AI1301 Read-Only Reader Card
User Guide

TransCore
8600 Jefferson Street NE
Albuquerque, New Mexico 87113

February 2012

P/N 411062-008

For further information, contact:

TransCore
3410 Midcourt Road, Suite 102
Carrollton, Texas 75006 USA

Phone: (214) 461-4031
Fax: (214) 461-6478

Technical Support
Web: transcore.com/rfidsupport
Phone: (505) 856-8007

For comments or questions about this document, e-mail tech.pubs@transcore.com.

WARNING TO USERS IN THE UNITED STATES

FCC RADIO FREQUENCY INTERFERENCE STATEMENT
47 CFR §15.105(a)

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency (RF) energy and may cause harmful interference to radio communications if not installed and used in accordance with the instruction manual. Operating this equipment in a residential area is likely to cause harmful interference, in which case, depending on the regulations in effect, the users may be required to correct the interference at their own expense.

NO UNAUTHORIZED MODIFICATIONS
47 CFR §15.21

CAUTION: This equipment may not be modified, altered, or changed in any way without permission from TransCore, LP. Unauthorized modification may void the equipment authorization from the FCC and will void the TransCore warranty.

USE OF SHIELDED CABLES IS REQUIRED
47 CFR §15.27(a)

NOTE: Shielded cables must be used with this equipment to comply with FCC regulations.

TransCore, LP
USA

Contents

Contents

1 Introduction

Purpose of Guide 1-3

Overview 1-3
General 1-3
System Configuration 1-4
System Components 1-6
AI1301 Reader Card 1-7
Interface Boards 1-9
RF Module 1-11
Antenna 1-12
Tag 1-13

2 Reader Card Installation Procedures

Selecting the Base Address 2-3

Setting the I/O Expansion Jumper 2-4

Selecting an Interrupt 2-4

Installing a Reader Card 2-5

3 Reader Card Operation

CPM Interface 3-3

Tag Communication Protocol 3-3

Command Syntax 3-3

Response Syntax 3-4

Tag Read Transactions 3-4

I/O Port Usage 3-6

Registers 3-6
Command Registers 3-7
Status Registers 3-8

Status Register Bit Definitions *3-9*

A Reader Card Specifications

B J1 Edge Connector Pin-Out to STD Bus

C Sample C Code

List of Figures

Figure 1-1 Sample read-only configuration using AR2200 RF module 1-4
Figure 1-2 Sample read-only configuration using AR2602 or AR2603 RF module 1-5
Figure 1-3 AI1301 read-only system components 1-6
Figure 1-4 AI1301 reader card 1-7
Figure 1-5 Interface board connectors 1-8
Figure 1-6 RF interface board 1-10
Figure 1-7 I/O interface board 1-10
Figure 1-8 RF interface wiring diagram for AR2200 RF module and AI1301 reader card 1-12
Figure 2-1 S1 dip switch 2-3
Figure 2-2 I/O expansion jumper 2-4
Figure 2-3 S2 dip switch 2-5

List of Tables

Table 1-1 LED Indicators 1-9

1

Introduction

Chapter 1
Introduction

Purpose of Guide

This guide contains information for using the TransCore model AI1301 read-only reader card in the TransCore International Standards Organization (ISO) format read-only radio frequency (RF) identification system.

The AI1301 User Guide covers the basic capabilities, operation, installation and use of the model AI1301 reader card in a read-only RF identification system. STD bus pin definitions are included as Appendix A.

In this guide, except where needed for clarity, ISO format is presumed, the system name and tag types referred to are read-only, all tags are presumed to be TransCore models, the word "system" implies "read-only system," and the model AI1301 read-only reader card is referred to as the reader card.

Overview

This chapter provides an overview of the AI1301 reader card and the system and technology it supports. The AI1301 reader card is used in the TransCore ISO-format RF identification system. The ISO read-only RF identification system automatically reads the IDs of read-only tags.

The reader portion of this system is modularized using individually addressable reader cards that communicate with a host computer and serve local equipment and devices.

General

The AI1301 provides the logic necessary to decode tag data and communicate with interface devices and users' application software. It acts as a "switchboard" for communicating between the system and tags, and is used to convey, process, direct and control the signals that flow through, report from, monitor or regulate other system equipment.

The AI1301 reader card is a rack-insertable, bus-operated, modular component of the system. The AI1301 reader card operates from an STD bus, offering the functionality of the AI1200 reader on a convenient STD bus-compatible card.

The reader card is mainly controlled by user-generated software. It has no embedded micro controller and is operated by a user-supplied central processing module (CPM) over a system bus that can hold other reader cards and optional equipment. The CPM controls and communicates with one or more reader cards. Any bus having one or

more reader cards must also have at least one CPM installed.

System Configuration

The read-only system may use multiple reader cards. Each reader card can be configured to a unique address and connected to an STD bus served by a CPM running user-generated software. An RF module in the system is "paired" with a reader card that controls it and any associated external actuators and sensors.

The read-only system is compatible with three RF modules: AR2200, AR2602, and AR2603.

In systems configured with AR2200 RF modules, a single RF module can control two antenna channels (Figure 1-1).

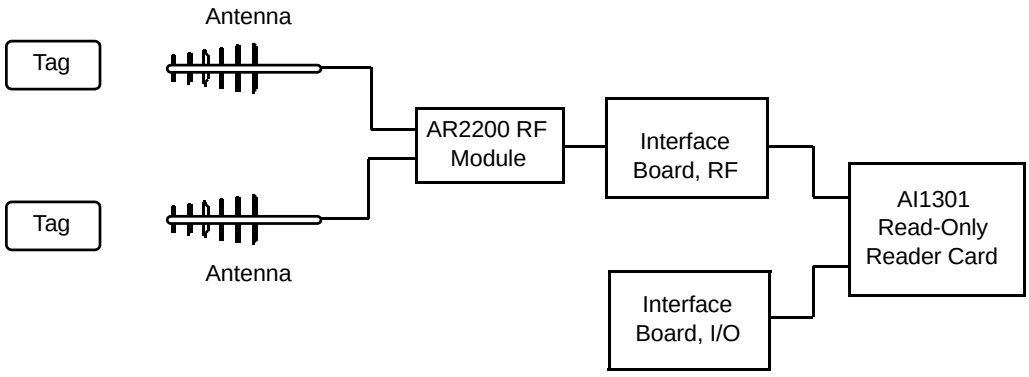

Figure 1-1 Sample read-only configuration using AR2200 RF module

In systems configured with AR2602 or AR2603 RF modules, two RF modules, each controlling one antenna, may be connected to the RF interface board (Figure 1-2).

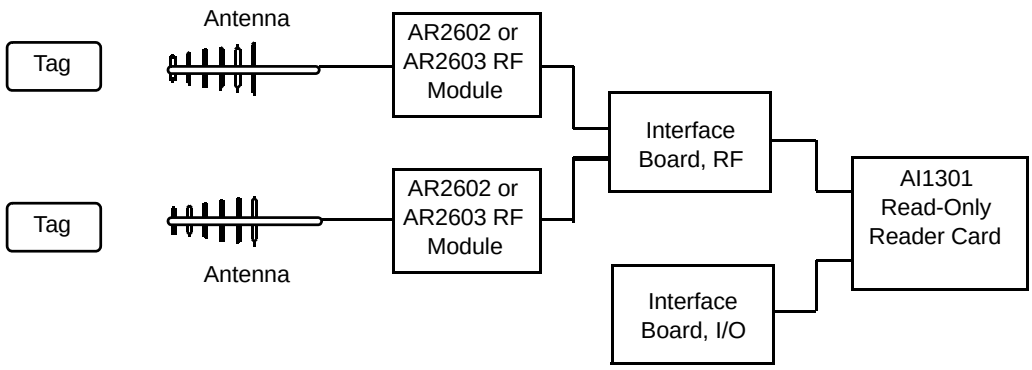

Figure 1-2 Sample read-only configuration using AR2602 or AR2603 RF module

System Components

Figure 1-3 illustrates the seven fundamental components of the AI1301 read-only system.

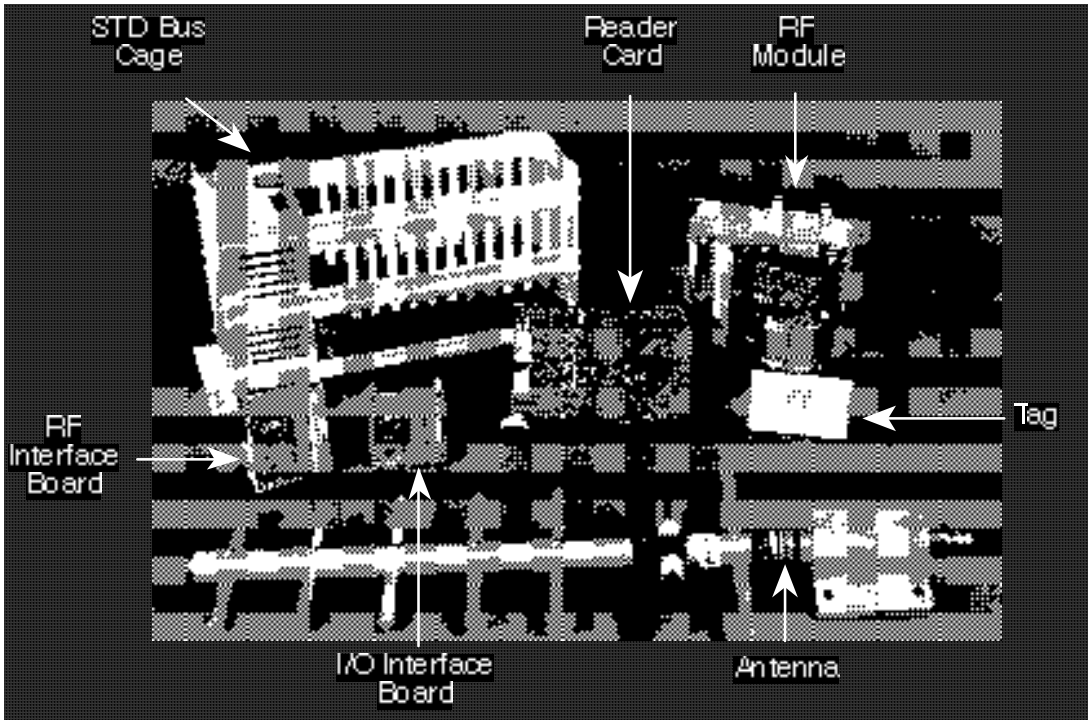

Figure 1-3 AI1301 read-only system components

STD Bus

The AI1301 (Figure 1-4) is designed to plug directly into the STD bus. The number of cards is limited by the physical size of the STD bus; the maximum is approximately 20 slots. Other devices on the bus communicate directly with the AI1301 using simple I/O addressing techniques. Multiple reader cards may be installed on the same bus by selecting different board addresses for each card.

AI1301 Reader Card

The reader card (Figure 1-4) operates from +5 VDC provided from the system bus through the card edge connector.

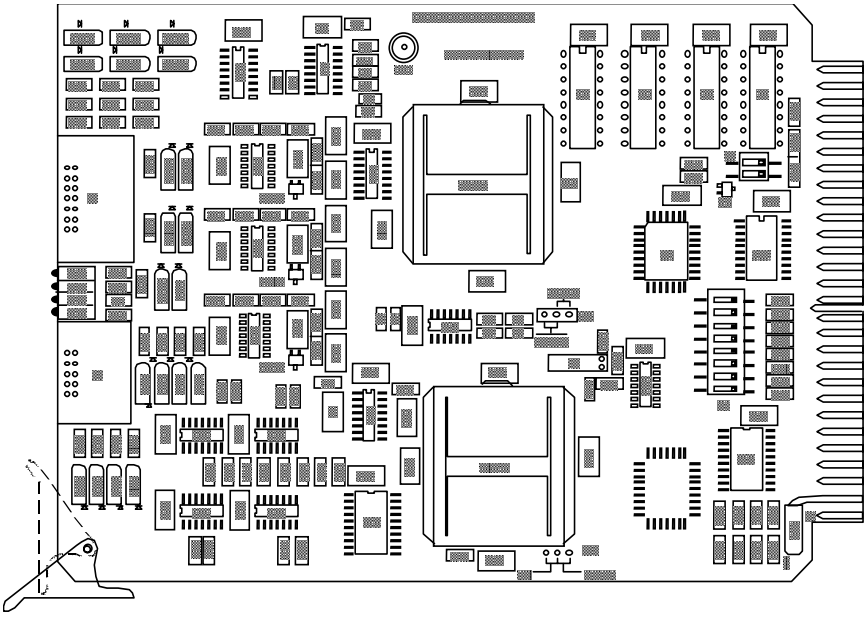

Figure 1-4 AI1301 reader card

Interface Board Connectors

Two connectors are provided for connection to interface boards which enable easy termination of field wiring (Figure 1-5). Each connector serves a single interface board.

Reader card connector J3 serves the RF interface board that operates the RF module(s). A 14-wire ribbon cable links the reader card's RF unit interface connector (J3) with an RF interface board.

Reader card connector J2 serves the I/O interface board that connects to the associated alarms, status lines, actuators, annunciators and other peripherals. A 10-wire ribbon cable links the reader card's digital I/O interface connector (J2) with an I/O interface board.

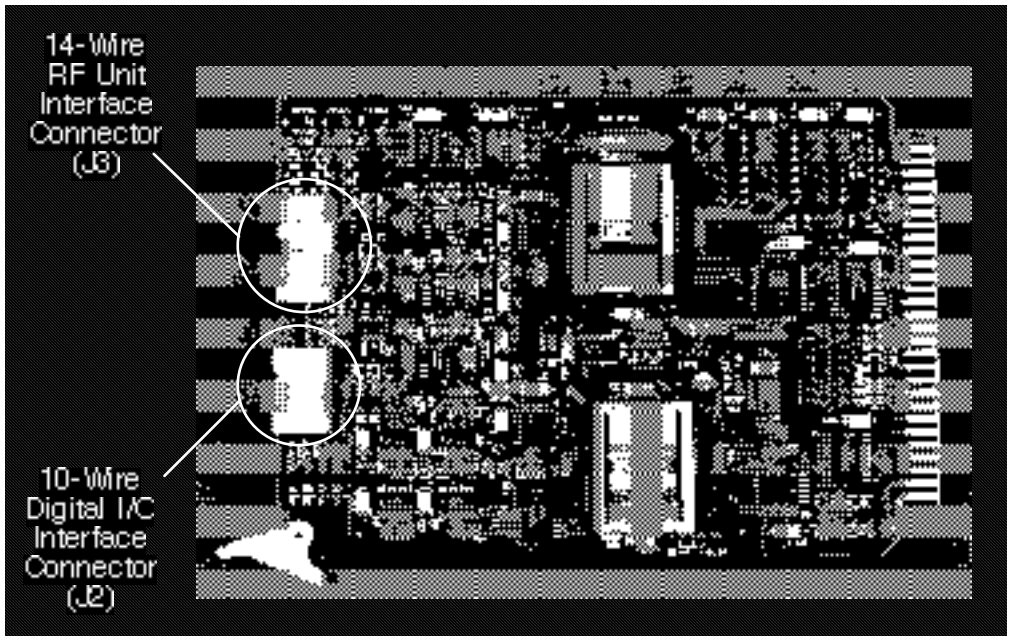

Figure 1-5 Interface board connectors

There are eight TTL I/O lines on the reader card which can be individually configured as input or output. These command- and status register-accessible TTL lines can be used for various ancillary functions such as loop detection, gate control and check tag control.

LED Indicators

Four LED indicators are mounted on the edge of the board to indicate the status of the operations being performed by the reader card. Each LED illuminates for one of the conditions shown in Table 1-1 below:

Table 1-1 LED Indicators

LED	Color	Condition
DS1	Red	Board Select
DS2	Red	Lock
DS3	Red	RFO Enabled
DS4	Red	RF1 Enabled

Interface Boards

Interface boards connect a reader card with external equipment. The AI1301 reader card system uses two types of interface boards: an I/O interface board and an RF interface board.

Each reader card can support one I/O interface board and one RF interface board. On both types of board, an IDC header connector serves the reader card and a terminal block serves the associated equipment. The connectors that mate with the terminal block use screw-type wire stations and are connectorized for board replacement without disturbing field wiring.

RF Interface Board

The RF interface board (Figure 1-6) connects the reader card to the RF module(s). A 14-wire ribbon cable links the reader card's RF unit interface connector (J3) with the RF interface board.

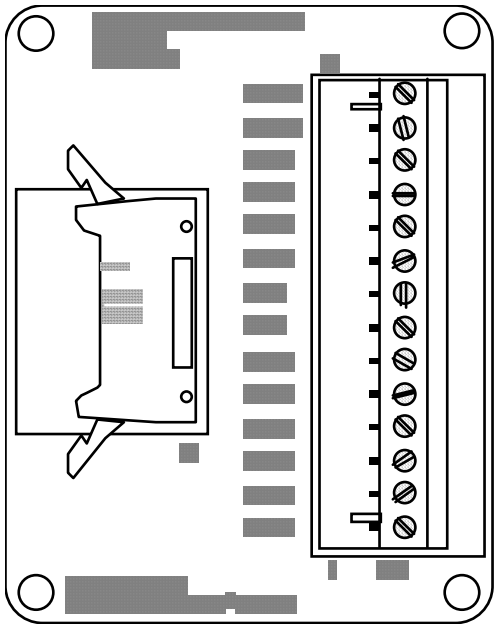

Figure 1-6 RF interface board

I/O Interface Board

The I/O interface board (Figure 1-7) connects the reader card to equipment such as alarms, status lines, actuators, annunciators and other peripherals. It may also control system check tags. A 10-wire ribbon cable links the reader card's digital I/O interface connector (J2) with the I/O interface board.

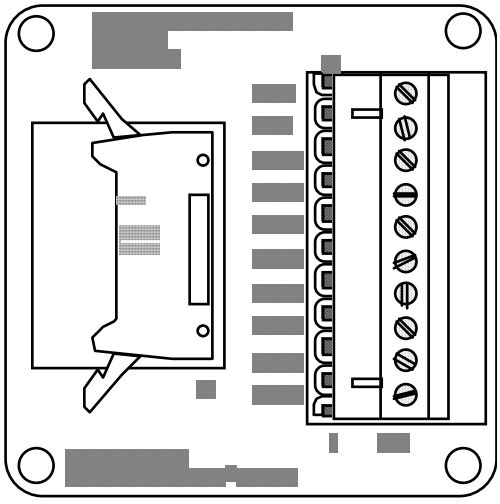

Figure 1-7 I/O interface board

RF Module

The AI1301 read-only system is designed to operate most effectively with models AR2200, AR2602 and AR2603 RF modules. The AR2200 RF module (dual-output) and the AR2602 and AR2603 RF modules (single-output) are radio transmitter/receivers which, upon command from the reader card, generate an RF signal to one or two antennas for broadcast to tags. The RF module receives and demodulates the modulated backscatter RF signal returned from a tag through the antenna. The RF module then preamplifies and conditions the demodulated signal before sending it to the reader card.

Each RF module generates a continuous-wave (CW) unmodulated RF signal. During read transactions, the RF module transmits an RF signal which is reflected by the tag. The RF module retrieves this reflected signal and sends it to the reader card for processing.

The RF module is software-controlled by the host computer. It can either operate continuously, or be turned on in response to a sensor input signal indicating the detected presence of an object. While only one AR2200 RF module can be connected to a reader card, it can control up to two antennas. Two AR2602 or AR2603 RF modules, each controlling a single antenna, may be connected to a reader card. (See Figure 1-1 and Figure 1-2.)

DC power to the RF module is externally supplied. For optimum performance, use only an TransCore-approved power supply.

RF Module Wiring Diagram

Figure 1-8 illustrates the wiring diagram for the AR2200 RF module and the AI1301 reader card.

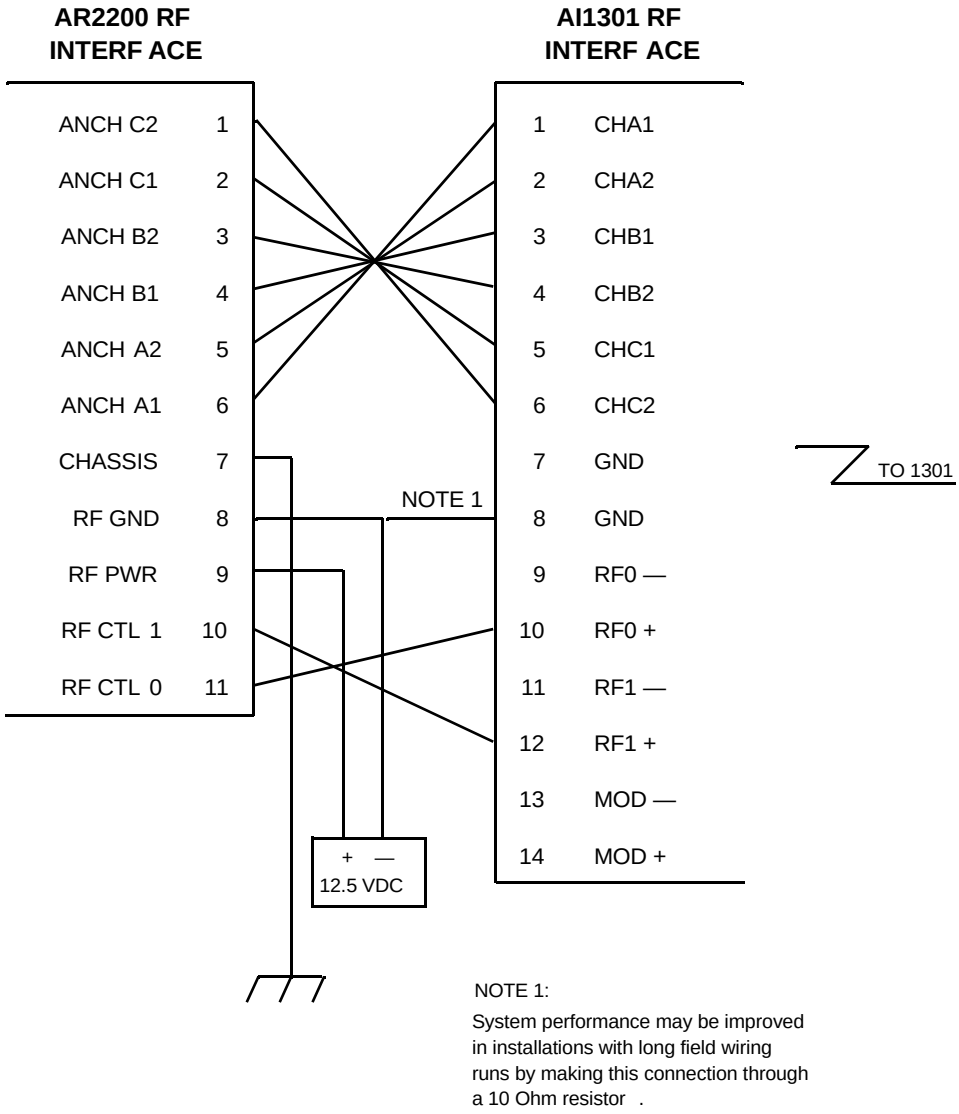

Figure 1-8 RF interface wiring diagram for AR2200 RF module and AI1301 reader card

Antenna

Each active antenna broadcasts the signal generated by its associated RF module and receives the reflections returned to that antenna from tags within range. The system response needed at each antenna site determines the antenna type, orientation and mounting configuration.

Tag

All TransCore tags are passive transponders that reflect or backscatter a part of the incident RF energy they receive from a system antenna. The signal backscattered by a tag is modulated in a code that identifies the tag to the interrogating system.

Compatibility

All TransCore ISO read-only and read/write systems can read TransCore ISO read-only and read/write tags. These tags include all TransCore toll and rail tags sold for domestic (U.S.) use.

ID Programming

Read-only tag IDs must be programmed by direct contact with the tag circuit. They cannot be programmed on equipment compatible with read/write systems.

ID and Memory Capacities

Read-only tags have a single, 128-bit frame that holds the tag ID. The tag ID is permanent unless manually reprogrammed.

Operation of Read-only Tags

When interrogated, read-only tags report only the ID numbers permanently programmed into the tag.

2

Reader Card Installation Procedures

Chapter 2

Reader Card Installation Procedures

Two dip switches (discussed below) are provided for setting the base address and any desired interrupt. The base address (S1) must be set before installing the card, but interrupt (S2) use is optional. (Refer to Figure 2-1 and Figure 2-3 for switch locations.)

Selecting the Base Address

The board's base address must be selected prior to installing the reader card. The base address is set on S1 (Figure 2-1) before the card is installed to identify the reader card to the CPM. Base addresses are defined in steps of four bytes and can span from 00 to 3FC hex.

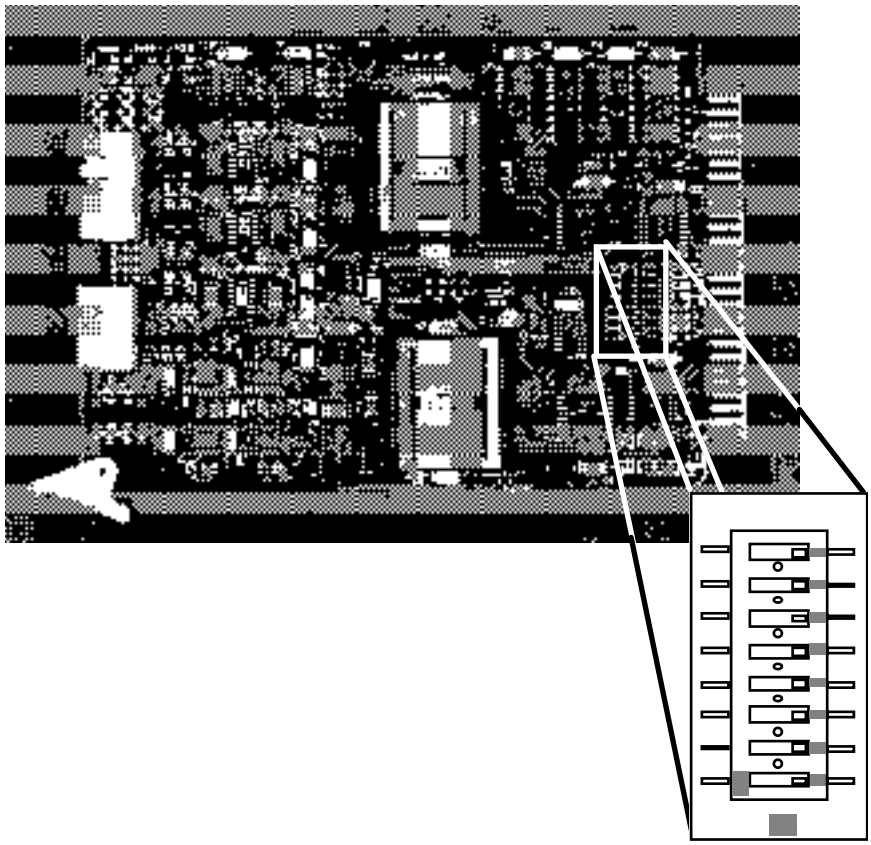

Figure 2-1 S1 dip switch

Moving an S1 lever to the "ON" position sets that bit to 0. The base address least significant bit (LSB) is at switch position 1. Each base address can be used only once on a bus. Base addresses correspond with bits 2-9 of the bus address, so S1 selects every fourth bus address. Each reader card uses one base address and the three consecutive, higher bus addresses. The base addresses can span from 00 hex to 3FC hex.

Caution
Care must be taken to choose a base address which does not interfere with the base addresses of other system cards.

Setting the I/O Expansion Jumper

The I/O expansion jumper block (JP2) should be connected in the EXPEN position (between the middle pin and the pin closest to the bus edge card fingers - Figure 2-2) when used with processor cards which require the use of the IOEXP signal to decode I/O space.

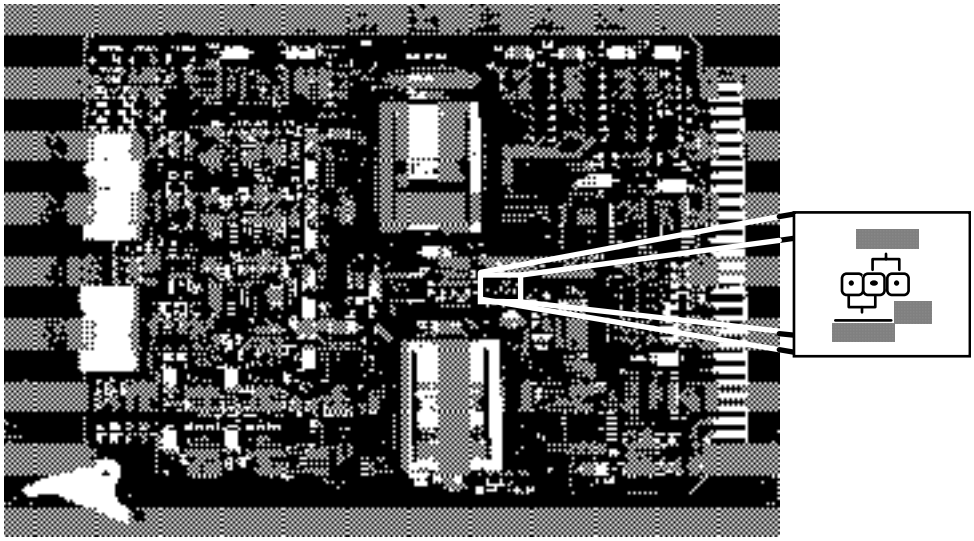

Figure 2-2 I/O expansion jumper

If the processor card does not require or generate this signal, the jumper should be placed in the **EXPEN** position (between the middle pin and the pin closest to the top of the card).

Selecting an Interrupt

If desired, an NMI or IRQ interrupt can be selected on reader card switch S2 (Figure 2-3). The interrupt will occur on a tag ID acquisition.

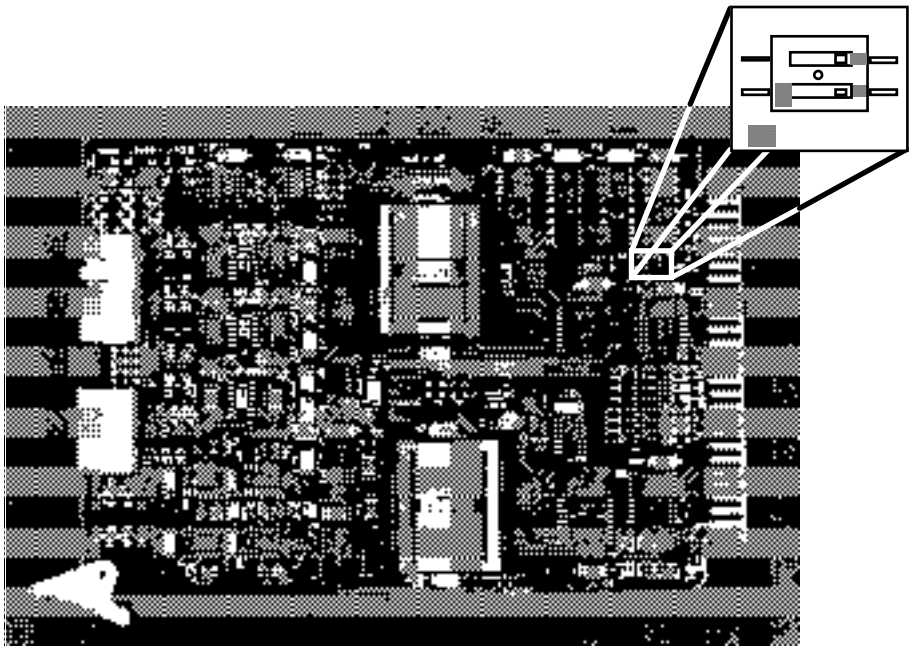

Figure 2-3 S2 dip switch

Moving a lever to the “ON” position selects an interrupt. The NMI position is at switch position 1 on S2. The IRQ position is at switch position 2.

If no interrupt is selected on S2, the CPM serving that reader card can poll the status register at intervals to determine if there is a tag ID in the reader card FIFO queue.

Installing a Reader Card

Be sure the base address and interrupt enable switches (discussed above) have been properly set **before** proceeding to this section.

1. **Turn off** all power to the target system.
2. **Install** the reader card in the target system card rack.
3. **Connect** the external components, RF module(s), and any ancillary I/O signals.
4. **Turn on** the power to the target system.

The target system will boot and accept software instructions.

3

Reader Card Operation

Chapter 3

Reader Card Operation

The reader card executes instructions sent to it from a CPM and allows status information to be read by the CPM. These instructions provide control of the reader system.

The card is identified to the CPM on the system bus by a unique, manually-selected base address (refer to the Reader Card Installation Procedures section for information on selecting the base address). The base address and the three higher consecutive addresses are used to access four pairs of registers on the card. Each address accesses a read-only register and a write-only register.

The AI1301 includes a FIFO queue for buffering data read from tags. The reader card's FIFO registers contains 512x9 bits.

The reader card can be configured to generate an interrupt, signaling a tag has been received and decoded. Or, the CPM can poll the reader card status registers to determine whether a tag is pending in the reader card's FIFO queue.

CPM Interface

At least one CPM must be installed on the bus with the reader card(s). The CPM can be any commercially available processing device of the user's choice, with CPM software developed by the user. The card interfaces with the end-user-developed software running on the CPM. While this software is not specified or supplied as part of the reader card system, code fragments are provided in the Appendix C of this guide.

The CPM can send commands to the reader card command registers and read status information from the status registers.

Tag Communication Protocol

The tag-to-reader card communication protocol is an FSK protocol identical to that used by the AI1200 reader. This protocol ensures compatibility between the read-only and the read/write systems. Modulated RF reflected by the tag is resolved into ones and zeros by the card decoding circuitry. Information specific to the system is contained in the data transferred from the tag to the card. In a read-only system, this information is the 128-bit frame that contains the permanent tag ID.

Command Syntax

All commands are hexadecimal bytes written to the command register at port address

Base+1. The AI1301 supports the following commands:

01	=	Turn RF O On
02	=	Turn RF O Off
03	=	Turn RF 1 On
04	=	Turn RF 1 Off
05	=	Enable Auto Mux Mode
06	=	Auto Mux Mode Disable
07	=	Perform RF Channel Force
08	=	Perform RF Channel Freeze
09	=	Tag Interrupt Enable
0A	=	Tag Interrupt Disable
0B	=	Clear Read FIFO
0F	=	Card Reset (resets to same states as SYSRESET*)

Refer to the Command Registers section for more detail on each command.

Response Syntax

The AI1301 does not send a response to commands, however, bits in the status registers at port addresses Base+1 and Base+2 can be read to indicate the effects of the commands sent (i.e., RF status and FIFO status).

Tag Read Transactions

Tag data is demodulated by the RF source/preamp assembly into an intermediate frequency pulse train. The reader card's differential amplifiers convert these to logic levels. Extensive signal processing and error detection circuitry decodes these signals, and if it is determined that a 128-bit frame has been received, it is stored in the read FIFO as 16 consecutive bytes with the MSB (D7) of the first byte holding the first bit, D6 the next, through D0, then the next bit in the MSB of the next byte, until all 128 bits are stored.

A read transaction can either be initiated as a response to an interrupt request, or as the result of a polling routine finding the READ FIFO EMPTY flag in the STAT0 register (bit D0) equal to 1, indicating that the read FIFO is not empty. To enable the incoming frame data to trigger an interrupt request, switch S2 must be set properly and an ENABLE IRQ command (09) must have been sent to the command register as part of the card initialization. No interrupt prioritization is provided, so in a multi-card system

polling to determine the source of the interrupt will be required.

A flag in the STAT0 status register at Base+1 indicates that the read FIFO is no longer empty or, optionally, an interrupt can be generated indicating data is available. These data bytes are read at port address Base+0 to obtain the tag data.

The read FIFO empty flag (bit 0 of STAT0) will read as a 1 (0 indicates empty) after the first byte is written into the FIFO. It is possible for the processor in some systems to empty the FIFO faster than it is being filled, so care must be taken to monitor the empty flag as the FIFO is being read to avoid reading erroneous data. Refer to Appendix C — Sample C Routines for this process.

If the system latency is such that the FIFO fills faster than it is emptied, eventually a flag in the STAT0 register (bit 1 of STAT0) will indicate that the FIFO is full and the FIFO will quit accepting tag data. All data after this point will be lost until the FIFO is read, clearing some of its memory space. You must monitor this flag to prevent the possibility of acquiring partial tag frames. To aid you in this case, bit 2 in the STAT0 register indicates that the last byte read from the FIFO is the sixteenth byte of a frame (i.e., data is on a frame boundary). If 16 bytes have been read and this flag is not set, synchronization of data acquisition has been corrupted and the user software must take into account that the data is suspect.

The sixteenth byte of frame data indicated by the framing bit in the STAT0 register contains two special bits of information. In bit position 126 (D1 of byte 16) LINK is indicated. If this bit is a 1 it means that the frame was acquired contiguous with one previous frame. That is, there were no error dropouts between this frame acquisition and the acquisition of the previous frame. This is an indication that this and the previous frame were acquired from the same tag. Careful use of this bit can help in the discrimination of cross-lane reading situations, and is useful in the reading of two-frame IntelliTags. Bit 127 (D0 of byte 16) is set to 0 if the tag was acquired while RF channel 0 was active, and is set to 1 if the tag was acquired while RF channel 1 was active.

Tags following the TransCore standard programming format contain checksum bits in bit positions 60, 61, 124, and 125. To obtain optimal system accuracy, the user software should compare these bits to a calculated checksum on the frame data, and discard frame data where these do not match. Refer to Appendix C — Sample C Routines for an example of this process and the conversion into 20 standard ASCII characters.

Bits in bit positions 62 and 63 indicate the tag type: IntelliTag or read-only. Read-only tags will have 11 in those bit positions. The IntelliTag ID frame will always have 00, and the IntelliTag variable frame will have a 01 or 10.

A bit in the STAT0 register (D3) indicates the read FIFO is currently being loaded with frame data. The user can concurrently be reading data from the FIFO as it is being loaded and monitor the empty/full status bits to synchronize the unload, or can monitor this bit and quickly cycle through 16 byte reads when it shows that the load is complete.

I/O Port Usage

Sense input and status output functions can be controlled by the AI1301's bit-selectable I/O port. Eight surge-protected TTL I/O lines can be individually configured for input or output. The command and status register-accessible TTL lines can be used for various ancillary functions such as loop detection, gate control, and check tag control.

Two steps must be taken in order to use the general purpose I/O port:

1. Initialize the I/O line as an input or an output.

   To configure a line as an output, write a '1' to the corresponding bit in the data direction register at address Base+2. To configure a line as an input, write a '0' to the corresponding bit in the data direction register at address Base+2.

2. Read or write to the port at address Base+3.

   If a line is configured as an input, any write to the port will have no effect. If a line is configured as an output, any reads of the port will return the current value latched into the output register.

Output signals are latched and will remain in the state they are set to until changed. Inputs are not latched and so must be sampled often enough to capture the signals they are attached to.

Although the I/O lines have some signal protection and conditioning, it is strongly recommended that only short wiring runs be attached to these lines. It is also recommended that external opto-isolation be used on all lines to prevent system problems such as ground loops and static damage.

Registers

Registers on the board are organized as shown below.

Command (Write)	Status (Read)	Address
I/O Data Out	I/O Data In	BA + 3
I/O Data Direction	Status I	BA + 2
Command	Status O	BA = 1
Read FIFO	BA = Base Address	

Command Registers

Command registers on the board are organized as shown below.

I/O Data OUT D7,D6,D5,D4,D3,D2,D1,D0
If I/O line is defined to be "OUTPUT" the bit value will be latched on the corresponding I/O line. If the I/O line is defined as an "INPUT" the bit will be a don't care.

I/O Data Direction D7,D6,D5,D4,D3,D2,D1,D0
Writing a one to a particular bit sets the I/O as OUTPUT. Writing a zero to a particular bit sets the I/O as INPUT. Reset initializes to all INPUTs.

Command D7,D6,D5,D4,D3,D2,D1,D0
The command register on the reader card can support 256 commands. Supported commands for this register (listed in hexadecimal) are:

01 —Turn RF0 On
This command forces RF channel 0 to become the active channel. The switch is permanent only if used while Auto Mux mode is disabled. The RF0 LED on the card edge illuminates after the Turn RF0 On command successfully implements.

02 —Turn RF0 Off
This command inhibits RF channel 0 and extinguishes the RF0 LED on the card edge.

03 —Turn RF1 On
This command forces RF channel 1 to become the active channel. The switch is permanent only if used while Auto Mux is disabled and RF channel 0 is not active. The RF1 LED on the card edge illuminates after Turn RF1 On successfully implements.

04 —Turn RF1 Off
This command inhibits RF channel 1 and extinguishes the RF1 LED on the card edge.

05 —Enable Auto Mux Mode
Auto Mux mode causes the reader card to alternately activate RF channel 0 and RF channel 1, to check for the presence of a tag. The active channel changes approximately every 3.2 milliseconds (a 32-bit period). If a tag is detected with "SNIFF" while Auto Mux is enabled, the reader card dwells on the detecting channel to capture an entire tag frame.

06 —Auto Mux Disable
This command disables the Auto Mux mode, allowing you to permanently activate either RF channel 0 or RF channel 1.

07 **—RF Force**
If the reader is in auto mux mode but dwelling on a channel due to the presence of a tag, RF Force will activate the alternate RF control line to check for the presence of a tag.

08 **—RF Freeze**
This command forces the RF control mechanism to stay (dwell) on the current channel during auto mux mode.

09 **—Tag Interrupt Enable**
Tag interrupts can be generated as an IRQ or an NMI, depending upon which lever (if any) of DIP switch S2 on the reader card, is on. If both S2 levers are off, this command does not generate an interrupt. An interrupt can be provided to the bus, only if:

- The hardware is correctly configured.
- DIP S2 is set to select one interrupt.
- Tag interrupt enable is asserted.

0A **—Tag Interrupt Disable**
This command allows you to disable interrupts at any time, regardless of the position of DIP switch S2 on the reader card.

0B **—Clear Read FIFO**
This command resets the Read FIFO to its first data position. All data in the Read FIFO is destroyed when this command is used.

0F **—Card Reset**
This command resets the contents of the card to the same states as SYSRESET*:

RF0 Off
RF1 Off
Auto Mux Off
Interrupt Disabled
FIFO Cleared

Status Registers

Status registers on the board are organized as shown below.

I/O Data IN **D7,D6,D5,D4,D3,D2,D1,D0**
If I/O line is defined to be "OUTPUT" the bit will reflect the state of the latched output. If the I/O line is defined as an "INPUT" the bit will reflect the state of the input signal.

Status Register 1 D7,D6,D5,D4,D3,D2,D1,D0
Bit definitions for this register are:

D7 —Reserved by TransCore for internal use.

D6 —Reserved by TransCore for internal use.

D5 —Reserved by TransCore for internal use.

D4 —1 indicates "LOCK" is active and that the system has acquired a valid tag.

D3 —1 indicates RF1 is currently active.

D2 —1 indicates RF0 is currently active.

D1 —1 indicates auto mux mode is enabled.

D0 —1 indicates "SNIFF" is active. This indicates a tag is in the field.

Status Register 0 D7,D6,D5,D4,D3,D2,D1,D0
Bit definitions for this register are:

D7 —Not applicable.

D6 —0 indicates write capability disabled.

D5 —Not applicable.

D4 —Not applicable.

D3 —1 indicates load is in progress.

D2 —1 indicates on frame byte.

D1 —0 indicates read FIFO is full.

D0 —0 indicates read FIFO is empty.

Read FIFO (RDFF) D7,D6,D5,D4,D3,D2,D1,D0
D7 is the first bit from tag and D0 is the last.

Status Register Bit Definitions

Two status registers are provided on the reader card. Each status register bit reflects the state or condition of a particular function of the reader card. These bits are described below.

Status Register 0

Bit 0 "0" indicates that the Read FIFO is empty.

"1" indicates that valid tag data (to be read by the CPM) is present in the Read FIFO.

Bit 1 "0" indicates that the Read FIFO is full and thus that tag data loss is possible.

A "1" means that the Read FIFO has not been overloaded.

Bit 2 "1" indicates an "on frame" byte. "On frame" is valid after the 16th byte of a frame is clocked out of the FIFO and remains 1 until another byte is clocked out of the FIFO.

Note: Bits 126 and 127 in the 16th byte of a frame are used to indicate the LINK status and to identify the channel on which a tag is detected. Bit 126 of this byte indicates the “LINK” status, and is a “1” if two tags have been read consecutively with no errors. (This information is useful when looking at a particular site for dropouts, etc.) Bit 127 identifies the RF channel on which a tag was detected. A “0” here means that RF channel 0 detected the tag, and a “1” means that RF channel 1 detected it.

Bit 3 "1" indicates that a read FIFO load is in progress.

Bit 4 Reserved by TransCore for internal use.

Bit 5 Reserved by TransCore for internal use.

Bit 6 "0" indicates a read-only card.

Bit 7 Reserved by TransCore for internal use.

Status Register 1

Bit 0 “1” indicates SNIFF is active.

Bit 1 “1” indicates auto mux mode is enabled.

Bit 2 “1” indicates RF0 is currently active.

Bit 3 “1” indicates RF1 is currently active.

Bit 4 “1” indicates LOCK is active.

Bit 5 Reserved by TransCore for internal use.

Bit 6 Reserved by TransCore for internal use.

Bit 7 Reserved by TransCore for internal use.

A

Reader Card Specifications

Appendix A

Reader Card Specifications

Tag ID storage (volatile):	512 x 9 bits, FIFO (for read/write)
Operating temp. range:	-40° to +85° C
Humidity:	Noncondensing only
Mounting orientation:	Any
Size (approx):	6 x 4.5 inches
Weight (approx.):	0.25 lb.
Field Termination:	Field wiring
DC power Input Voltage: Current drain:	 Regulated +5 VDC, from bus. Approx. 325 mA.
Switches:	S1 Base Address Select S2 Interrupt Select
LED Indicators:	BOARD SELECT LOCK RF 0 ON RF 1 ON
Bus format:	STD (see Appendix B)
Connectors:	J2 (serves I/O interface board) J3 (serves RF interface board) Bus (56 pins), Double-sided
RF channel capacity:	Two RF control channels

RF Control	14-Pin IDC Header. Pins defined as follows: Pin 1 IF Channel A 1 Pin 2 IF Channel A 2 Pin 3 IF Channel B 1 Pin 4 IF Channel B 2 Pin 5 IF Channel C 1 Pin 6 IF Channel C 2 Pin 7 Ground Pin 8 Ground Pin 9 RF 0 - Pin 10 RF 0 + Pin 11 RF 1 - Pin 12 RF 1 + Pin 13 MODULATE - Pin 14 MODULATE +
Status Input/Outputs:	Eight TTL user-definable input/output lines (non-isolated, surge-protected). Ten-pin IDC header. Pins defined as follows: Pin 1 I/O Data Bit 0 Pin 2 I/O Data Bit 1 Pin 3 I/O Data Bit 2 Pin 4 I/O Data Bit 3 Pin 5 I/O Data Bit 4 Pin 6 I/O Data Bit 5 Pin 7 I/O Data Bit 6 Pin 8 I/O Data Bit 7 Pin 9 +5VDC* Pin 10 Ground

* +5VDC is intended for diagnostic purposes and should NOT normally be connected.

B

J1 Edge Connector Pin-Out to STD Bus

Appendix B

J1 Edge Connector Pin-Out to STD Bus

	Component Side				Circuit Side			
	Pin	**Mnemonic**	**Signal Flow**	**Description**	**Pin**	**Mnemonic**	**Signal Flow**	**Description**
Logic Power Bus	1 3 5	+5VDC GND VBATT	In In In	Logic Power Logic Ground Battery Power	2 4 6	+5VDC GND DCPWRDWN*	In In In	Logic Power Logic Ground DC Power Down
Data Bus	7 9 11 13	D3 (A19) D2 (A18) D1 (A17) D0 (A16)	In/Out (Out) In/Out (Out) In/Out (Out) In/Out (Out)	Low-Order Low-Order Low-Order Low-Order	8 10 12 14	D7 (A23) D6 (A22) D5 (A21) D4 (A20)	In/Out In/Out In/Out In/Out	High Order High Order High Order High Order
Address Bus	15 17 19 21 23 25 27 29	A7 A6 A5 A4 A3 A2 A1 A0	Out Out Out Out Out Out Out Out	Low-Order Low-Order Low-Order Low-Order Low-Order Low-Order Low-Order Low-Order	16 18 20 22 24 26 28 30	A15 (D15) A14 (D14) A13 (D13) A12 (D12) A11 (D11) A10 (D10) A0 (D9) A8 (D8)	Out Out Out Out Out Out Out Out	High Order High Order High Order High Order High Order High Order High Order High Order
Control Bus	31 33 35 37 39 41 43 45 47 49 51	WR* IORQ” IOEXP INTRQ1* STATUS1* BUSAK* INTAK* WAITRQ* SYSRESET* CLOCK* PCO	Out Out Out In Out Out Out In Out Out Out	Write Mem or I/O I/O Address Select I/O Expansion Interrupt Req 1 CPU Status 1 Bus Acknowledge Int Acknowledge Wait Request System Reset Processor Clock Priority Chain Out	32 34 36 38 40 42 44 46 48 50 52	RD* MEMRQ* MEMEX MCSYNC* STATUSO* BUSRQ* INTRQ* NMIRQ* PBRESET* CNTRL* PCI	Out Out Out Out Out In In In In In/Out In	Read Mem or I/O Mem Address Select Memory Expansion CPU Mach Cycle Sync CPU Status O Bus Request Interrupt Request Nonmaskable Int Push-button Reset AUX Timing Priority Chain In

	Component Side				Circuit Side			
	Pin	**Mnemonic**	**Signal Flow**	**Description**	**Pin**	**Mnemonic**	**Signal Flow**	**Description**
Auxiliary Power Bus	53	AUX GND	In	AUX Ground (bussed)	54	AUX GND	In	AUX Ground (bussed)
	55	AUX+V	In	AUX Positive (+12V DC)	56	AUX-V	In	AUX Negative (-12V DC)

* Low-level active indicator.

Note Address lines A16-A19 are multiplexed on data lines D0-D3 on each address cycle. Typically PCO and PCI are not used on peripheral cards and should be connected together.

C

Sample C Code

Appendix C

Sample C Code

```
#include <stdlib.h>
#include <stdio.h>
#include <io.h>
#include <dos.h>
#include <string.h>

/* Reader command definitions */
#define RF0_ON   0x01
#define RF0_OFF  0x02
#define RF1_ON   0x03
#define RF1_OFF  0x04
#define AUTOMUX_ON0x05
#define AUTOMUX_OFF0x06
#define RF_FORCE 0x07
#define RF_FREEZE0x08
#define TAG_IRQ_ON0x09
#define TAG_IRQ_OFF0x0a
#define CLR_RD_FIFO0x0b

/* Status register 0 mask value definitions */
#define RD_FF_EM 0x01
#define RD_FF_FULL0x02
#define FRM_BYTE 0x04
#define LOAD_BSY 0x08
#define RD_WRITE 0x40

/* Status register 1 mask value definitions */
#define SNIFF    0x01
#define MUX_MODE 0x02
```

```
#define RF0      0x04
#define RF1      0x08
#define LOCK     0x10

/* External STD card bus I/O memory map variables */
unsigned int rdr_fifo;/* Initialize to base address +0 */
unsigned int rdr_status_0;/* Initialize to base address +1 */
unsigned int rdr_command;/* Initialize to base address +1 */
unsigned int rdr_status_1;/* Initialize to base address +2 */
unsigned int rdr_ddr;/* Initialize to base address +2 */
unsigned int io_data;/* Initialize to base address +3 */

/* External buffer variables */
unsigned char frame_buf[16];
unsigned char char_buf[25];
/*********************************************************************
Get a frame from the read fifo - if 16 bytes not available (the fifo is empty but
no load in progress) return error. Also checks the Frame byte status bit to insure
we are on frame boundary on exit. A sixteen byte data frame is loaded into the
external buffer frame_buf if succesful. Frame_buf may be corrupted if unsuccess-
ful. *********************************************************************/
int get_frame()
{
                int i;
                for(i=0;i<16;i++)
                {
                while(((inportb(rdr_status_0)) & RD_FF_EM) == 0)
                {
                if(((inportb(rdr_status_0)) & LOAD_BSY) == 0)
if(((inportb(rdr_status_0)) & RD_FF_EM) == 0)
{
/* A frame length error has occured */ return(-1);
                }
                }
                frame_buf[i] = inportb(rdr_fifo);
```

```
                }
                if(((inportb(rdr_status_0)) & FRM_BYTE) == 0)
                {
                /* A framing error has occured */
                return(-1);
                }
                /* Frame_buf now holds a valid frame */
                return(0);
}
/*************************************************************************
Takes the sixteen bytes from frame_buf encoded with TransCore 6-bit ascii and con-
vert to a twenty character string in char_buf - bits 60,61 (chksum) 62,63 (inner
frame bits) 124,125 (chksum) 126,127 (end frame marker) are ignored. Terminate
char_buf with a 0.
*************************************************************************/
void conv_TransCore_to_ascii()
{
                unsigned char *p;
                unsigned char *q;
                unsigned char temp0,temp1;
                int i,j;

                p = frame_buf;
                q = char_buf;

                for(j = 0;j < 2;j++)
                {
                for(i = 0;i < 2;i++)
                {
                temp0 = (*p & 0xfc);
                *q++ = ((temp0 >> 2) + 0x20);

                temp0 = ((*p++ & 0x03) << 4);
                temp1 = ((*p & 0xf0) >> 4);
                *q++ = ((temp1 | temp0) + 0x20);
                temp0 =(*p++ & 0x0f) << 2;
```

```
                temp1 =(*p & 0xc0) >> 6;
                *q++ = ((temp0 | temp1) + 0x20);

                *q++ = ((*p++ & 0x3f) + 0x20);
}
                temp0 = (*p & 0xfc);
                *q++ = ((temp0 >> 2) + 0x20);
                temp0 = ((*p++ & 0x03) << 4);

                temp1 = ((*p++ & 0xf0) >> 4);
                *q++ = ((temp1 | temp0) + 0x20);
                }
                *q = 0;
}
/*********************************************************************
Check for proper chksum bits (60,61,124,125) for the frame in framebuf.
These bits should be checked with every frame read.
*********************************************************************/
int chk_chksum()
{
                int i;
                unsigned char count = 0x00;
                unsigned char count1 = 0x00;
                unsigned char lookup[] = {0x00, 0x01, 0x01, 0x02, 0x01, 0x02,
0x02, 0x03,
                0x01, 0x02, 0x02, 0x03, 0x02, 0x03, 0x03, 0x04};
                for(i=0;i<7;i++)
                {
                count += lookup[((frame_buf[i] & 0xf0) >> 4)];
                count += lookup[(frame_buf[i] & 0x0f)];
                count1 += lookup[((frame_buf[i+8] & 0xf0) >> 4)];
                count1 += lookup[(frame_buf[i+8] & 0x0f)];
                }
                count += lookup[((frame_buf[i] & 0xf0) >> 4)];
                count1 += lookup[((frame_buf[i+8] & 0xf0) >> 4)];
```

```
                count1 += lookup[(frame_buf[7] & 0x03)];


                if(((frame_buf[7] & 0x0c) == ((count & 0x03) << 2)) &&
                ((frame_buf[15] &0x0c) == ((count1 & 0x03) << 2)))


                return(0);
else
                return(-1);
}
```