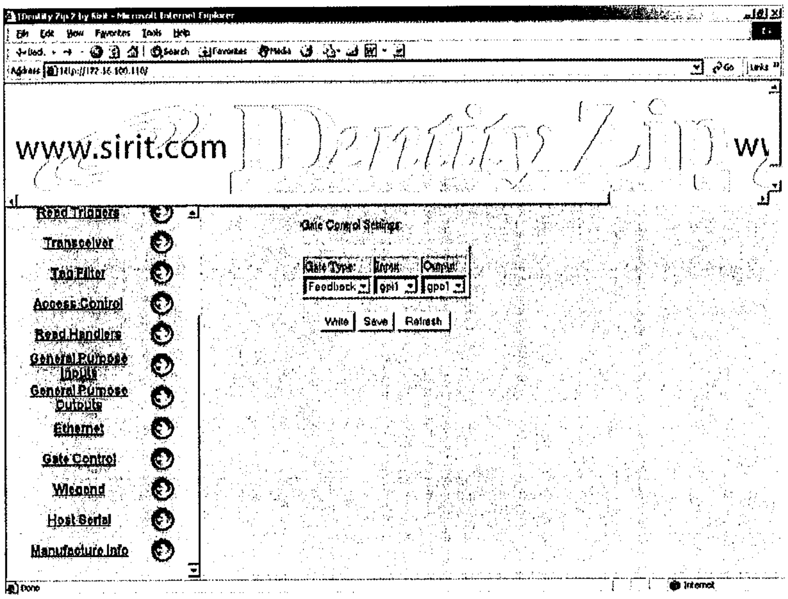


frm_read_event_handler

Read Handler	Output 0:	Output 1:	Output 2:	Output 3:	Output 4:	Output 5:	Output 6:	Output 7:	Discription:	
Read Handler 0:	gpo0	gpo1	gpo2	gpo3	wiegand	serial	ethernet	none	debug handler	Confirm
1:		none	none	none	none	none	none	none		Confirm
Read Handler 2:	none	none	none	none	none	none	none	none		Confirm
Read Handler 2:	none	none	none	none	none	none	none	none		Confirm
Read Handler 2:	none	none	none	none	none	none	none	none		Confirm
Read Handler 2:	none	none	none	none	none	none	none	none		Confirm
Read Handler 2:	none	none	none	none	none	none	none	none		Confirm
Read Handler 2:	none	none	none	none	none	none	none	none		Confirm

4.2.7 Gate Control

The Gate Control menu is used to select the gate type (normal or feedback), the control input should synchronization be used, and the control output. All parameters are changed using the respective pull down menus.



Address http://172.16.100.110/

Gate Control Settings

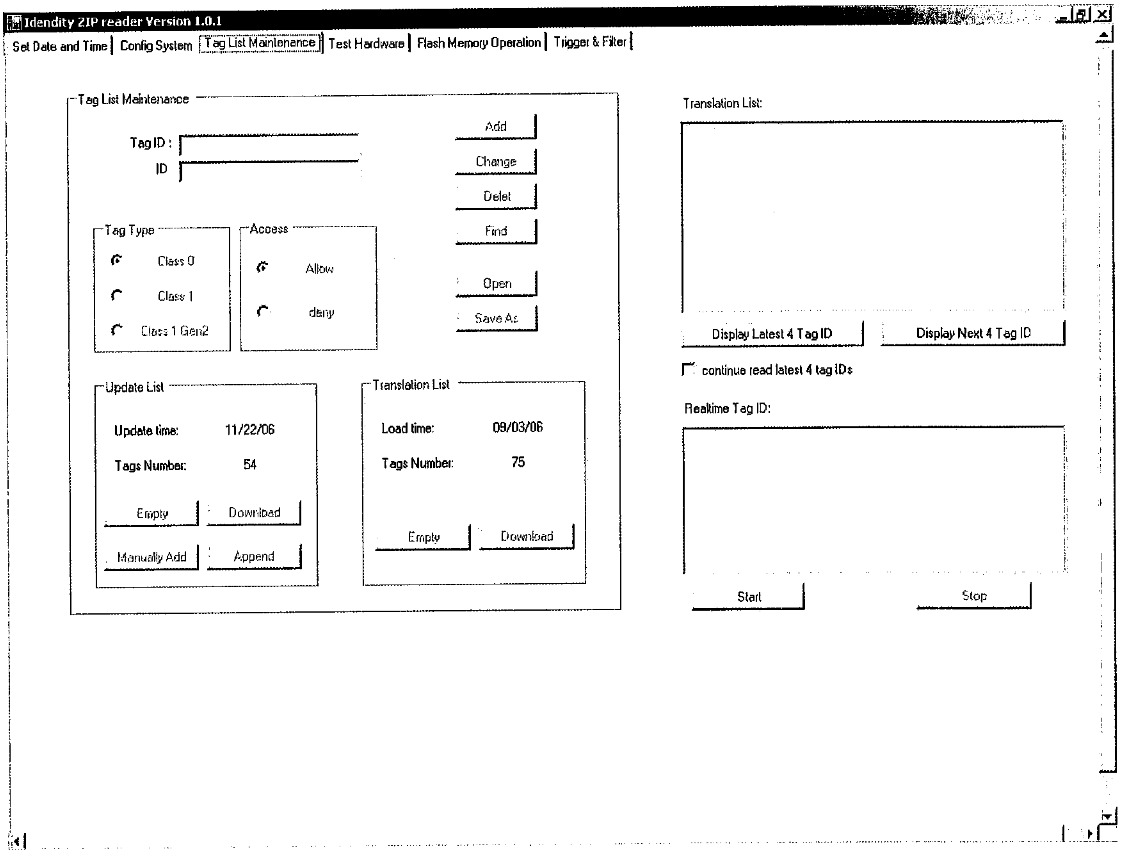
Gate Type:	Input:	Output:
Feedback	gpi1	gpo1

4.2.7 AutoConfig

All configuration changes made using this application and subsequently "confirmed" are stored only in the Random Access Memory (RAM) of the reader and will be lost upon power down. Should the user desire to create a permanent configuration, press the "Save Config on Board" button. This will instruct the reader to copy the contents of the configuration to Flash memory so that it can be retrieved upon reader power up or by using the "load" command. The "Load Config from Board" command will load the set up currently in Flash memory into the on board RAM. Note that loading the configuration into RAM will not change the settings in this application. The "file path browse" button is a debug feature only and should not be used.

4.3 Tag List Maintenance

The Tag List Maintenance tab is used to view events monitored by the ID*entity* MaX reader. The only two active windows in this section are the "Translation List" and the "Realtime Tag ID".



Tag Type: ● Class 0 ○ Class 1 ○ Class 1 Gen2

Access: ● Allow ○ deny

Update List

Update time: 11/22/06

Tags Number: 54

Translation List

Load time: 09/03/06

Tags Number: 75

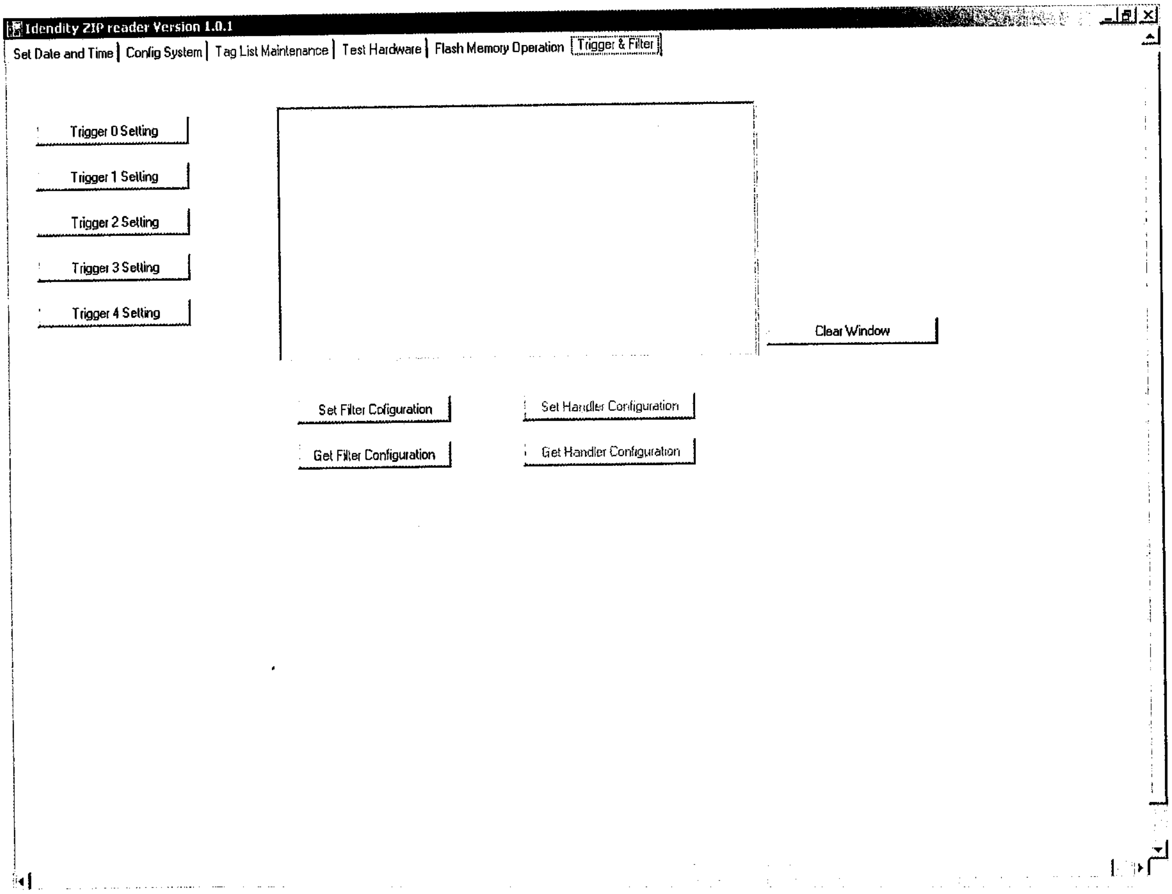
□ continue read latest 4 tag IDs

The "Display Latest 4 Tag ID" button will show the most recent four tags read. The source of information for this window is the USB port on the controller.
The "Display Next 4 Tag ID" will show the next most recent four tags read. Repeated uses of "Display Next 4 Tag ID" will in essence allow the user to scroll through all read activity four tags at a time.
The "continue read latest 4 tag IDs" feature is not used in this version of this application.

Hit the Start button to display all tag reads as they happen. Use the Stop button to discontinue the display. Note that this will not disable reading of tags. The source of information for the "Realtime TagID" window is the host serial port.

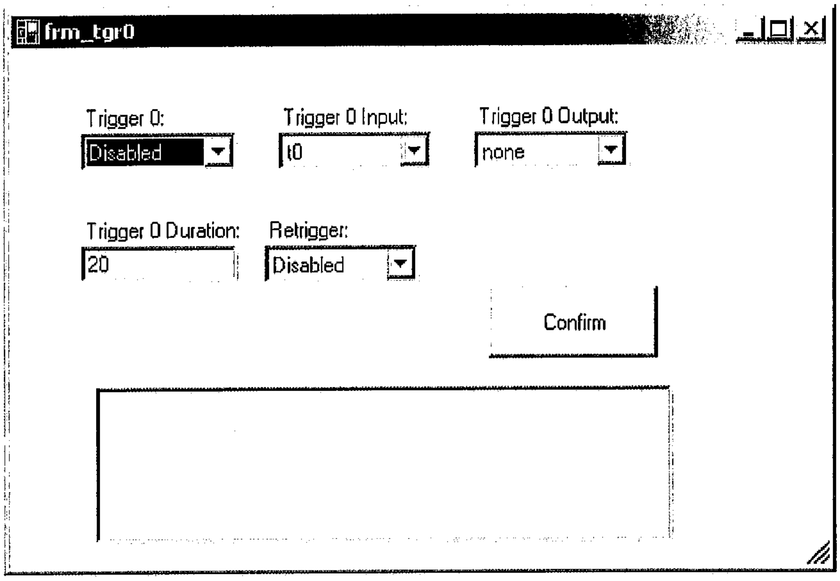
4.4 Trigger and Filter

The Trigger and Filter tab allows the user to set specific events which will initiate RFID transactions and to filter incoming data to reduce the amount of redundant information being reported back to the user.



4.4.1 Trigger Settings

There are five independent triggers that are given to the user to use. Use the "Trigger Settings" menu to configure the respective triggers to be used. All trigger event parameters are configured using the pull down menus.



Trigger 0: Disabled

Trigger 0 Input: t0

Trigger 0 Output: none

Trigger 0 Duration: 20

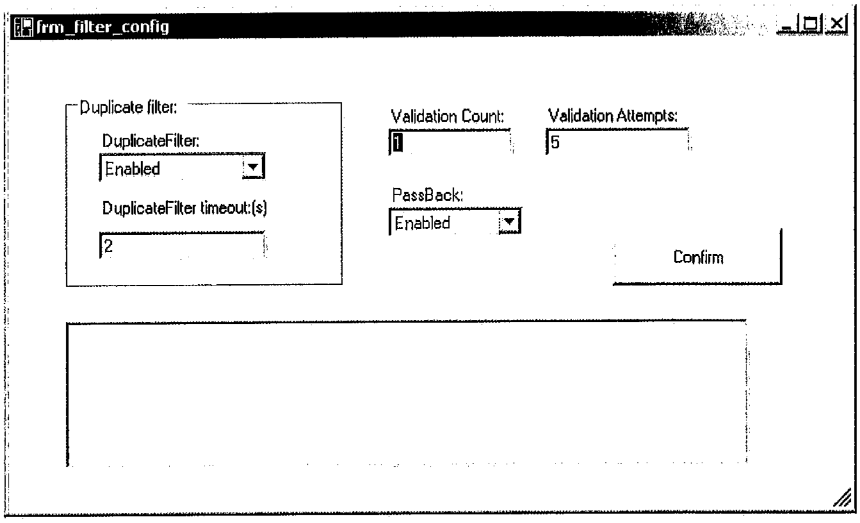
Retrigger: Disabled

Each trigger can have the input linked to a general purpose input or to the internal timer (t0). The Trigger duration determines the amount of time the reader will attempt an RFID transaction before going inactive..
The resultant output will be routed to the general purpose output selected here. By enabling retrigger, all subsequent occurances of this event can be monitored.

4.4.2 Filter Configuration

To view the current filter parameters, use "Get Filter Parameters".

To edit or create a filter, use "Set Filter Parameters"



frm_filter_config

Duplicate filter:

DuplicateFilter: Enabled

DuplicateFilter timeout:(s) 2

Validation Count: 1

Validation Attempts: 5

PassBack: Enabled

The duplicate filter will remove all subsequent instances of a particular tag after the initial read. This can greatly reduce the amount of data reported back to the user caused by a tag remaining in the read zone for a longer than average time. This will not suspend the actual tag read for the time out period, but rather will cease duplicate data reporting only.

Validation is designed to discard all erroneous incoming data prior to reporting and processing. The validation count is the number of times a tag must be successfully read to be considered valid. The validation attempt is the number of tries the reader will make in order to obtain a validation count. Naturally, the value in validation attempts needs to be greater than validation count.

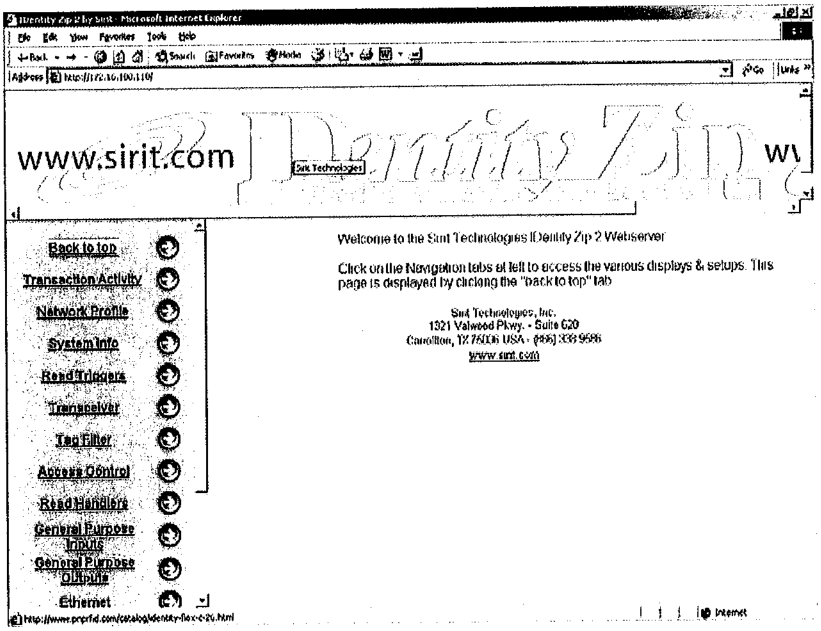
The passback feature will allow the time stamp on all duplicate tag reads to be updated for each read, extending the timeout. If passback is enabled, a tag must leave the read zone for the duration of the duplication filter timeout period.

5.0 Web based User Interface

The second user interface included with the ID*entity* MaX reader is a web based interface that is actually built into the control hardware. There is no extra application needed to use this interface. This web based interface is designed to be used with the Ethernet channel rather than the USB port.

The one requirement to use the interface is that the ip address of the reader must be known. If used in a networked environment with a DHCP server, simply use the network admin application to determine the reader ip address.

To begin, open an internet browser window. This GUI has been tested only with Microsoft's Internet Explorer v6. However, it is very likely that it will function properly with other browsers such a Mozilla or Firefox. Enter the reader ip address previously determined into the URL line of the browser and press Enter. The title screen of the User Interface should appear. All other menus are brought up using the links along the left side. Simply click on the text or arrow to navigate to the desired menu. All sub menus will have "write", "save" and "refresh" buttons. The write button will update the configuration in the RAM on the reader. The Save button will update the configuration in the Flash memory on the reader, thus making it permanent. The Refresh button will refresh the screen of the maintenance PC with any new configuration parameters.



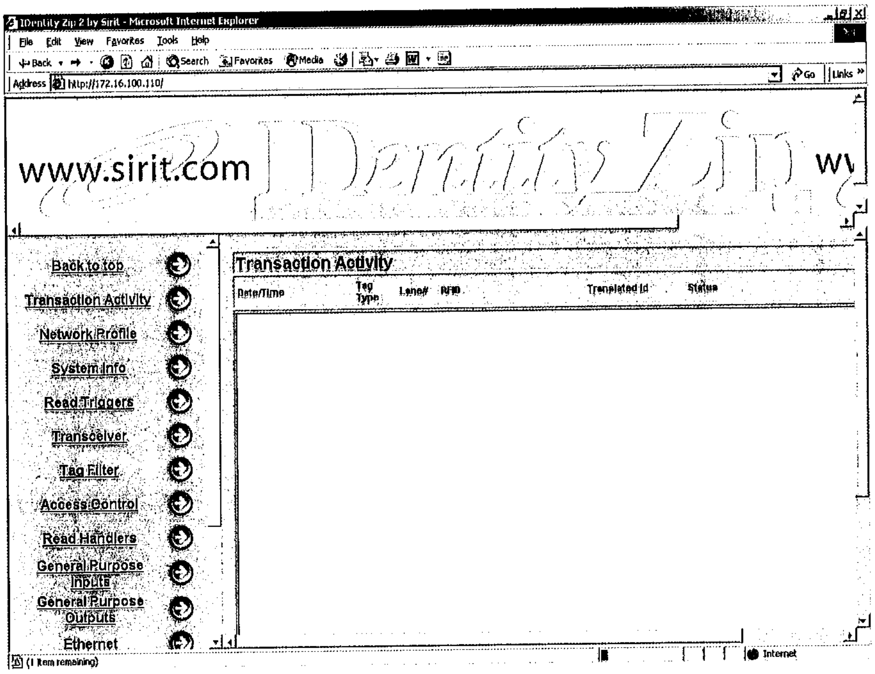
Welcome to the Sirit Technologies IDentity Zip 2 Webserver

Click on the Navigation tabs at left to access the various displays & setups. This page is displayed by clicking the "back to top" tab

Sirit Technologies, Inc.
1321 Valwood Pkwy. - Suite 620
Carrollton, TX 75006 USA - (866) 338 9586
www.sirit.com

5. 1 Transaction Activity

This is a display only screen showing a list of all tag reads in realtime.

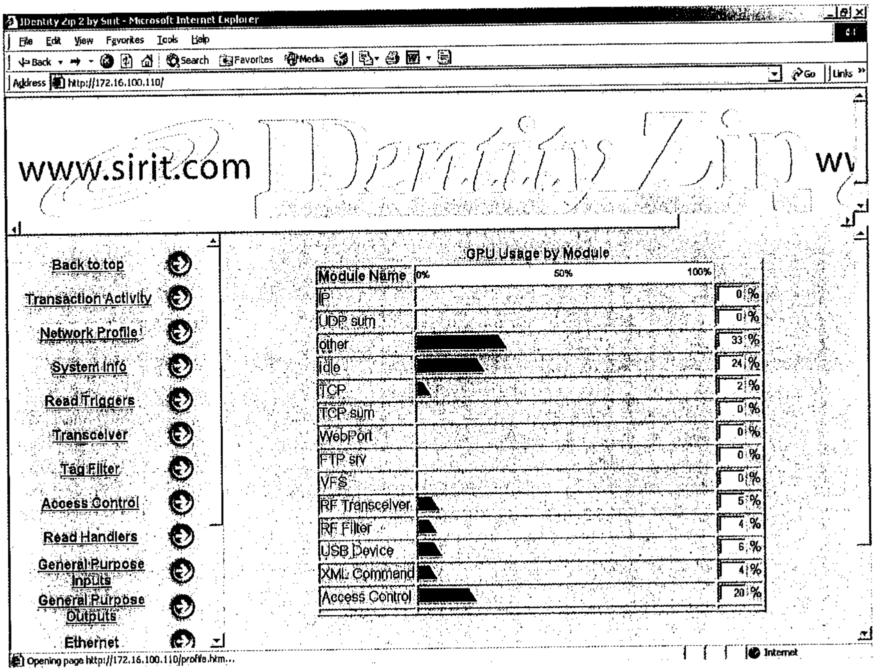


Transaction Activity

Date/Time	Tag Type	Lane#	RFID	Translated Id	Status

5.2 Network Profile

Another display only screen showing the loading upon the internal CPU of the reader by the various software and hardware modules.

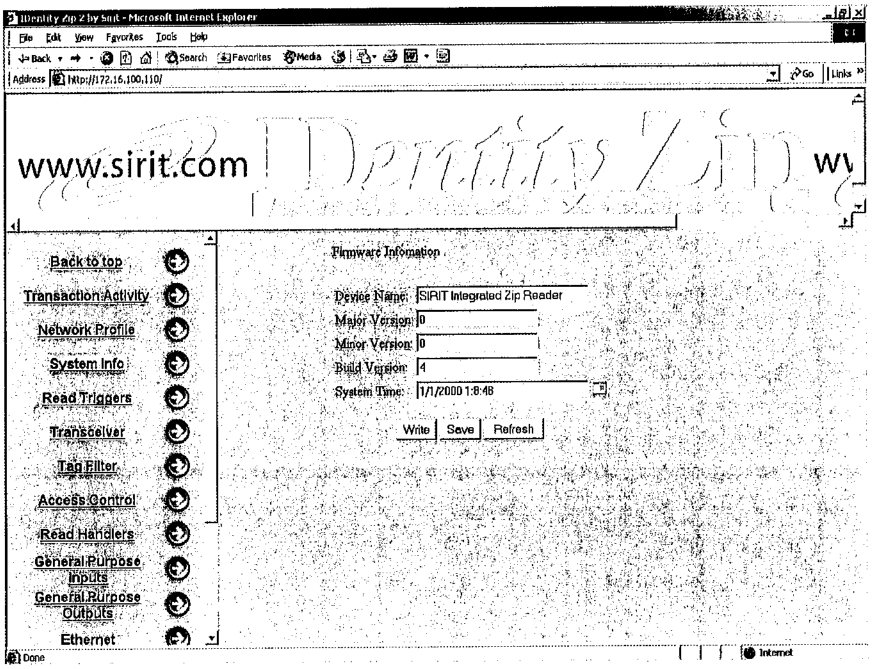


CPU Usage by Module

Module Name	0% 50% 100%	
IP		0 %
UDP sum		0 %
other		33 %
idle		24 %
TCP		2 %
TCP sum		0 %
WebPort		0 %
FTP srv		0 %
VFS		0 %
RF Transceiver		5 %
RF Filter		4 %
USB Device		6 %
XML Command		4 %
Access Control		20 %

5.3 System Info

This display screen will show the hardware and firmware revisions of this particular IDe*ntity* MaX reader. The internal system date and time are also shown.



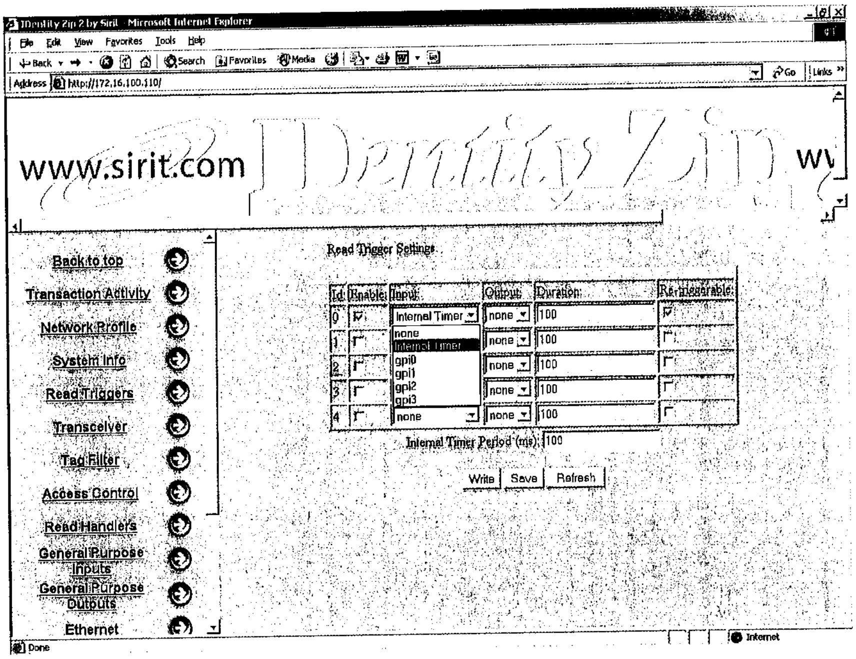
Address http://172.16.100.110/

Firmware Information

Device Name: SIRIT Integrated Zip Reader
Major Version: 0
Minor Version: 0
Build Version: 4
System Time: 1/1/2000 1:8:48

5.4 Read Triggers

The Read trigger menu allows the user to set specific events that will initiate RFID transactions. There are five independent triggers that are given to the user to use. All trigger event parameters are configured using the pull down menus.



Read Trigger Settings

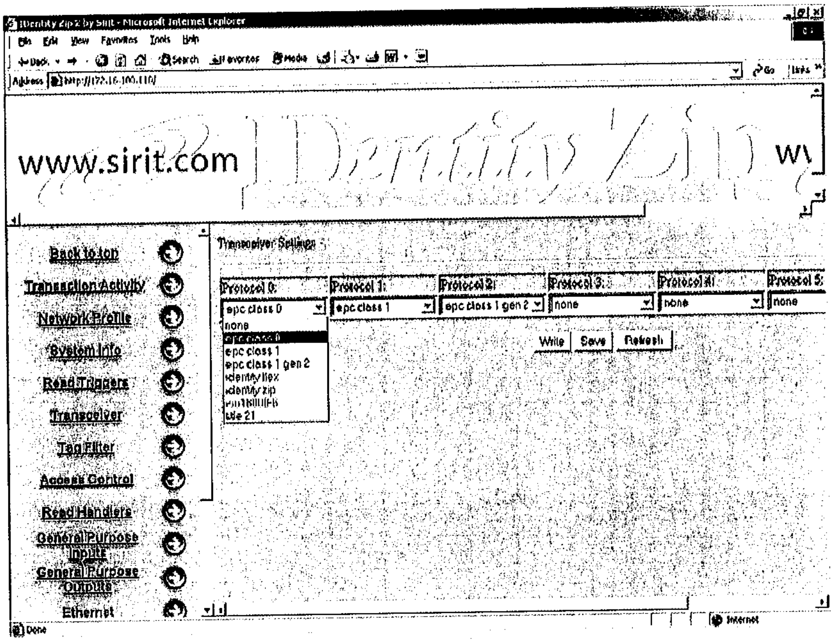
Id:	Enable:	Input:	Output:	Duration:	Re-triggerable:
0	☑	Internal Timer	none	100	☑
1	□		none	100	□
2	□		none	100	□
3	□		none	100	□
4	□	none	none	100	□

Internal Timer Period (ms): 100

Each trigger can have the input linked to a general purpose input or to the internal timer (t0). The minimum valid signal pulse width is set using the Trigger duration parameter.
The resultant output will be routed to the general purpose output selected here. By enabling retrigger, all subsequent occurrences of this event can be monitored.

5.5 Transceiver

The ID*entity* MaX reader is capable of reading a wide variety of UHF tag protocols;,. The "Transceiver" menu allows the user to specify which tag protocols will be used.

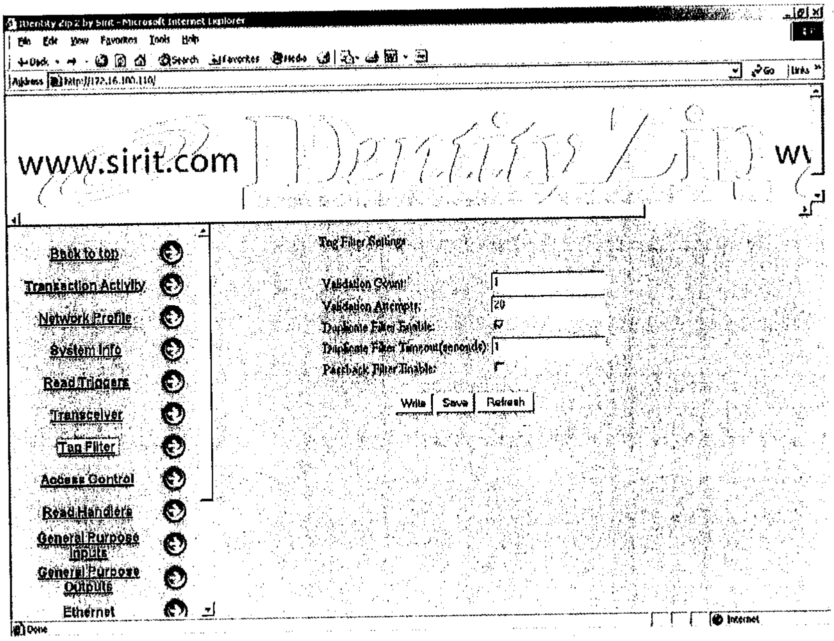


Transceiver Settings

Protocol 0: epc class 0
Protocol 1: epc class 1
Protocol 2: epc class 1 gen 2
Protocol 3: none
Protocol 4: none
Protocol 5: none

5.6 Tag Filter

The Tag Filter menu allows the user to filter incoming data to reduce the amount of redundant information being reported back to the user.



Tag Filter Settings

Validation Count: 1
Validation Attempts: 20
Duplicate Filter Enable: ☑
Duplicate Filter Timeout(seconds): 1
Passback Filter Enable: □

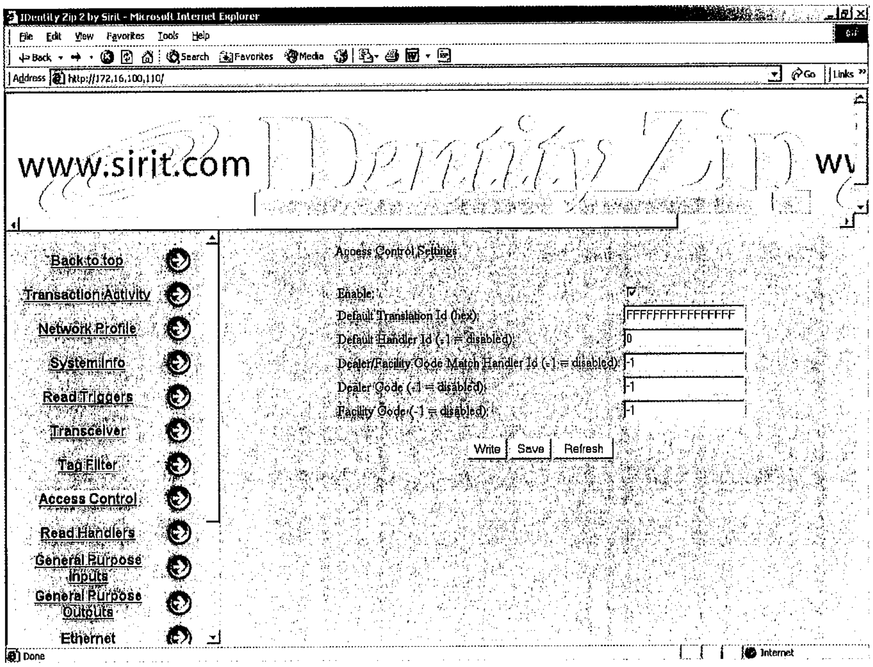
The duplicate filter will remove all subsequent instances of a particular tag after the initial read. This can greatly reduce the amount of data reported back to the user caused by a tag remaining in the read zone for a longer than average time. This will not suspend the actual tag read for the time out period, but rather will cease duplicate data reporting only.

Validation is designed to discard all erroneous incoming data prior to reporting and processing. The validation count is the number of times a tag must be successfully read to be considered valid. The validation attempt is the number of tries the reader will make in order to obtain a validation count. Naturally, the value in validation attempts needs to be greater than validation count.

The passback feature will allow the time stamp on all duplicate tag reads to be updated for each read, extending the timeout. If passback is enabled, a tag must leave the read zone for the duration of the duplication filter timeout period.

5.7 Access Control

The ID*entity* Max reader uses a translation table to reference all actions and ID associated with specific tags. Access control is used to determine the action(s) taken for any tag read.



Access Control Settings

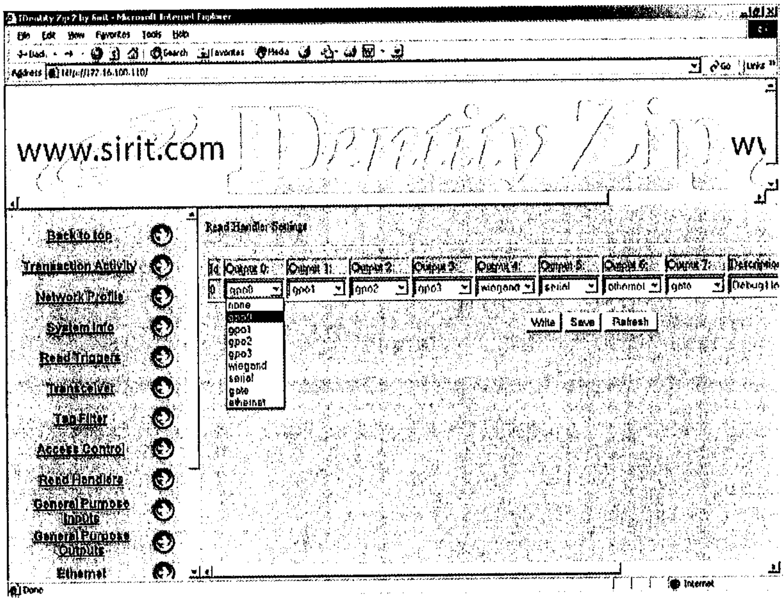
Enable: ☑
Default Translation Id (hex): FFFFFFFFFFFFFFFF
Default Handler Id (-1 = disabled): 0
Dealer/Facility Code Match Handler Id (-1 = disabled): -1
Dealer Code (-1 = disabled): -1
Facility Code (-1 = disabled): -1

The default translation ID is the value the reader will enter in the log and is set by the user to any 16 digit or less value. The default handler indicates the reporting action(s) the system will take if an unknown tag is read. See "Read Handler" for a full list of options available.

The dealer and facility codes are programmed by Sirit during the manufacturing process and are unique to every Sirit customer. However, both Dealer code and the Dealer handler ID are not implemented in this version of the ID*entity* MaX reader.

5.8 Read Handler

The read event handler allows the user to specify a variety of actions to be taken when a tag is read and looked up in the translation table. In the event that an unknown tag is read, the default event handler specified in the Access Control menu will be used. With this menu, the user will be able to turn on a warning lamp, send a warning using the host serial port, raise the gate, or all of these.
To configure the handler, select the desired action from the pull down menu.

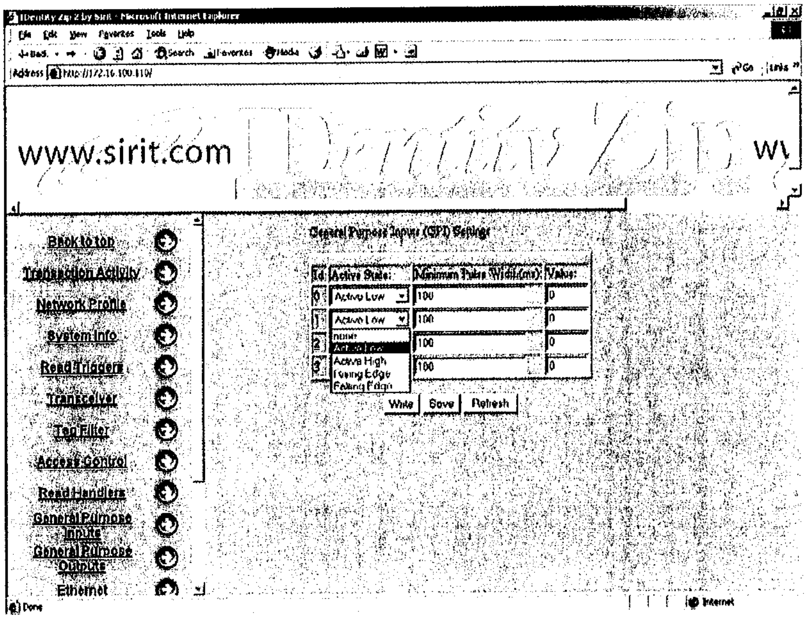


Read Handler Settings

Id	Output 0:	Output 1:	Output 2:	Output 3:	Output 4:	Output 5:	Output 6:	Output 7:	Description
0	gpo0	gpo1	gpo2	gpo3	wiegand	serial	ethernet	gate	Debug l

5.9 General Purpose Inputs (GPI)

Each of the four inputs can be set to an active high, active low, or edge triggered event.

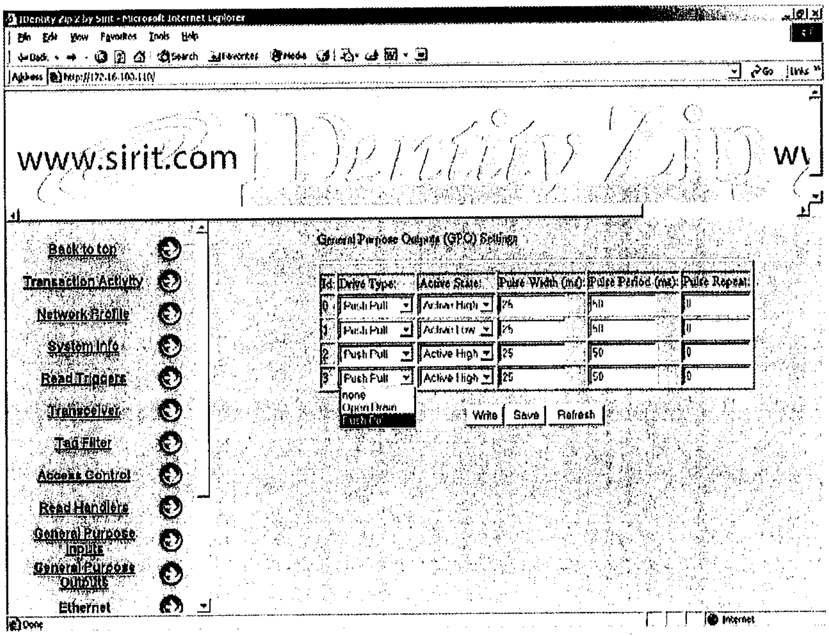


General Purpose Inputs (GPI) Settings

Id:	Active State:	Minimum Pulse Width(ms):	Value:
0	Active Low	100	0
1	Active Low	100	0
2		100	0
3		100	0

5.10 General Purpose Outputs (GPO)

The four general purpose outputs have several parameters that can be configured for a variety of applications.

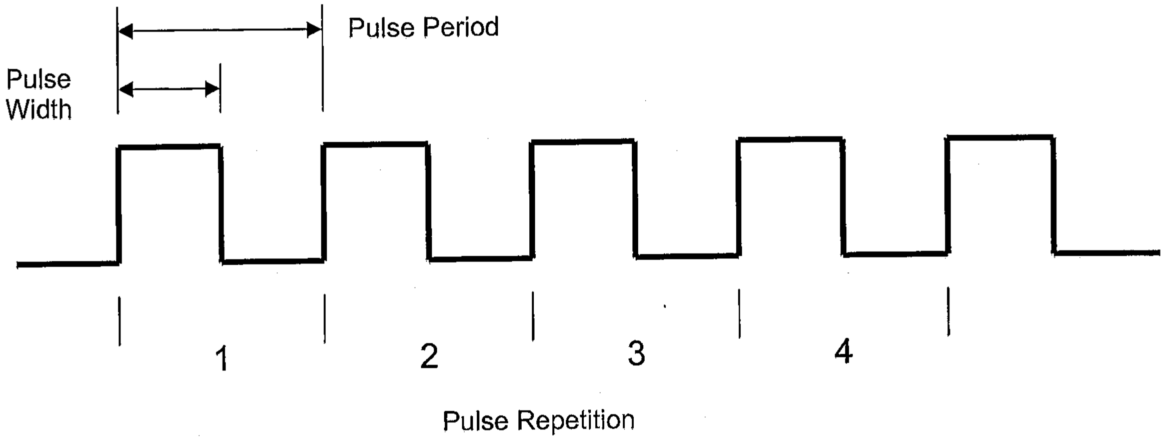


General Purpose Outputs (GPO) Settings

Id:	Drive Type:	Active State:	Pulse Width (ms):	Pulse Period (ms):	Pulse Repeat:
0	Push Pull	Active High	25	50	0
1	Push Pull	Active Low	25	50	0
2	Push Pull	Active High	25	50	0
3	Push Pull	Active High	25	50	0

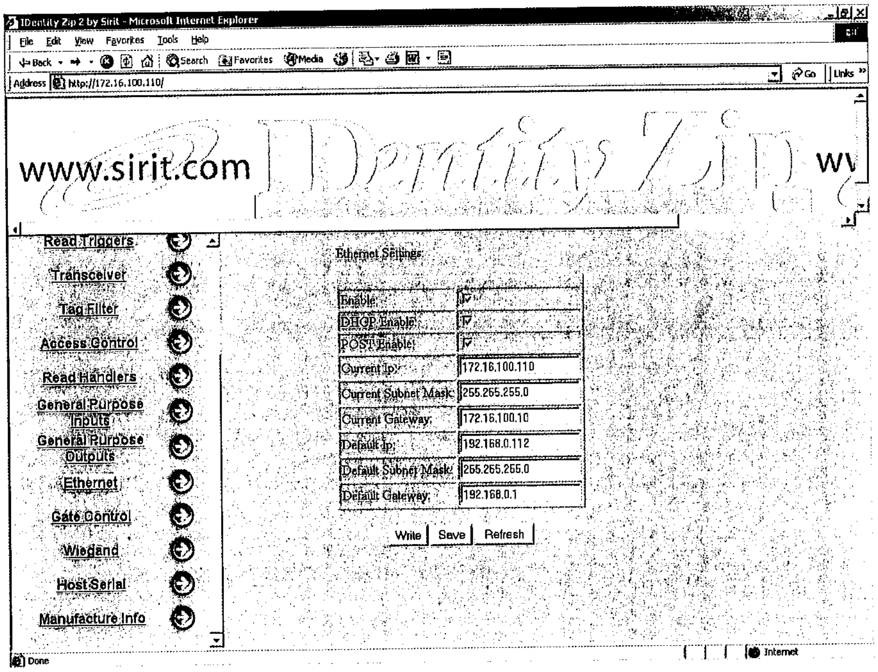
Active state can be set to active low or active high and the there are three options for output drive state. Open drain, Push pull, and none.

Each output waveform is specified with three parameters: Pulse width, pulse period and pulse repetition. The width is the time that the signal is active. Period is the time from rising edge to rising edge (Active high state) or falling edge to falling edge (Active low state). Pulse repeat is the number of times the active signal is repeated. The relationship between the three is shown below:



5.11 Ethernet

The Ethernet menu allows the user to enable various network functions and view and modify all ip information,



Ethernet Settings:

Enable:	☑
DHCP Enable:	☑
POST Enable:	☑
Current Ip:	172.16.100.110
Current Subnet Mask:	255.255.255.0
Current Gateway:	172.16.100.10
Default Ip:	192.168.0.112
Default Subnet Mask:	255.255.255.0
Default Gateway:	192.168.0.1

Write | Save | Refresh

The first enable is the overall control of the Ethernet port. Disabling this function turns off all TCP/IP communication to the reader. The DHCP enable activates the dynamic ip assignment to the reader. A DHCP server must be present to issue the ip address. The POST enable controls the XML transactions to and from the reader. By disabling this function, the user will not be able to read nor configure the ID*entity* MaX reader using the Ethernet port and the web based user interface. This is a security feature so that unauthorized personnel are not able to reconfigure the reader on an open network. While the POST enable is inactive, the admin will be able to configure the reader using the USB port and the PC based user interface.

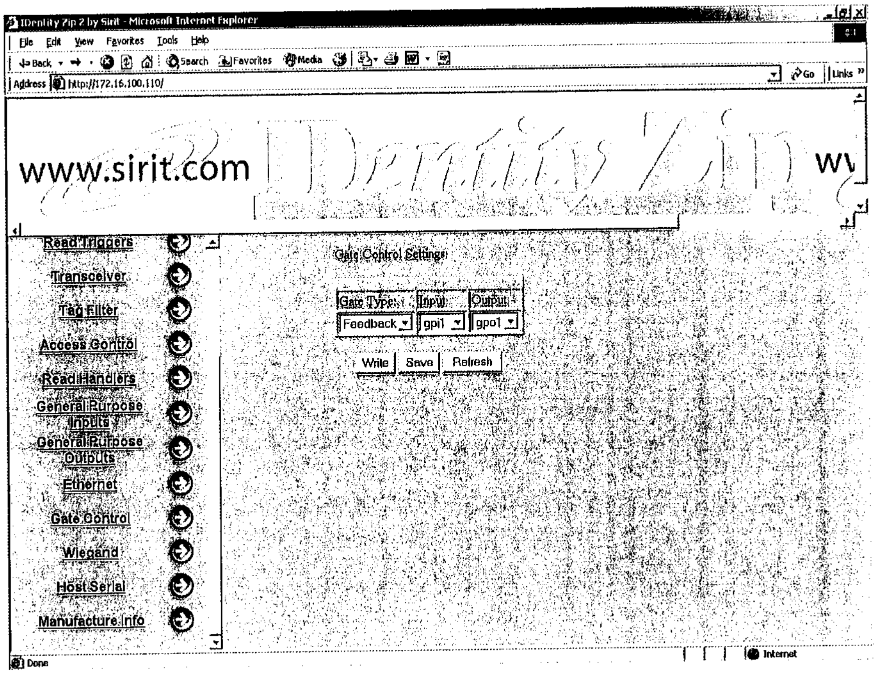
The remaining fields show the current and default network configuration. To change any of the parameters, simply enter in the desired values and click "write" or "save".

Note: Both current and default network parameters have been set to the following during this test phase

Ip: 172.18.27.231
Subnet mask: 255.255.255.0
Gateway: 172.18.27.254

5.12 Gate Control

The Gate Control menu is used to select the gate type (normal or feedback), the control input should synchronization be used, and the control output. All parameters are changed using the respective pull down menus.

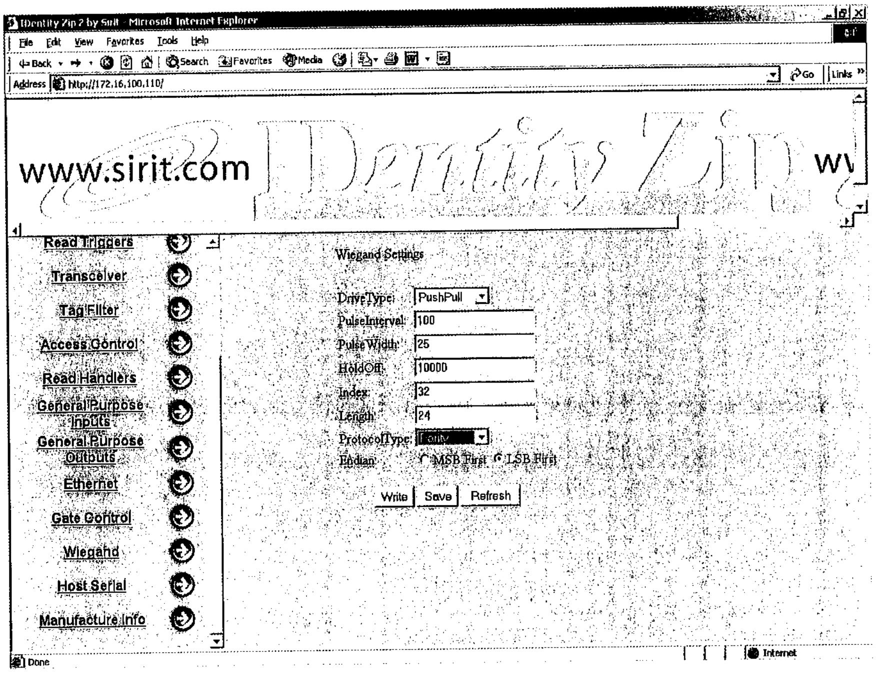


Gate Control Settings

Gate Type	Input	Output
Feedback	gpi1	gpo1

5.13 Wiegand Port Configuration

The Wiegand port is a simplex type communication compliant to the SIA 26-bit standard as well as extended formats.



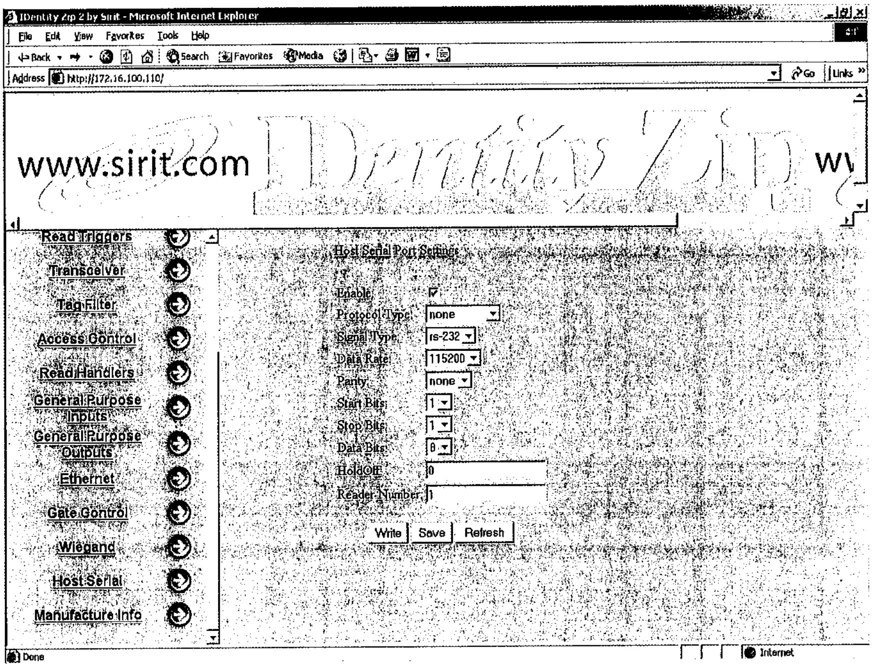
Wiegand Settings

DriveType: PushPull
PulseInterval: 100
PulseWidth: 25
HoldOff: 10000
Index: 32
Length: 24
ProtocolType: Parity
Endian: ○ MSB First ● LSB First

Transmission on the Wiegand interface can only occur when a corresponding Host Upload Event has been configured and satisfied. As before, the pulse width is the time that a data pulse is active for valid data and the pulse interval is the time between those pulses. Index is the offset (in bits) within the translated tag data where the serialization information will begin. Length specifies the length of the data packet, not including any protocol overhead. Both Parity and Checksum protocols are supported in addition to raw data format. Endian type denotes whether most significant (Big) or least significant (Little) will be transmitted first.

5.14 Host Serial

The serial port supports a variety of protocols and signaling.



Host Serial Port Settings

Enable ☑
Protocol Type: none
Signal Type: rs-232
Data Rate: 115200
Parity: none
Start Bits: 1
Stop Bits: 1
Data Bits: 8
HoldOff: 0
Reader Number: 1

To enable the host port, check the enable box. This serial port supports WPS and ACOM protocols, as well as RS-232, RS-422, and RS-485 signaling. To change any of the parameters, simply choose the value from the pull down menus. Hold off time refers to the time that is inserted between output data.

6.0 XML Command Set

XML Remote Procedure Calls (RPC) are used to remotely manipulate parameters on the IDentity MaX reader. These calls may be executed using the HTTP POST command issued to the reader device web-server. The XML command set is detailed in this section.

6.1 Get Parameter

When executed, this command will return the requested parameter value. One or more parameter names may be requested. The number of parameters is limited by overall document size limitation.

Command:

```
<methodName>GetParam</methodName>
<params>
        <param>
                <name><string> System.Time</string></name>
                <value><dateTime.iso8601>20050717T14:08:55</dateTime.iso8601></vlaue>
        </param>
</params>
```

Response:

```
<methodResponse>
        <params>
                <param>
                        <name><string>System.Time</string></name>
                        <value><>
                </param>
                <param>
                        <name><string>ReadTrigger.Enable</string></name>
                        <value>
                                <struct>
                                        <member>
                                                <name><string>Index</string></name>
                                                <value><i4>0</i4></value>
                                        </member>
                                </struct>
                        </value>
                </param>
        </params>
</methodResponse>
```

6.2 Set Parameter

When executed, this command will set the requested parameter.

Command:

```
<methodName>SetParam</methodName>
<params>
        <param>
                <name><string>System.Time</string></name>
                <value><dateTime.iso8601>20050717T14:08:55</dateTime.iso8601></vlaue>
        </param>
</params>
```

Response:

```
<methodResponse>
        <params>
                <param>
                        <name><string>System.Time</string></name>
                        <value><dateTime.iso8601>20050717T14:08:55</dateTime.iso8601></value>
```

```
            </param>
        </params>
</methodResponse>
```

6.3 Parameter Summary

Each of the following parameters may be configured using the Get and/or Set Parameter commands.

Field Name	Attrib.	Type	Description	Notes
Firmware.Name	R	string	Name of firmware. Ex) "Integrated Zip Reader"	
Firmware.MajorVer	R	i4	Major release version.	
Firmware.MinorVer	R	i4	Minor release version. i.e. updates	
Firmware.BuildVer	R	i4	Internal development version.	
System.Time	RW	Iso8601	Current reader date/time.	
ReadTrigger.Enable	RW	boolean	1 = Read trigger is enabled, 0 = Disabled	2,3
ReadTrigger.Input	RW	string	GPI# = A General Purpose Input is selected.	2
ReadTrigger.Output	RW	string	GPO# = The General Purpose Output to be used.; Blank = no discrete output.	2
ReadTrigger.Duration	RW	i4	Minimum amount of time the transceiver will attempt to interrogate the RF field.	2
ReadTrigger.Interval	RW	i4	Amount of time between internal timer input assertions.	
ReadTrigger.Retrigger	RW	boolean	1 = Read Duration restarts each time an active input is detected, 0 = Duration doesn't re-start.	2,3
Transceiver.TagType	RW	array[string]	"EPC Class 0", "EPC Class 1", "EPC Class 1 Gen 2", "Identity Flex", "Identity Zip", "Title 21" and/or "iso18000-6".	
ReadFilter.ValidationCount	RW	i4	Number of times a Tag ID must be received before being processed. 1 = First Id processed.	
ReadFilter.ValidationAttempts	RW	i4	Number of times a Tag ID verification may be attempted in attaining the required ValidationCount.	
ReadFilter.DuplicateFilterEnable	RW	boolean	1 = Duplicate Tag ID's are discarded, 0 = Duplicate ID's are not discarded.	3
ReadFilter.DuplicateTimeout	RW	i4	Minimum amount of time, in seconds, before allowing a duplicate Tag ID to be processed.	
ReadFilter.PassbackEnable	RW	boolean	1 = Tag ID timeout is re-started if a duplicate Tag ID is found within the original timeout period; 0 = Timeout is not changed/updated.	3
AccessControl.Enable	RW	boolean	1 = Tag ID is used to reference table entries to determine action.; 0 = All tag ID's are discarded.	3
AccessControl.TranslationDefault	RW	string	The translation ID to be used when a table lookup fails.	
AccessControl.HandlerIDDefault	RW	i4	The Handler ID to be used when a table lookup fails. A value of '-1' disables this feature.	
AccessControl.DealerHandlerID	RW	I4	Indicates the handler to be used when access is granted based upon Dealer Code. A value of '-1' disables this feature.	
AccessControl.DealerCode	RW	I4	If set, this code must match the tags dealer code value. A value of '-1' disables this feature.	
AccessControl.FacilityCode	RW	I4	If set, this code must match the tags facility code value. A Dealer Code must also be present for this value to be used. A value of '-1' disables this feature.	

ReadHandler.Description	RW	string	User-configured string describing the action. Ex) "Grant Access - Green Light" or "Deny Access - Red Light"	2
ReadHandler.Output	RW	array[string]	Array of up to 8 items referencing the desired output module; GPO#, WIEGAND, etc...	2
GateControl.GateType	RW	string	"Normal" or "Feedback"	
GateControl.Input	RW	string	GPI# = The General Purpose Input to monitor for gate feedback.	
GateControl.Output	RW	string	GPO# = The General Purpose Output to signal for gate control.	
HostSerial.Enable	RW	boolean	1 = Serial module is enabled, 0 = Serial module is disabled.	3
HostSerial.ProtocolType	RW	string	WPS, ASCOM	
HostSerial.SignalType	RW	string	RS-232, RS-422, RS-485	
HostSerial.DataRate	RW	i4	Numeric data rate. Ex) 115200, 57600	
HostSerial.Parity	RW	string	Parity bit type preceeding each byte transmitted/received; Ex) None, Odd, Even	
HostSerial.StartBits	RW	i4	Number of data '1' start bits to preceed each data byte transmitted.	
HostSerial.StopBits	RW	i4	Number of data '0' stop bits to follow each data byte transmitted.	
HostSerial.DataBits	RW	i4	Number of data bits per byte transmitted.	
HostSerial.HoldOff	RW	i4	Amount of time between any two packets, in milli-seconds.	
HostSerial.ReaderNum	RW	i4	Reader number used for calculating lane number in some host protocols. Valid values are 1-64.	
Wiegand.DriveType	RW	string	"none", "OpenDrain" or "PushPull"	
Wiegand.PulseInterval	RW	i4	Amount of time, in milliseconds, between output pulses.	
Wiegand.PulseWidth	RW	i4	Amount of time, in milliseconds, per pulse.	
Wiegand.HoldOff	RW	i4	Amount of time between any two packets, in milli-seconds.	
Wiegand.Index	RW	i4	Indicates the number of least significant translated data bits to disregard.	
Wiegand.Length	RW	i4	Indicates the number of data bits to transmit regardless of index.	
Wiegand.ProtocolType	RW	i4	"none", "parity" or "checksum"	
Wiegand.Endian	RW	boolean	1 = LSB first, 0 = MSB first.	3
GPI.ActiveState	RW	string	"ActiveLow", "ActiveHigh", "RisingEdge", "FallingEdge"	2
GPI.PulseWidth	RW	i4	Input de-glitch timeout in milliseconds. Maximum value of 1 second.	
GPI.Value	R	boolean	1 = Input high, 0 = Input Low	
GPO.DriveType	RW	string	"none", "OpenDrain" or "PushPull"	2
GPO.ActiveState	RW	boolean	1 = Active High, 0 = Active Low	2,3
GPO.PulseWidth	RW	i4	Active state pulse width, in milliseconds.	2
GPO.PulsePeriod	RW	i4	Minimum inactive state period between pulses, in milliseconds.	2
GPO.PulseRepeat	RW	i4	Number of pulse iterations per outputs.	2
Ethernet.Enable	RW	boolean	1 = Ethernet I/O is enabled, 1 = Ethernet is disabled	3
Ethernet.DefaultIP	RW	string	Default IP address in octet format. Ex)192.168.0.4	
Ethernet.DefaultMask	RW	string	Default subnet mask in octet format. Ex)255.255.255.0	
Ethernet.DefaultGateway	RW	string	Default subnet mask in octet format. Ex)192.168.0.1	
Ethernet.IP	RW	string	Default IP address in octet format. Ex)192.168.0.4	5
Ethernet.Mask	RW	string	Default subnet mask in octet format. Ex)255.255.255.0	5
Ethernet.Gateway	RW	string	Default subnet mask in octet format. Ex)192.168.0.1	5
Ethernet.DHCPEnable	RW	boolean	1 = DHCP is enabled, 0 = DHCP is disabled.	3
Ethernet.POSTEnable	RW	boolean	1 = HTTP POST command is enabled, 0 = POST is disabled.	3

Translation.IndexStored	R	boolean	True = Index backup is up-to-date, False = Index has not been stored.	3
Esn.Nomenclature	RW	string	Manufacturing Nomenclature. ex)"IntegratedZip"	4
Esn.BoardVer	RW	String	Manufacturing Board Version. ex)"2.2.3"	4
Esn.Serial	RW	String	Manufacturing Serial Number. ex)"0123456789"	4
Esn.TestDate	RW	String	Manufacturing Test Date. Format: MM/DD/YYYY	4
Esn.ManufactureCode	RW	String	Manufacturing Code. Ex)"0123"	4
Esn.MacAddress	RW	String	Manufacturing assigned unique Ethernet HW address. Ex)"001122334455"	4

Note	Description
1	Field instance must be specified with Tag ID (string type) as an index.
2	Field instance must be specified with a numeric index value appended to the field name. Ex) "GPI.ActiveState0". The default index assumed is zero if none is specified. Ex) "GPI.ActiveState"
3	For 'boolean' fields, '0' = False and '1' = True
4	Fields are read-write to RAM address space. Storage of values to non-volatile memory require usage of the 'StoreEsn' command.
5	Run-time parameter only, variable is not stored in ROM at any point

6.4 Save Configuration

This method will commit configuration parameters to non-volatile memory.

Command:

```
<methodName>SaveConfigurationParams</methodName>
```

Response:

```
<methodResponse>
        <params>
                <param>
                        <name><string>StatusCode</string></name>
                        <value><i4>1</i4></value>
                </param>
        </param>
</methodResponse>
```

6.5 Load Configuration

This method will restore configuration parameters from non-volatile memory.

Command:

```
<methodName><string>LoadConfigurationParams</string></methodName>
```

Response:

```
<methodResponse>
        <params>
                <param>
                        <name><string>StatusCode</string></name>
                        <value><i4>1</i4></value>
                </param>
        </param>
</methodResponse>
```

6.6 Read Translation Table Entries

This method will read one or more records from the translation table. The specified Tag ID parameter will indicate the first record to be read. The Count parameter indicates the requested number of Tag ID records to read in ascending order. The actual number of records returned is limited by the number of ascending entries available and maximum file length. If the specified Tag ID is "00", the lowest numeric value ID will be the first record returned. If the specified Tag ID cannot be found in the translation table, the next available Tag ID will be returned first based on ascending numeric Tag ID value. A method response without any 'struct' entries indicates no records were available to be read.

Each entry within the translation table is unique based upon Tag ID. Therefore, to sequentially read a table that exceeds the maximum individual file length, this command should be executed multiple times with the specified Tag ID equal to the last ID read plus 1.

Command:

```
<methodName>ReadTranslationRecords</methodName>
<params>
        <param>
                <name><string>TagID</string></name>
                <value><string>1</string></value>
        </param>
        <param>
                <name><string>Count</string></name>
                <value><i4>1<i4></value>
        </param>
</params>
```

Response:

```
<methodResponse>
        <structs>
                <struct>
                        <member>
                                <name><string>TagID</string></name>
                                <value><string>01234567890ABCDEF</string></value>
                        </member>
                        <member>
                                <name><string>TranslationID</string></name>
                                <value><string>0011002200330044</string></value>
                        </member>
                        <member>
                                <name><string>HandlerID</string></name>
                                <value><i4>2</i4></value>
                        </member>
                </struct>
                .............
                <struct>
                        <member>
                                <name><string>TagID</string></name>
                                <value><string>1100220033004400</string></value>
                        </member>
                        <member>
                                <name><string>TranslationID</string></name>
                                <value><string>0011002200330044</string></value>
                        </member>
                        <member>
                                <name><string>HandlerID</string></name>
                                <value><i4>2</i4></value>
                        </member>
                </struct>
        </structs>
</methodResponse>
```

6.7 Write Translation Table Entries

This method will write translation table entries to the internal reader translation table. For optimum indexing speed, entries should be pre-sorted in ascending order according to Tag ID. If a large number of existing Tag IDs are to be uploaded, the existing table should be deleted and new sorted table uploaded for optimal upload performance. Once translation tables and/or entries are written, the Store Index command should be executed to store the indexing updates and facilitate quicker system reboot time.

Command:

```
<methodName>WriteTranslationRecords</methodName>
        <structs>
                <struct>
                        <member>
                                <name><string>TagID</string></name>
                                <value><string>01234567890ABCDEF</string></value>
                        </member>
                        <member>
                                <name><string>TranslationID</string></name>
                                <value><string>0011002200330044</string></value>
                        </member>
                        <member>
                                <name><string>HandlerID</string></name>
                                <value><i4>2</i4></value>
                        </member>
                </struct>
                .............
                <struct>
                        <member>
                                <name><string>TagID</string></name>
                <value><string>1100220033004400</string></value>
            </member>
            <member>
                <name><string>TranslationID</string></name>
                <value><string>0011002200330044</string></value>
            </member>
            <member>
                <name><string>HandlerID</string></name>
                <value><i4>2</i4></value>
            </member>
        </struct>
    </structs>
</methodName >
```

Response:

```
<methodResponse>
    <params>
        <param>
            <name><string>Count</string></name>
            <value><i4>0</i4></value>
        </param>
    </params>
</methodResponse>
```

6.8 Delete Translation Table Entries

This method will delete one or more translation table entries starting with the specified Tag ID and continuing in ascending Tag ID order until the Count value is reached or the translation table is empty. The number of entries deleted is returned.

If the ExactMatch parameter is set to '1', the specified TagId must exist within the Translation Table. Otherwise, TagId's greater than the specified Id will be deleted. To delete all records within the Translation Table regardless of Id, specify a TagId of "", Count of of '-1' and ExactMatch of '0'.

Command:

```
<methodName>DeleteTranslationRecords</methodName>
<params>
    <param>
        <name><string>TagID</string></name>
        <value><string>1</string></value>
    </param>
    <param>
        <name><string>Count</string></name>
        <value><i4>1<i4></value>
    </param>
    <param>
        <name><string>ExactMatch</string></name>
        <value><boolean>1<boolean></value>
    </param>
</params>
```

Response:

```
<methodResponse>
        <params>
                <param>
                        <name><string>Count</string></name>
                        <value><i4>1<i4></value>
                </param>
        </params>
</methodResponse>
```

6.9 Store Translation Table Index

This method will commit the RAM based translation table indexes to FLASH memory. If an up-to-date index table has already been transferred to FLASH memory, it will not be re-written.

Command:

```
<methodName>StoreTranslationIndex</methodName>
```

Response:

```
<methodResponse>
        <params>
                <param>
                        <name><string>StatusCode</string></name>
                        <value><i4>0</i4></valuelean>
                </param>
        </params>
</methodResponse>
```

6.10 Read Transaction Log

This method will read one or more records from the transaction log. The first execution of this command should specify the 'ResetStart' as '1' (True) to mark the extraction start point. Subsequent execution(s) of this command will sequentially return older remaining entries provided the 'ResetStart' parameter is specified as '0'. This results in the most recent transaction log entries being extracted first.

Command:

```
<methodName>ReadTransactionRecords</methodName>
<params>
        <param>
                <name><string>ResetStart</string></name>
        <value><boolean>1</boolean></value>
    </param>
    <param>
        <name><string>MaxCount</string></name>
        <value><i4>2</i4></value>
    </param>
</params>
```

Response:

```
<methodResponse>
    <structs>
        <struct>
            <member>
                <name><string>TagID</string></name>
                <value><string>0011002200330044</string></value>
            </member>
            <member>
                <name><string>Time</string></name>
                <value><dateTime.iso8601>20050717T14:08:55</dateTime.iso8601>
                </value>
            </member>
            <member>
                <name><string>HandlerRec</string></name>
                <value><i4>2</i4></value>
            </member>
        </struct>
        .............
        <struct>
            <member>
                <name><string>TagID</string></name>
                <value><string>0011002200330044</string></value>
            </member>
            <member>
                <name><string>Time</string></name>
                <value><dateTime.iso8601>20050717T14:08:56</dateTime.iso8601>
                </value>
            </member>
            <member>
                <name><string>HandlerRec</string></name>
                <value><i4>2</i4></value>
            </member>
        </struct>
    </structs>
</methodResponse>
```

6.11 Upload Firmware

This method is used to buffer firmware on the device for upgrade. Multiple methods may be executed to load an entire firmware image. A status code response is generated indicating whether or not parameters are formatted correctly and expected values were found.

Command:

```
<methodName>UploadFirmware</methodName>
<params>
    <param>
        <name>Format</name>
        <value>extendedIntelHex</value>
        </param>
        <param>
                <name>RecordIndex</name>
                <value><i4>0</i4></value>
        </param>
        <param>
                <name>TotalRecords</name>
                <value><i4>3</i4></value>
        </param>
        <param>
                <name>TargetCode</name>
                <value><i4>1234</i4></value>
        </param>
        <param>
                <name>DataRecords</name>
                <value><array><data>
                        <value>:020000040002F8</value>
                        <value>:1000000000000FE11F00C0E3120080E300F021E1D7</value>
                        ...
                        <value>:00000001FF</value>
                </data></array></value>
        </param>
</params>
```

Response:

```
<methodResponse>
        <params>
                <param>
                        <name><string>StatusCode</string></name>
                        <value><i4>0</i4></value>
                </param>
        </param>
</methodResponse>
```

6.12 Execute Firmware

This method initiates internal transfer/upgrade of firmware from after buffers have been successfully loaded using the Upload Firmware command. Following execution of this command, two seconds will be given before the device goes off-line & begins the firmware transfer.

Command:

```
<methodName><string>ExecuteFirmwareLoad</string></methodName>
```

Response:

```
<methodResponse>
        <params>
                <param>
                        <name><string>StatusCode</string></name>
                        <value><i4>0</i4></value>
                </param>
        </param>
</methodResponse>
```

6.14 Example XML-RPC transaction

```
POST /RPC2 HTTP/1.0
User-Agent: ZipReader
Host: ZipReader.sirit.com
Content-Type: text/xml
Content-length: 181

<?xml version="1.0"?>
<methodCall>
        <methodName>GetParam</methodName>
        <params>
                <param>
                        <name><string>Firmware.Name</string></name>
                </param>
                <param>
                        <name><string>Firmware.MajorVer</string></name>
                </param>
                <param>
                        <name><string>Firmware.MinorVer</string></name>
                </param>
        </params>
</methodCall>
```

An example XML-RPC response is as follows:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 158
Content-Type: text/xml
Date: Fri, 17 Jul 2005 19:55:08 GMT
Server: ZipReader

<?xml version="1.0"?>
<methodResponse>
        <params>
                <param>
                        <name><string>Firmware.Name</string></name>
                        <value><i4>1</i4></value>
                </param>
                <param>
                        <name><string> Firmware.MajorVer</string></name>
                        <value><i4>1</i4></value>
                </param>
                <param>
                        <name><string>Firmware.MinorVer</string></name>
                        <value><i4>2</i4></value>
                </param>
        </params>
</methodResponse>
```

Appendix A

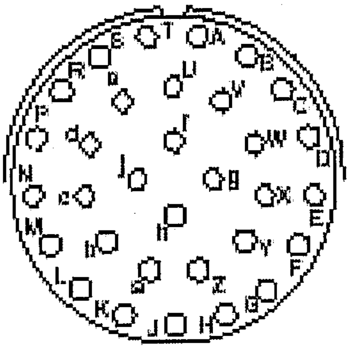
Default Values for Configuration Parameters

Parameter	Default Value	Range
Date and Time	US Central (GMT-6)	
GPI		
Active state	Low	
Pulse width	100 ms	
GPO		
Active state	Low	
Drive type	Push pull	
Pulse width	25 ms	1 – 999 ms
Pulse period	50 ms	1 – 999 ms
Pulse repeat	0	
Serial Port		
Port enable	Enabled	
Protocol	None	
Signal type	RS-232	
Baud rate	115200	9600 - 115200
Parity	None	
Start bit	1	
Stop bit	1	1 - 2
Data bits	8	5 - 8
Hold off	20 ms	0 – 9999 ms
Wiegand Port		
Hold off	10 ms	0.1 - 1000 ms
Drive type	Push pull	
Pulse width	25 us	20 – 100 us
Pulse Interval	100 us	100 – 20000 us
Index	24	
Length	32	24 - 128
Protocol	Parity	
Endian type	Little	
Access Control		

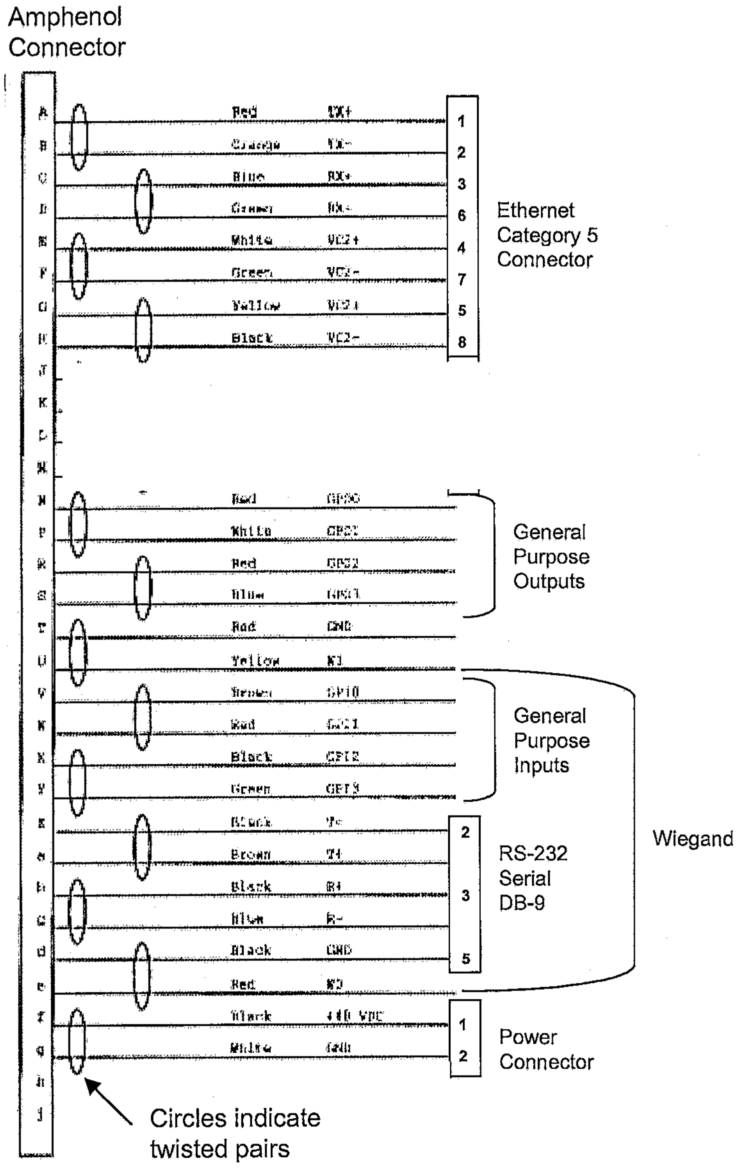
Control enable	Disabled	
Default Translation ID	FFFFFFFFFFFFFFFF	
Handler ID	0	0 - 8
Dealer handler ID	0	0 - 8
Dealer code	Not used	
Facility code	Not used	
Gate Control		
Gate type	Normal	
Input	None	GPI0 – GPI3
Output	None	GPO0 – GPO3
Trigger Settings		
Trigger enable	Enabled	
Trigger input	T0	GPI0 – GPI3
Trigger output	None	GPO0 – GPO3
Trigger duration	100 ms	
Retrigger	Enabled	
Filter		
Duplicate Filter Enable	Enabled	
Filter timeout	2 sec	1 – 255 sec
Validation count	1	1 - 255
Validation attempts	5	1 - 255
Passback	Disabled	
Ethernet		
Enabled	Enabled	
DHCP Enabled	Enabled	
POST Enabled	Enabled	
Current ip	172.18.27.231	
Currnet subnet mask	255.255.255.0	
Current gateway	172.18.27.254	
Default ip	172.18.27.231	
Default subnet mask	255.255.255.0	
Default gateway	172.18.27.254	

Appendix B

Interface Cable Pinout



Amphenol connector pinout
(rear face of socket insert shown)



Interface cable and connector pin assignments