Updated: 05/2025

Support Document

SDS-PCI-2 Interface Card
User's Guide

PK 80137 Issue 2B

For application assistance, contact InfiniDrive™ Motor Mfg.at 1-616-965-9898

CHAPTER 1 1

Introduction 1
About this User's Guide 1
Sales and Service 3
Internet Addresses 3
Warranty/Remedy 3

CHAPTER 2 4

Installation 4
System Overview 4
PC to PCI Card Interface 5
Control 5
Bus Channels 5
Order Guide 5
PC Card Layout and Configuration 5
Connectors and Cabling 5
Connector Pin Assignments 6
Termination Resistors 6
Card Installation 7
PC to System Cabling 7
Hardware Initialization Instructions 8
Software Installation and Configuration 9
What is on the Configuration Disk 9
Installing the Configuration Software 9
Specifications 11
Host System Requirements 11
PCI Bus Requirements 11
PC Interface Card Specifications 11

CHAPTER 3 12

DLL Interface 12
Operation Order 12
API Services 13
Cfg Bld 13
Cfg Dev Status 13
Cfg Sys Status 13
Cfg2Cfx 14
Change Address 14
Clear Device Error 14
Create Handle 14
Destroy Handle 14
Device Action 14
Device Present 14
Device Prim Tag 14
Download APL 14
Finish 14
Generate Unique Addresses 14
Get Attr 15

Get Available Interfaces 1
Get Bus Error 1
Get Bus Error Description 1
Get Device Error 1
Get Device Error Description 1
Get I/O Descriptor 1
Get I/O Type 1
Get Info 1
Get Privilege 1
Get Returncode Description 1
Get Supported Bus Type 1
Get Supported Channel 1
Get Supported Privilege 1
Get Supported Speed 1
Init 1
Is Analog 1
Multi-Errors 1
Multi-Read Input 1
Multi-Read Output 1
Multi-Write Output 1
Read Input 1
Set Attr 1
System Info 1
Watchdog Control 1
Watchdog Status 1
Watchdog Sustain 1
Write Output 1

CHAPTER 4 1

Configuration File 1
Start Up the PC Interface Card 1
System Configuration File Syntax 1
Case 1
Special Character 1
Organization 2
System Configuration Commands 2
Board nnnn 2
Baud 2
Channel 1 and 2 2
Address 2
Replacement_Address 2
Hierarchy 2
Input 2
Output 2
Attr 2
Example #1 2
Example #2 2

Log File and Error Messages 27
The Anatomy of the Log File 27
Example Log File 27
Information Messages 28
Date: *nn/nn/nn* Time: *nn:nn:nn* 28
File: *name* Line: *number* 28
Honeywell - MICRO SWITCH System Initialize In Progress (Version:*nn*) 28
Initialize the PC Interface Board: *nnnn Bus#n: nnn Bus#n: nnn* 28
Parameter: *nnn* 28
System Initialization Complete -- No Errors. 28
Error Messages 29
Initialization Error Message 29
Errors Found in Smart Distributed System Initialization 29
Initialize Error: *nnnn* 29
<<<Unknown Channel Error Bit>>> 29
Device Error Messages 33
Device Error for Address: *nn* Error: *nnn* 33
Configuration Error Messages 38
Board: *nnn* Channel: *nn* Address: *nn* 38
Duplicate Configuration Files Specified 39
Error in Specified Board Address 39
Error Initializing Interface Board Channel *nn* Status: *nnnn* 39
Error in Initialize Write Status: *nn* 39
Extra Device on the Bus 39
Error Invalid Channel Number 39
Error Processing Line: <*line from file*> 39
Error Reading System Bus errors for Channel *nn* Status: *nnn* 39
Error -- NO Parameters passed to STARTSDS 39
Incorrect Attribute Value Attr: *nnn* 39
Initialize Errors for Channel *nn* 40
Invalid Automatic_attribute_update parameter 40
Invalid Device Address 40
Invalid Option Specified: *nnnnn* 40
Invalid Speed Selected -- MUST BE 125, 250, 500 or 1000 40
Missing Address Information. 40
Missing Board Parameter 40
Missing Channel Information 40
Missing Device 40
No Configuration File Specified 40
Unable to Open Configuration File: *nnn* 41
Summary Screen 41

Introduction

About this User's Guide

This User's Guide for the Smart Distributed System PCI Interface Card is organized into a Table of Contents, five chapters, and an index. It contains the information needed to install and configure the PCI Interface Card. Throughout the User's Guide important information is highlighted with the NOTICE banner as shown in the example below:

NOTICE

Failure to read this User's Guide may cause undue delays in implementation of the PCI Interface Card.

Please take special note of these highlighted portions. They will help make your installation of the PCI Interface Card faster and easier.

Chapter 1 This chapter contains organizational information about this User's Guide, well as information on sales and service and product warranties.

Chapter 2 This chapter contains an overview of the PCI Interface Card and in-depth information on the setup and operation.

Chapter 3 This chapter contains information regarding the DLL interface

Chapter 4 This chapter explains how to setup and use the PCI Interface Card configuration file.

Chapter 5 This chapter explains the organization of the log file and details the messages that appear in both the log file and on the screen.

Index

Sales and Service

Holjeron Corp. provides application and sales assistance to SDS customers directly through its Customer Service Department. For application assistance, current specifications, pricing or name of the nearest Authorized Distributor contact:

Phone: (616) 965-9898

Toll-free: (877) 415-9898

General info: info@infinidrivemotors.com

Sales: sales@infinidrivemotors.com

Support issues: support@infinidrivemotors.com

Specifications may change at any time and without notice. The information we supply is believed to be accurate and reliable as of this printing. However, we assume no responsibility for its use.

While we provide application assistance, personally and through our literature, it is up to the customer to determine the suitability of the product in the application.

Internet Addresses

To request information on the Smart Distributed System via the Internet send an e-mail to:

info@infinidrivemotors.com

Or, if you have access to the World Wide Web, point your browser to:

http://www.holjeron.com

Warranty/Remedy

Holjeron warrants goods of its manufacture as being free of defective materials and faulty workmanship. Commencing with date of shipment, Holjeron's warranty runs for 18 months. If warranted goods are returned to Holjeron during that period of coverage, Holjeron will repair or replace without charge those items it finds defective. The foregoing is Buyer's sole remedy and is **in lieu of all other warranties, express or implied, including those of merchantability and fitness for a particular purpose**.

Chapter 2

Installation

System Overview

The PCI Interface Card for the Smart Distributed System provides the interface between an IBM-compatible personal computer (PC) and two separate Smart Distributed System buses.

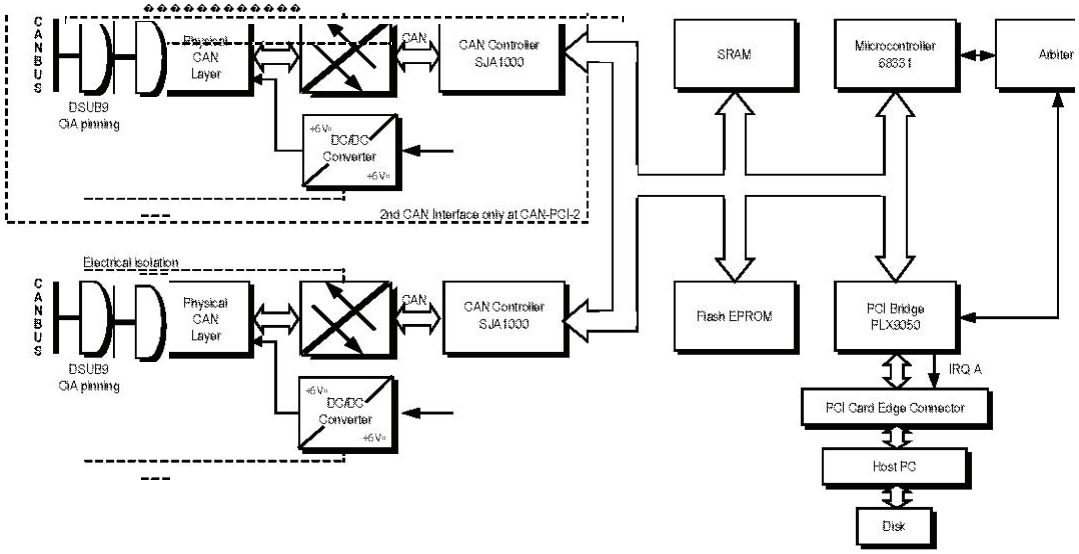


Figure 2-1 PCI Interface Card System Diagram

PC to PCI Card Interface

The hardware interface between the PC and the PCI Interface Card is the shared memory, located in the PCI Card SRAM.

Control

The control section contains the CPU and the RAM. The CPU oversees the operation of the card and the communication of both bus channels.

The Flash EPROM is where the application layer of the Smart Distributed System protocol is downloaded. At Smart Distributed System PCI Interface power-up time, the application layer software is moved to the SRAM and execution is started. This happens automatically with no user intervention required.

Bus Channels

There are two independent, optically isolated CAN channels on each PC Interface Card: bus one (B1) and bus two (B2). See Figure 2-3 PC Card Layout on page 5. Each channel has an individual driver, optical isolator, and a 9-pin Male D-sub Connector.

Order Guide

Catalog Listing Description

PC Interface Card

SDS-PCI-2 PC Interface Card Kit with one "PCI" PC Interface Card (RoHS compliant version of SDS-PCI-2). (Itools configuration software is available at www.holjeron.com)

Cables Available for Purchase Separately

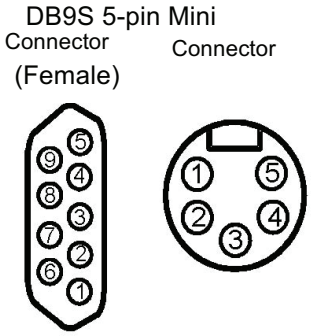
SDS-AFA-003 9-Pin Female D-sub Connector to 5-Pin Male Mini Connector cable, 3 ft.

PC Card Layout and Configuration

Connectors and Cabling

Connections to the two independent Smart Distributed System buses are made via the two 9-pin D-sub male connectors (DB9S) B1 and B2. Connection to the bus is made via a 5-pin Mini connector. See Figure 2-2 Pin Assignment and Connector Types for wiring details.

Connector Pin Assignments



5-Pin	9-Pin Mini D *	Wire Color	Function
1	-	Shield	Bus Shield
2	9	Brown	Bus Power (DC+)
3	3	Blue	Bus Power (DC -)
4	7	Black	Bus Comm (Bus+)
5	2	White	Bus Comm (Bus -)

* Pins not mentioned in this column have no connection.

Figure 2-2 Pin Assignment and Connector Types

Termination Resistors

NOTICE

Each bus must have two — and only two — termination resistors - one resistor at each extreme end of the trunk. (See Figure 2-5).

The PCI Interface Card has 120 Ohm terminating resistors for each bus. They are enabled or disabled by DIP switches near the connector edge and between the bus connectors. Disable the resistor if the PCI card is not at one end of the bus.

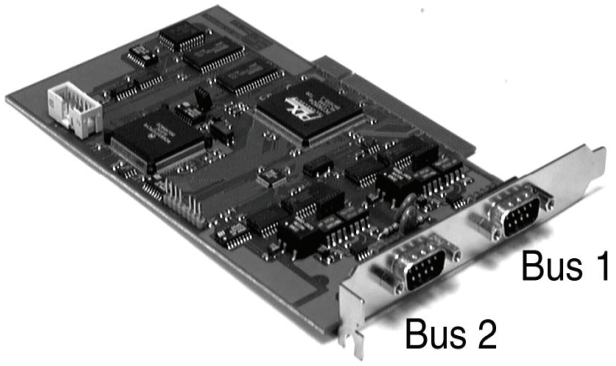


Figure 2-3 Bus Assignments

Each bus must be configured with a single trunk, with one termination resistor located at each extreme end of the trunk. The termination resistors are connected between the Bus + and Bus - wires (See Figure 2-4). If you are using hardwire cabling you will need two 120 Ohm, 1/2 Watt, 2% resistors. Use a Termination Cap (SDS-TERM or SDS-TERM-F) with the plug-and-play cabling system.

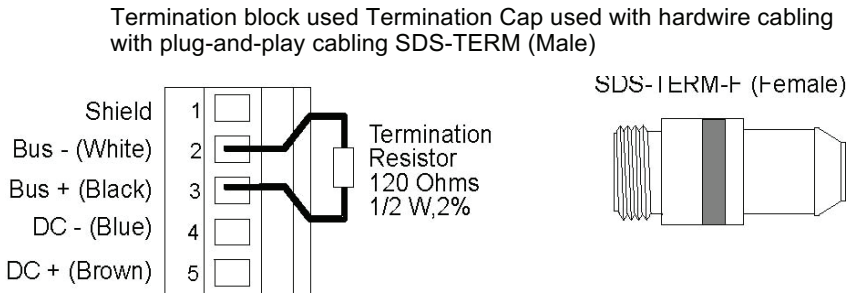


Figure 2-4 Termination for hardwire and plug-and-play cabling

Figure 2-5 shows the PC Interface Card connected to the middle of the bus using B1. It also shows the PC Interface Card terminating one end of a bus using B2.

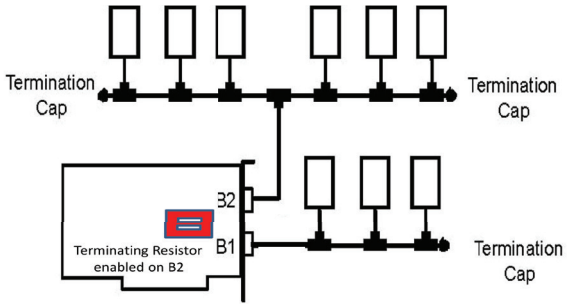


Figure 2-5 Examples Using the PC Card's Termination Resistor

Card Installation

Install the SDS-PCI-2 Card in a PCI expansion slot. Refer to your PC's documentation for expansion card installation procedures.

PC to System Cabling

Figure 2-6 shows one method of connecting the PC interface card to your Smart Distributed System bus or buses. You can use either the three foot 9-Pin Female D-sub Connector to 5-Pin Male Mini Connector cable (SDS-AFA-003) and one of the bulkhead cables (SDS-BKHD-001/003/006/010) (See Figure 2-6) or the 10 foot 9-Pin Female D-sub Connector to pigtails cable.

NOTICE

The user must provide the power supply for the Smart Distributed System bus power. Figure 2-6 PC to System Cabling Example shows the proper cabling to connect the bus and bus power to the system trunk.

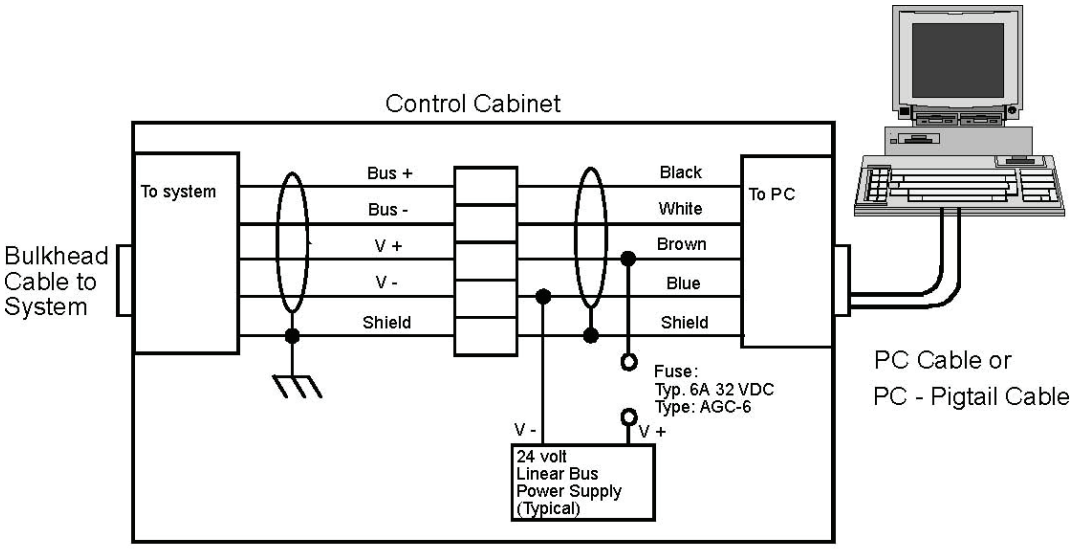


Figure 2-6 PC to System Cabling Example

Hardware Initialization Instructions

The iTools program allows you to initialize or upgrade the Smart Distributed System PCI Interface Card. A "new" SDS-PCI-2 card is preloaded with the embedded code, so iTools is not required for initialization, only for upgrading the embedded code or changing the board address. Consult www.holjeron.com for the latest version of iTools.

1. With the computer shut down, install the card(s) using standard precautions against ESD damage.
2. Power up the computer and run the iTools.exe program.
3. If more than one card is installed, the "Select Next" button allows you to step through the list of installed cards. Complete steps 4 and 5 below for each card installed. Label each card as you assign ID numbers.
4. The "SDS ID" field will indicate a value of 65535 for a new card. If multiple cards are used in a single computer, each card must have a unique SDS Card ID value. The "Change ID" button allows you to change the Card ID value to any number from 0 to 65535. The Card ID value is stored in nonvolatile memory on the card. It may be changed as needed with the iTools program. iTools does not allow you to use the same Card ID value on multiple cards.
5. Press the "Program" button to install the Smart Distributed System embedded code in the flash memory on the card. (The file, pci.s19, contains the code. This file must be in the same directory folder as iTools.exe.) The "Embedded Code Version" field will now display the code version that was just installed.

CAUTION: PC Manufacturers recommend power be removed before removing or installing PCI cards. Follow PC manufacturer's directions for safe installation procedures.

The "Verify" button compares the code in flash memory with the data in pci.s19 file and reports the results.

The "Program" function automatically performs a "Verify" after the actual programming sequence. It also resets the card after verification.

The "Reset Card" button resets (reboots) the card. Under normal conditions, this button is not necessary. It is included here as a convenience for troubleshooting.

Software Installation and Configuration

The Application Layer software that operates the "PCI" PC Interface Card must be downloaded to RAM on the PC Card at power-up or when the system is reset. The following section describes the software used to install and configure the "PCI" PC Interface Card.

NOTICE

You should read the "VERSION.TXT" file on the Configuration CD before installing this software. It contains the most current information.

What is on the Configuration Disk

At installation, the Configuration CD loads several files into the SDS_PCI directory it creates on your hard drive. These files include the Application Layer and the utilities to configure the PC Card.

There are example subroutines that can be used by software developers to access the shared memory. README.TXT has more information on these developer's tools.

Installing the Configuration Software

The Application Layer Configuration software supplied on the CD with this "PCI" PC Interface Card includes basic testing and troubleshooting tools. Most system control software requires its own software driver. Please refer to documentation included with your software for driver installation and user instructions.

To install the software on your PC (Windows 2000,Windows XP, or Windows7:

1. Insert the CD in the CD drive.
2. Click the START button and select RUN.
3. Type D:\SETUP (or E:\SETUP – change for your computer's CD drive "letter") and click OK. A directory called \SDS will be created on your hard drive and the files will be copied into it.
4. Installation is complete. Remove the CD.

NOTICE

SDS-PCI-2 requires iTools Version 6.0 or higher.

Specifications

Host System Requirements

Hardware	Intel® Pentium microprocessor or better running at 33 mHz or faster
Hard Drive Space	4 Megabytes
Interface	PCI Bus
Memory	4 Megabytes (8 recommended)
Software	Microsoft Windows 7, Windows 2000, or Windows XP OS

PCI Bus Requirements

Host Bus	PCI in compliance with PCI Local Bus Spec 2.1
PCI Data Bus	32 bit
Interrupt	Interrupt Signal A
Slot Position	No restrictions, PCI bridges are tolerated
Board Dimension	"Short" PCI board
Connector	PCI Card edge connector

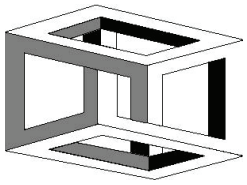
PCI Interface Card Specifications

System Bus	Number of Buses	2 Independent buses per Card
	Bus Voltage	11 to 25 VDC
	Addresses	126 maximum per bus channel
	Electrical Loads	64 maximum per bus channel
	Data Rates	125 Kbps typical (250 Kbps, 500 Kbps, and 1 Mbps available dependent on aapplication)
	Response Time	2 ms max.
Microcontroller	Type	68331
	Memory	128 k x 16 bits SRAM
		128 k x 8 bits Flash EPROM
Host Bus	Interface Type	PCI
	Interface bits	8
	Voltage Range	5 VDC
	Current Consumption	1 Amp. Maximum
Environmental	Temperature	Operating 0° to 70° C (32° to 158°F)
		Storage -40° to 85° C (-40° to 185° F)
	Humidity	10% to 90% RH, Non-condensing
	Shock	5 G
	Vibration	.25 G at 10 to 500 Hz
	Standards	FCC Article 15, Class A
		UL/CSA Class 2

Physical	Terminations	2, 9-pin "D" Connectors (Male)

Updated: 05/2025
Support Document

Chapter 3



DLL Interface

The Smart Distributed System DLL provides a standardized interface between the control software and the interface hardware. The DLL provides an API for accessing Smart Distributed System buses on a Windows NT platform via multiple types of Smart Distributed System interface hardware.

This API presents a C style interface to the Smart Distributed System buses. The actual implementation of the SDSIF DLL is accomplished using a C++ style object oriented approach.

NOTE: The following description of the DLL is provided for expert users or software driver developers who must interface directly at the DLL level. Users of packaged PC Control application software should ask their software supplier for a Smart Distributed System driver which supports the SDS-PCI-2 card and interfaces to the API and DLL services described below.

Operation Order

The order of operations when accessing a bus is:

1. Request a handle to the Smart Distributed System bus. The desired bus is indicated through parameters calling out the type of Smart Distributed System interface hardware, the location (address, ID) of the interface hardware, and the channel number to use on the interface.
2. Start the bus at a particular desired baud rate. If desired, system start-up behavior may be provided by use of a CFX file, and the results of the start-up may be logged to a data file.
3. Access the bus through a series of API calls that provide access to I/O, diagnostics, device attributes, and many other features.
4. Stop the bus in a controlled manner.
5. Destroy the handle to the bus.

Internally the Smart Distributed System DLL maintains knowledge about which buses are currently allocated. This allows buses to be shared.

Because a single Smart Distributed System interface card may contain multiple channels, the linkage between the bus object and the actual interface hardware is somewhat complex. It is necessary for multiple Smart Distributed System bus objects to communicate via the same piece of hardware. Some driver utilities, in particular WinDriver, lock the interface hardware when a communications link is established. This locking process prevents other entities from accessing the hardware. Because of this locking process, it may be necessary to use the same hardware link for multiple bus objects. The handling of these hardware links is handled via the Hardware Linkage Factory object. It maintains knowledge about which hardware links are in existence.

Requests for Smart Distributed System bus handles result in the generation of bus interface objects. When new Smart Distributed System interface hardware is developed, appropriate new bus objects would also be developed, if necessary, to interface with this hardware.

API Services

The following list provides a brief description of each API service. The actual SDSIF DLL API calls are described in SDSIF.h and SDSIFDEF.h.

Cfg Bld

Builds a text based CFG (configuration) file from the current Smart Distributed System bus. This CFG file may be used during enrollment to ensure that the state of the bus is the same as it is was when the CFG file was created. This service can be used with the bus running or not running. In either case, the bus will be returned to its initial state (running or not running) when the build service is completed.

Cfg Dev Status

Indicates the device level results of the Smart Distributed System bus initialization utilizing a configuration file. For a selected Smart Distributed System device, a mask allows status inquiry of any group of the first 32 attributes. Indications for each attribute are "Attribute was updated successfully," and "Attribute update attempt was unsuccessful".

Cfg Sys Status

Indicates the system level results of the Smart Distributed System bus initialization utilizing a configuration file. Indications are any missing or extra devices, any completed attribute updates, and any uncompleted attribute updates (e.g., missing device, read-only attribute, etc.).

Cfg2Cfx

Converts a text based CFG (configuration) file to a binary CFX file that may be downloaded to the Smart Distributed System interface to guide the initialization process. See Smart Distributed System documentation about CFG and CFX files in Chapter 4 of this User's Guide.

Change Address

Changes the address of a Smart Distributed System device. The bus must be re-enrolled before the controller can recognize any new addresses.

Clear Device Error

Clears any/all diagnostic indications from an individual Smart Distributed System device.

Create Handle

Generates a handle that is used to refer to a particular bus.

Destroy Handle

Releases a previously requested handle.

Device Action

Initiates a specified action within a Smart Distributed System device.

Device Present

Indicates whether or not a Smart Distributed System device is enrolled on the system.

Device Prim Tag

Reads the primitive tag of a specified attribute of the Smart Distributed System device.

Download APL

Downloads application layer code (interface card embedded code) to the Smart Distributed System interface card. This service is only supported by certain interface cards that require a code download.

Finish

Stops a Smart Distributed System bus from executing. (Note that finishing a bus does not release its handle. Use the Destroy Handle service.)

Generate Unique Addresses

Resolves any duplicate addresses that may be present on a Smart Distributed System bus. Returns the original and final addresses of each device that had its address changed, as well as the Vendor ID and Serial Number of the device.

Get Attr

Reads data from a specified attribute of a Smart Distributed System device.

Get Available Interfaces

Returns a list of installed Smart Distributed System interfaces of a specified type.

Get Bus Error

Obtains any system level diagnostics that might be present. The act of reading these system level diagnostics clears the diagnostic indication.

Get Bus Error Description

Returns a text description of a specified Smart Distributed System level diagnostic value.

Get Device Error

Obtains any device level diagnostics that have been reported by a Smart Distributed System device. The act of reading the diagnostic indication does NOT clear the diagnostic indication in the device. This must be done with the API Service, Clear Device Errors.

Get Device Error Description

Returns a text description of specified Smart Distributed System device level diagnostic value(s). If multiple bits are selected, each bit is described within comma delimiters.

Get I/O Descriptor

Returns a pair of I/O descriptors that may be used to interpret the actual I/O data for a given Smart Distributed System device.

Get I/O Type

Indicates the number of input and output bits that are used by a Smart Distributed System device.

Get Info

Returns information about the Smart Distributed System device such as vendor, catalog listing, serial number, etc.

Get Privilege

Returns the privilege level for the Smart Distributed System bus.

Get Returncode Description

Returns a text description of the return code for a specified API service.

Get Supported Bus Type

Returns the numeric ID and text description of all bus types supported by the Smart Distributed System DLL.

Get Supported Channel

Returns the numeric ID and text description of all channels supported by the specified Smart Distributed System hardware interface.

Get Supported Privilege

Returns the numeric ID and text description of all privileges supported by the Smart Distributed System DLL.

Get Supported Speed

Returns the numeric ID and text description of all speeds supported by the specified Smart Distributed System hardware interface.

Init

Initializes a Smart Distributed System bus. The bus will be enrolled.

Is Analog

Indicates whether the specified input or output value is analog or digital. More detailed I/O information may be obtained by decoding the value returned by the Get I/O Descriptor service.

Multi-Errors

Obtains all system and device level diagnostics at once. The behavior of the system and device diagnostic indications are the same for each as if they were read individually.

Multi-Read Input

Reads all Smart Distributed System input data at once.

Multi-Read Output

Reads all Smart Distributed System output data at once.

Multi-Write Output

Writes all Smart Distributed System output data at once.

Read Input

Reads the input for a Smart Distributed System device.

Set Attr

Writes data to a specified attribute of a Smart Distributed System device.

System Info

Returns information about the Smart Distributed System bus, such as the hardware type and location, whether or not the bus is running, the version number of the interface software, etc.

Watchdog Control

Configures the system watchdog timer. This watchdog trips if the system controller does not refresh it at the configured rate.

Watchdog Status

Indicates the status of the system watchdog timer.

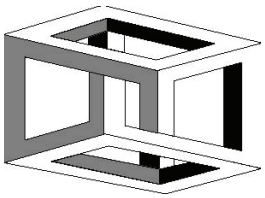
Watchdog Sustain

Sustains the system watchdog timer (keeps it from tripping). Once configured, the system watchdog must be sustained faster than the configured rate to keep it from tripping.

Write Output

Writes the output value to a Smart Distributed System device.

Chapter 4



Configuration File

You may, at your option, specify a configuration file describing the low-level details of your system. **The use of a configuration file is highly recommended**; however, you can elect to omit a system configuration file. The purpose of the system configuration file is to establish and verify each attribute in every device on all of the buses in the system. By specifying a Configuration File, you will be assured that each and every device is present and properly configured prior to system startup. Any text editor or word processor (text mode) may be used to create a configuration file, or software is available from third-parties to automatically create a configuration file from a properly configured network.

Start Up the PC Interface Card

The Smart Distributed System PC Interface is engineered to automatically begin running when power is applied to the PC. No other action is needed to start the PC Interface Card.

System Configuration File Syntax

There are very few rules to remember when setting up a system Configuration File.

Case

Case is not significant. Capitalize the commands and comments any way that looks good and makes sense.

Special Character

The asterisk (*) is the only significant special character. Everything on the line after an asterisk is considered a comment and is not executed. The following special characters are not significant and can be used to organize your file.

Name	Character
Parenthesis	()
Square Brackets	[]
Tab	
Space	
Comma	,
Colon	:
Equal Sign	=

Organization

The Configuration File has a relatively simple hierarchical organization (See Figure 4-1 Configuration File Organization Example). The commands for each board (CHANNEL, BAUD, etc.) must be grouped under the BOARD command. The ADDRESS commands must be grouped under the appropriate CHANNEL command and the INPUT, OUTPUT, and ATTR commands must be grouped under the appropriate ADDRESS command.

```
*** BNF Syntax for the SDS CFG File ***
BOARD <ISA | PCI> <board_address>
    Channel < 1 | 2 > Baud < 125 | 250 | 500 | 1000 >
    Address < device_address >

                Replacement_Address < device_address >
                Hierarchy <hierarchy_number>
                [ Input | Output ] <io_bits>
                Attr < attribute_number > [ Type ] [ Value ]
<board_address> if ISA = <D000> <D400> <D800> <DC00> <E000>
<E400>
<board_address> if PCI = <numeric_value>
< device_address > = <1 | 2 | ... 125 | 126 >
<hierarchy_number> = <n>.<n>.<n>.<n>. < etc >
<io_bits> = <numeric_value> 1 ... 32
< attribute_number > = < 0 | 1 ... 254 | 255 >
[ Type Array n ] = <Byte | Word | Long |_Float>
[ Value ] = <0 | 1… | 4294967295 | “char”>
```

Figure 4-1 Configuration File Organization Example

System Configuration Commands

Board

The BOARD command allows you to specify the address of the PC Interface Card which has been selected via the iTools program. (You must specify the Card ID number you selected for each card with the iTools program).

Syntax: BOARD <PCI> <*nnnn*>

Baud

The BAUD command specifies the bus speed of the buses connected to B1 and B2 of the PC Interface Card respectively. The BAUD option is subordinate to the CHANNEL option.

Syntax: BAUD <*125/250/500/1000*>

NOTE: Different bus topology rules are applicable to each selectable baud rate.

Channel 1 and 2

The CHANNEL 1 and CHANNEL 2 commands specify that the network device data in the section following pertains to connector B1 or B2, respectively. The CHANNEL option is subordinate to the BOARD option.

Syntax: CHANNEL 1

Address

The ADDRESS option specifies the device address number. The data following this option pertains to the selected device number. Each ADDRESS specified must be followed by a corresponding INPUT or OUTPUT command (see below). The ADDRESS option is subordinate to the CHANNEL option.

Syntax: ADDRESS <*nnn*>

Replacement_Address

The REPLACEMENT_ADDRESS command specifies the address which the Smart Distributed System PC Interface will interrogate in an effort to locate a device which is missing from the system. The user must also specify the HIERARCHY command whenever the REPLACEMENT_ADDRESS command is specified. When the Smart Distributed System PC Interface Card discovers a device is missing from the system, the card will examine the specified replacement address

and object model to determine if the missing device is located at the replacement address. If this situation is **true** and **unambiguous**, the Smart Distributed System PC Interface will move the device from the replacement address to the device address. The REPLACEMENT_ADDRESS command is subordinate to the ADDRESS option.

Syntax: REPLACEMENT_ADDRESS <device_address>

Hierarchy

The HIERARCHY command specifies the object model hierarchy number of the device used at the specified address. The system uses the option only with the REPLACEMENT_ADDRESS command. Hierarchy information is available from the device manufacturer or by reading Attribute 2 within the device. When the following conditions are met, the Smart Distributed System PC Interface will automatically readdress a device from the specified replacement address to the specified device address:

1. The device address is missing from the bus.
2. One device is present on the bus at the replacement address.
3. The device at the replacement address has the specified object model number.

Syntax: Object Model Number <nn>.<nn>.<nn>.<nn>.<etc>

Input

The INPUT command specifies that the device is an input device. The user then specifies the number of bits of input which are contained in the device. Check the device documentation to determine how many bits are contained in the device. Most digital sensors (On or off) contain only one bit. Analog inputs or some specialty devices may contain more. The INPUT command is subordinate to the ADDRESS option.

Syntax: Input <io_bits>

Output

The OUTPUT command specifies that the device on the bus is an output device. The user then specifies the number of bits of output which are contained in the device. Check the device documentation to determine how many bits are contained in the device. Most digital actuators (On or off) contain only one bit. Analog outputs or some specialty devices may contain more. The OUTPUT command is subordinate to the ADDRESS option.

Syntax: Output <io_bits>

Attr

The ATTR option allows you to verify or set specific device attributes on power-up or reset. The ATTR option is subordinate to the ADDRESS option.

Syntax: ATTR <nn> <type> <value>

<nn> The number of the attribute to be checked and updated if necessary.
<type> The type of the attribute in the device. The following types of attributes are supported by devices:

BYTE - A single 8 bit character attribute.
WORD - A numeric attribute which is 2 bytes in length. This type of attribute may take values from 0 to 65,535.
LONG - A numeric attribute which is 4 bytes in length. This type of attribute may take values from 0 to 4,294,967,295.
FLOAT – A single-precision floating point numeric value. The ARRAY modifier can be used with all three types of descriptors to describe an array of the attribute types.

Syntax: ATTR <*nn*> <*type (ARRAY <nn>)*> <*value*$_1$> <*value*$_2$>…<*value*$_{nn}$>

ARRAY <nn> The number of values in the array.

<*value*> The desired value of the attribute in the device. Character strings should be between quote marks and are connected to individual byte values.

Example Configuration Files

Example #1

- One PC Interface Card is installed in the system, configured with iTools, if needed.
- Bus #1 is started at a bus speed of 125K bits per second.
- Input devices 1, 2, and 3 are connected to the Bus. The Normally Open or Normally Closed attribute in the device is set to the appropriate value.
- Output devices 17 and 18 are connected to the Bus.
- One input device uses the Automatic Replacement address feature of the system.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
* System Configuration File
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * Board #1 * * * * * * *

Board: 6      * Board ID 6 configured with iTools


* * * * * * * * * * * * * Channel #1 * * * * * * * * * * * * * * *

Channel 1

Baud1 = 125   * Select Speed of Bus #1

Address: 1 Input: 1 Attr 60 (Byte) = 0 * NO/NC Option = NO
Replacement_Address = 126 Hierarchy = 1.1.1.1.1


Address: 2 Input: 1 Attr 60 (Byte) = 0 * NO/NC Option = NO


Address: 3 Input: 1 Attr 135 (Byte Array 6) = “ABCDEF” *
NO/NC Option = NC

Address: 17 Output: 1 Attr 60 (Byte) = 0 * NO/NC Option =
NO

Address: 18 Output: 1 Attr 60 (Byte) = 0 * NO/NC Option =
NO
```

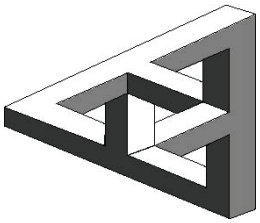
Example #2

- Two PC Interface Cards are installed on the system; comfigured with iTools.
- All Channels are started at a bus speed of 125K bits per second. For this example 4 Buses will be enabled.
- For the sake of this illustration, one input device and one output device per bus will be installed.
- One input device uses the automatic replacement address feature of the system.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
* System Configuration File
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * Board #1 * * * * * * *

Board: 6      * Board ID 6 configured with iTools

* * * * * * * * Board # 1 - Channel #1 * * * * * * * * * * * * * * * *

Channel 1

Baud = 125           * Select Speed of Bus #1

Address: 6           Input: 1      * MHP Device
Replacement_Address: 126   Hierarchy: 1.1.1.1.1
       Attr 6 (Byte) = 1           * Un/Solicited
       Attr 58 (Byte) = 5          * Marginal Gain Count
       Attr 60 (Byte) = 0          * NO/NC Option = NC
       Attr 62 (Word) = 5000       * On Delay = 5000 ms
       Attr 63 (Word) = 0          * Off Delay = 0 ms
Address: 37          Output: 1
       Attr 6 (Byte) = 1           * Un/Solicited
       Attr 60 (Byte) = 0          * NO/NC Option = NC


* * * * * * * * * Board # 1 - Channel #2 * * * * * * * * * * * * * * * *

Channel 2

Baud = 125           * Select Speed of Bus #2

Address: 36          Input: 1      * MHP Device
       Attr 6 (Byte) = 1           * Un/Solicited
       Attr 58 (Byte) = 5          * Marginal Gain Count
       Attr 60 (Byte) = 0          * NO/NC Option = NC
       Attr 62 (Word) = 5000       * On Delay = 5000 ms
       Attr 63 (Word) = 0          * Off Delay = 0 ms
Address: 90          Output: 1
       Attr 6 (Byte) = 1           * Un/Solicited
       Attr 60 (Byte) = 0          * NO/NC Option = NC
```

```
* * * * * * Board #2 * * * * * *

Board: 7            * Board ID 7 configured with iTools

* * * * * * * * * *Board #2 - Channel #1 * * * * * * * * * * * * * * *

Channel 1

Baud = 125          * Select Speed of Bus #1

Address: 85         Input: 1      * MHP Device
       Attr 6 (Byte) = 1          * Un/Solicited
       Attr 58 (Byte) = 5         * Marginal Gain Count
       Attr 60 (Byte) = 0         * NO/NC Option = NC
       Attr 62 (Word) = 5000      * On Delay = 5000 ms
       Attr 63 (Word) = 0         * Off Delay = 0 ms
Address: 107        Output: 1
       Attr 6 (Byte) = 1          * Un/Solicited
       Attr 60 (Byte) = 0         * NO/NC Option = NC


* * * * * * * * * Board #2 - Channel #2 * * * * * * * * * * * * * * *

Channel 2

Baud = 125          * Select Speed of Bus #2

Address: 22         Input: 1      * MHP Device
       Attr 6 (Byte) = 1          * Un/Solicited
       Attr 58 (Byte) = 5         * Marginal Gain Count
       Attr 60 (Byte) = 0         * NO/NC Option = NC
       Attr 62 (Word) = 5000      * On Delay = 5000 ms
       Attr 63 (Word) = 0         * Off Delay = 0 ms
Address: 23         Output: 1
       Attr 6 (Byte) = 1          * Un/Solicited
       Attr 60 (Byte) = 0         * NO/NC Option = NC
```

Chapter 5



Log File and Error Messages

The SDS.LOG file keeps an on-going record of the power-up condition of the Smart Distributed System bus. This provides an excellent source of information for:

- Application development
- System maintenance
- System operational documentation

The Anatomy of the Log File

The Log File contains several step-by-step records of a system's initialization. The Log File size is restricted to approximately 2000 lines, so the file will not grow to a large, unmanageable size. The oldest bus start up data is automatically removed from the file in favor of the new data. The name of the Log File is SDS.LOG and cannot be changed.

Example Log File

The following example of the SDS.LOG file contains no errors. See the following section on error messages for a description of error messages that may appear in the Log File.

Parameter: SDS.cfg Date:
04/18/95 Time: 09:07:34

Initialize the PC Interface Board: 6 Bus #1: 125 Bus #2: 125
System Initialization Complete -- No Errors.

---------- End Of File ----------

Information Messages

This section details the informational messages that can appear on the screen or in the Log File.

Date: *nn/nn/nn* Time: *nn:nn:nn*

Time and date stamp the start of the program for diagnostic purposes.

File: *name* Line: *number*

Information message telling the user the File name and Line number that contained an error. Used for correcting errors in syntax in the Configuration File.

SDS System Initialize In Progress (Version:*nn*)

Start up message to inform the user which program is running and the version number of the program. *nn* is the version number.

Initialize the PC Interface Board: *nnnn Bus#n: nnn Bus#n: nnn*

Information message informs the user of the buses initialized on the specified PC Interface card. *nnnn* is the card's ID number, *n* is the bus number, *nnn* is the baud rate.

Parameter: *nnn*

Message displaying the parameter passed to the program when it is started. *nnn* is the parameter value.

System Initialization Complete -- No Errors.

The Smart Distributed System was successfully started. No errors were detected when the system was started.

Error Messages

There are three types of errors you can receive from the SDSBEGIN.BAT program: system initialization errors, device errors, or configuration file errors. These error messages may appear on the screen, in the log file, or both.

Initialization Error Message

The initialization error message indicates that there has been an error in initializing the Smart Distributed System bus.

Errors Found in Smart Distributed System Initialization.

The Smart Distributed System was NOT successfully started. Other errors were detected in system start up. See the other error messages for more details.

Initialize Error: *nnnn*

In the log file or on screen the initialize error code will be represented by one of the following messages. *nnnn* is the error code for programmer's use.

<<<Unknown Channel Error Bit>>>

This means that the bit is undefined and should always have a value of 0.

Initialization Error Messages – Global Diagnostic Register

Bit: 0 - Power Error detected by the Host Interface
Bit: 1 - Embedded Software Receive FIFO Overflow
Bit: 2 - Embedded Software Transmit FIFO Overflow
Bit: 3 - Embedded Software I.M.B. FIFO Overflow
Bit: 4 - Device Errors Present on the Channel
Bit: 5 - CAN Chip Report -- Overrun Error bit asserted
Bit: 6 - CAN Chip Report -- Error Status bit asserted
Bit: 7 - CAN Chip Report -- Bus Status bit asserted
Bit: 8 - Embedded Software Health Check Error
Bit: 9 - Host Watchdog Tripped
Bit: 10 - <<< Unknown Channel Error Bit >>>
Bit: 11 - <<< Unknown Channel Error Bit >>>
Bit: 12 - <<< Unknown Channel Error Bit >>>
Bit: 13 - <<< Unknown Channel Error Bit >>>
Bit: 14 - <<< Unknown Channel Error Bit >>>
Bit: 15 - <<< Unknown Channel Error Bit >>>

With one exception, all of the messages indicate a problem with bus operations. To troubleshoot these problems you should:

- Check the bus power to insure it is correct
- Confirm that the system cabling is correct
- Refer to the following Initialize Error Message Descriptions

If these steps don't resolve the problem you should contact Holjeron Customer Service at 1-800-691-8302 for assistance. See Sales and Service on page 2.

The one exception is Device Errors Present on the Channel. See the next section for details on device errors.

Initialize Error Message Descriptions

Bit: 0 - Power Error detected by the Host Interface

This bit reflects the state of the power on the bus.

Value 0 = Bus power OK Value 1 = Bus power error – NO POWER on the BUS.

Note: This status indicator is reserved for later hardware compatibility on the Holjeron PC Interface Card, and cards from various other vendors. When implemented, the value will always be coded as a 0.

Bit: 1 - Embedded Software Receive FIFO Overflow

Serious internal error on the Smart Distributed System PC Interface. This condition occurs rarely, if at all.

This bit is set if the Receive FIFO (First In - First Out buffer storage system) is full and can accept no more messages. All input bus traffic is processed on the control PC Interface by the ISR (Interrupt Service Routine). This software module places the raw CAN data into the receive FIFO for processing by the Input Task. This error occurs when the Input Task is unable to process the data from the ISR in a timely fashion. At some time the data passed between the ISR and the Input Task accumulates to the point where the FIFO buffer is entirely full.

Solution:
- Select a slower baud rate to reduce the bus traffic the PC Interface must process.
- Reduce the bus traffic by other means including turning off cyclical timers in devices, etc.

Bit: 2 - Embedded Software Transmit FIFO Overflow

Serious internal error on the PC Interface. This condition occurs rarely, if at all.

This bit is set if the Transmit FIFO (First In - First Out buffer storage system) is full and can accept no more messages. All output bus traffic is placed in the output FIFO to be sent to devices via the CAN controller chip. If the CAN controller chip cannot process the data as quickly as the data is placed in the Output FIFO, the number of messages may accumulate to the point where the FIFO buffer is entirely full. At that point, the Transmit FIFO Overflow error indication will be posted.

Solution:

- This error is usually caused by the inability of the host to send any data on the bus due to serious bus error conditions. Ensure the bus is functioning properly by checking the power supply and the connections to devices. This also includes main trunk cable, branch cables and tees. The existence of other error bits usually accompany this type of failure.
- Reduce the output load on the system by increasing the scan time to the point where the PC Interface card is NOT flooded with output requests.

Bit: 3 - Embedded Software I.M.B. FIFO Overflow

Serious internal error on the PC Interface. This condition occurs rarely, if at all.

All bus traffic which is NOT understood by the PC Interface is placed into the IMB Buffer (Incoming Message Buffer) This buffer may be read by the host driver software to determine what bus traffic exists on the bus which the interface card is unable to decode. This buffer will overflow when enough bus traffic is present which the interface card is unable to understand.

Solution:

- Error is due to the fact that a device is sending unknown bus traffic. Replacement of this device will correct this error condition.

Bit: 4 - Device Errors Present on the Channel

This bit indicates that at least one device on the PC Interface channel has reported a device error. Please see the report of device errors for more details.

Bit: 5 - CAN Chip Report -- Overrun Error bit asserted

Serious internal error on the PC Interface. This condition occurs rarely, if at all.

This bit indicates that a low level CAN error has occurred in the CAN Controller which is used on the PC Interface. This internal error is due to the fact that the embedded process is unable to retrieve the CAN message from the CAN Controller chip before another message was present on the bus.

Solution

- Reduce the baud rate of the bus to allow the embedded system more time to process CAN messages present on the bus.

Bit: 6 - CAN Chip Report -- Error Status bit asserted

This error condition occurs rarely. It is indicative of a bus problem affecting the PCI Interface Card's ability to communicate. The cause of the problem should be corrected.

This bit indicates that a low level CAN error has occurred in the CAN Controller which is used on the PC Interface. This internal error is due to excessive error traffic on the bus causing an error counter on the CAN Controller to be incremented to the point where the CAN Controller has declared a Warning. This error indicator indicates a heavily disturbed bus.

Solution:

- Ensure the bus is functioning properly by checking the power supply and the connections to devices. This includes the tees, main trunk cable and branch cables.
- Consult "Installation and Troubleshooting Guide" (Honeywell PK-80062) for detailed troubleshooting procedures.

Bit: 7 - CAN Chip Report -- Bus Status bit asserted

This condition occurs rarely. It is indicative of a bus problem affecting the PCI Interface Card's ability to communicate. The cause of the problem should be corrected.

This bit indicates that a low level CAN error has occurred in the CAN Controller which is used on the PC Interface. This internal error occurs when the error traffic on the bus has caused an error counter on the CAN Controller to be incremented to the point where the CAN Controller has declared it is no longer involved in Bus communications. This bit indicates a heavily disturbed bus and control will be disrupted until the underlying problem is corrected.

Solution:

- Ensure the bus is functioning properly by checking the power supply and the connections to devices. This includes the tees, main trunk cable and branch cables.
- Consult "Installation and Troubleshooting Guide" (Honeywell PK-80062) for detailed troubleshooting procedures.

Bit: 8 - Embedded Software Health Check Error

Serious internal error on the PC Interface. This condition occurs rarely, if at all.

The embedded software contains a Health Check task which verifies that the embedded software system is functioning properly. This check occurs every 500 ms. This error is an indication that the PC Interface has failed internally.

Solution:

- Shut down the application, Reset the card using the iTools program, and restart the application
- If the situation continues, replace the card.

Bit: 9 - Host Watchdog Tripped

The PC Interface card supports a feature which allows the Host driver to enable a watchdog system such that the PC Interface monitors the Host Control software to ensure it is running properly. Should the host ever fail to update the output watchdog feature, the PC Interface will infer that the host control program has failed. At that time, the PC Interface will either clear all outputs or maintain their state, depending on pre-selected options specified when the watchdog is first enabled. This bit indicates that the Host driver has failed to update the Watchdog output on the PC Interface card at the proper time.

Solution:
- Increase the watchdog time specified in the driver when the Host to Smart Distributed System PC Interface watchdog is enabled.
- Contact the supplier of the driver software for support to correct this error.

Bit: 10 - <<< Unknown Channel Error Bit >>>
Bit: 11 - <<< Unknown Channel Error Bit >>>
Bit: 12 - <<< Unknown Channel Error Bit >>>
Bit: 13 - <<< Unknown Channel Error Bit >>>
Bit: 14 - <<< Unknown Channel Error Bit >>>
Bit: 15 - <<< Unknown Channel Error Bit >>>

Undefined bit in the Error register. These bits should NEVER be set.

Device Error Messages

Device errors are generated by individual devices. They can notify the PC system of device failures or they may be a device diagnostic message.

Device Error for Address: *nn* Error: *nnn*

The Device Error indicates which device has had an error. Address : *nn* is the device address. Error: *nnn* is the error code from the device with the error. More information detailing the error will be displayed following the device error message.

The device error message may indicate a device failure or a diagnostic message from a device. There are a total of 32 device diagnostic bits, ten of which have a common definition for all Smart Distributed System devices.

The following list details the possible error codes. For definition of device-specific errors please refer to the manufacturer's documentation for the device.

Device Error Messages

			Set by Host	Device Specific
Byte	Byte Bit	Error Bit		
Byte 0	0	0	RomChecksum	This bit indicates the device has detected a program checksum error.
	1	1	Watchdog	This bit indicates the device watchdog has detected a failure.
	2	2	BusOff	This bit is set when a device is present and has reported a bus off error.
	3	3	FatalError	This bit is set when a fatal device error has occurred.
	4	4	MissingNode	This bit is set when a device responded to the configuration sequence but subsequently failed to communicate, even if only temporarily.
	5	5	DuplicateNode	This bit is set when multiple devices with the same address have been detected.
	6	6	Unused	
	7	7	EEPROM	This bit is set if the device has detected a non volatile memory failure.
Byte 1	0 - 7			
Byte 2	0 - 7			
Byte 3				
	5	29	Configuration Error	If a CFX File exists This bit is set if there is there is any difference between the configuration file and the actual configuration. If a CFX file does not exist If an enrolled device drops off the bus, and later another device begins responding at the same address, the system checks if the new device's NDD is the same as the original device's NDD. If the new device's NDD is not the same as the original device's NDD, then this error bit will be set.
	6	30	Write or Re-Write Error	This bit is set if 1.) a device has failed to acknowledge Write commands from the Host after three attempts, or 2.) the output value of the device was not corrected after it was sent three times.
	7	31	Device Error Code	This bit is set when a device responds with an error code to any bus transmission.

Device Error Message Descriptions

Bit: 0 - Rom Checksum Error in Device

This bit indicates the device has detected a program memory checksum error. The device will undergo a self-test every time the PC Interface starts the bus. The program checksum is a check of all of the program memory in a device to ensure the device is properly functioning and programmed correctly.

Solution:
- Replace the device.

Bit: 1 - Device Watchdog Trip

Certain Output devices may implement the Device Watchdog timer function. If enabled, communication must occur within the time interval configured for each device in order to prevent the device watchdog from tripping. This bit indicates the device watchdog has detected a failure. The device supports a watchdog feature and the system has NOT met the requirements of the watchdog.

Solution:
- Examine the watchdog setting selected, and reconfigure the device so that the system will meet the requirements of the device watchdog. The exact details of the device watchdog should be contained in the documentation for the device. Please refer to those documents.

Bit: 2 - Device reports Bus Off Condition

This bit indicates the device has reported a bus off error. The meaning of the Bus Off Condition is that there is at least a temporary loss of communication between the device and the bus, which may disrupt system control.

Solution:
- Ensure the bus is functioning properly by checking the power supply and the connections to devices. This includes the tees, main trunk cable and branch cables. Consult "Installation and Troubleshooting Guide" for detailed troubleshooting procedures.

Bit: 3 - Fatal Device Error Detected

This bit indicates the device has detected a fatal device error.

Solution:
- Replace the defective device. The exact details of the fatal device error indications should be contained in the documentation for the device. Please refer to those documents.

Bit: 4 - Host Detected - Device Missing from the Channel

The PC Interface checks communications to each device on the bus at least once every 2.5 seconds. If the device does NOT communicate in that period of time, the host will generate a message to the device in an effort to ensure the device is still present and functioning properly on the bus. Should the device be removed from the bus, or fail to communicate after the initial communications check and 2 additional retries, this bit will be asserted by the host to indicate a missing device error.

Solution:

- Check the wiring between the PC Interface and the device. Either the bus signal or the power to the device may be incorrect.
- Replace the device if it isn't functioning properly.

Bit: 5 - Host Detected - Duplicate Device Address on the Channel

During enrollment and bus startup, the host PC Interface found more than one device at the specified address. This is an invalid situation and must be corrected to ensure proper operation of the system.

Solution:

- Find the device that is addressed improperly and set the address to a unique device address.

Bit: 6 - <<< Device Specific Bit >>>

Consult device manufacturer for details if an error occurs.

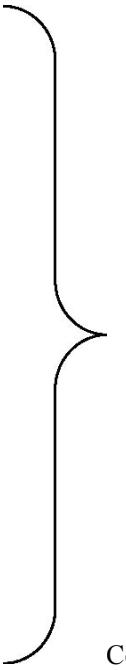
Bit: 7 - Device EEPROM Error Detected

This bit indicates the device has detected an error in the EEPROM system on the device.

Solution:

- Write data to the device at a slower rate to allow the device to update the internal EEPROM. This may be the problem if the user has elected to write data to device attributes in their application.
- Replace the device.

Bit: 8 - <<< Device Specific Bit >>>
Bit: 9 - <<< Device Specific Bit >>>
Bit: 10 - <<< Device Specific Bit >>>
Bit: 11 - <<< Device Specific Bit >>>
Bit: 12 - <<< Device Specific Bit >>>
Bit: 13 - <<< Device Specific Bit >>>
Bit: 14 - <<< Device Specific Bit >>>
Bit: 15 - <<< Device Specific Bit >>>
Bit: 16 - <<< Device Specific Bit >>>
Bit: 17 - <<< Device Specific Bit >>>
Bit: 18 - <<< Device Specific Bit >>>
Bit: 19 - <<< Device Specific Bit >>>
Bit: 20 - <<< Device Specific Bit >>>
Bit: 21 - <<< Device Specific Bit >>>
Bit: 22 - <<< Device Specific Bit >>>
Bit: 23 - <<< Device Specific Bit >>>
Bit: 24 - <<< Device Specific Bit >>>
Bit: 25 - <<< Device Specific Bit >>>
Bit: 26 - <<< Device Specific Bit >>>
Bit: 27 - <<< Device Specific Bit >>>
Bit: 28 - <<< Device Specific Bit >>>



for details

Consult device manufacturer

Bit: 29 - Host Detected - Device Configuration differs from the CFG File

The host has detected that the configuration data contained in the SDS.CFG file differs from the current configuration in the device and that the device was unable to be updated to the configuration that was contained in the SDS.CFG file.

Solution:

- Examine the SDS.LOG file to determine the exact details of which attributes are in error.
- This error most commonly occurs when the host attempts to write to a device during a time when it is missing from the bus, or cannot communicate due to a bus communication problem. Correct any underlying communication problem(s) first.
- If this error persists, replace the device.

Bit: 30 - Host Detected - Error Writing output data to the device

The host has attempted to write output data to the device, and that request has been refused 3 times by the device. This indicates that the device is not functioning properly.

Solution:

- Replace the device.

Bit: 31 - Host Detected - Negative ACK received from device

The host has received a Negative Acknowledgment to a request from a device. The system is organized such that all requests from senders must receive an acknowledgment from the receiver. Should the receiver refuse to carry out the request because the request is invalid or the receiver is currently unable to perform the desired command, the receiver may elect to send a Negative Acknowledgment to the sender. In this case, this error bit usually indicates that the device is not able to properly carry out the request from the host.

Solution:

- Reset the device and retry the request.
- Replace the device.

Configuration Error Messages

The following list of error messages may be displayed on either the system screen or in the SDS.LOG file or both. They indicate problems or errors in the Configuration File.

Board: *nnn* Channel: *nn* Card ID: *nn*

Part of an error message used to tell the user of Card ID *nn*, Channel number *nn,* and the Device number *nn*.

Duplicate Configuration Files Specified

Too many Configuration File names were specified when the program was started. Only one Configuration File name is allowed when the program is started.

Error in Specified Card ID

The board parameter is set to an invalid Card ID.

Error Initializing Interface Board Channel *nn* Status: *nnnn*

An error has occurred when attempting to initialize the specified channel of the system. The status information displayed is the raw error code returned from the PC Interface. More detailed information describing the status follows this message.

Error in Initialize Write Status: *nn*

An error has occurred in writing the initial value *nn* to an output device on the bus.

Extra Device on the Bus

An extra device was found on the Bus which was not specified in the Configuration File.

Error Invalid Channel Number

An invalid channel number was specified in the Configuration File.

Error Processing Line: *<line from file>*

********** Processing Aborted **********

Displays the line of the Configuration File which is incorrect.

Error Reading System Bus errors for Channel *nn* Status: *nnn*

The global error location which contains errors for the entire Bus contained an error indicator. This message displays the raw data value. This message is then followed by an interpretation of each portion of the error message.

Incorrect Attribute Value Attr: *nnn*

The value of an attribute to be set in a device was incorrectly specified in the Configuration File.

Initialize Errors for Channel *nn*

Heading for the error messages describing errors that have occurred in initializing the specified channel. More information detailing the exact nature of the error follows this heading.

Invalid Automatic_attribute_update parameter.

The Automatic_attribute_update parameter is set to an invalid selection.

Invalid Device Address.

An invalid device address was specified in the Configuration File. Device addresses must be 1 to 126.

Invalid Option Specified: *nnnnn*

The specified option *nnnnn* is invalid.

Invalid Speed Selected -- MUST BE 125, 250, 500 or 1000

The setting of the BAUD parameter is invalid.

Missing Address Information.

The address parameter was not complete. The address parameter must contain the INPUT or OUTPUT parameter to be considered complete.

Missing Board Parameter

The board parameter is missing from the Configuration File.

Missing Channel Information

The channel parameter was not found in the Configuration File. This parameter must be specified before any channel data is specified.

Missing Device

A device which was specified in the Configuration File was not found on the bus.

No Configuration File Specified.

No Configuration File was specified when the program was started.

Unable to Open Configuration File: *nnn*

The Configuration file was NOT found. The configuration file name must be passed to the program when the program is started.

Summary Screen

A shorthand display has been developed to condense the report output to a simple summary screen. Here is an example:

```
Initialize the SDS PC Interface Board: 6 Bus #1: 125 Bus #2: 125

* *        * * *   Summary of Bus Startup   Channel: 1                 * * *
*                            Errors
Extra Input:             3   4    5   11   12   13   19   20   21   70   71
                        72   73   74   75   76   77   78

Extra Output:            6   7    8   14   15   16   22   23   24
```

Index

9-Pin Female D-sub Connector · 4, 6
9-pin Male D-sub Connector · 4

A

ADDRESS · 19
API Services · ii, 12
Application Layer · 8
asterisk · 17
ATTR · 21
Authorized Distributor · 2

B

B1 · 4, 6, 19
B2 · 4, 6, 19
BAUD1 · 19
BAUD2 · 19
BOARD · 19

cables · 6
Cables · 4
CAN channels · 4
Case · 17
Cfg Bld · 12
Cfg Dev Status · 12
Cfg Sys Status · 12
Cfg2Cfx · 12
Change Address · 13
channel · 4
CHANNEL 1 · 19
CHANNEL 2 · 19
Clear Device Error · 13
Configuration · 8
Configuration Commands · 19
Configuration Disk · 8
Configuration Error Messages · 35
Configuration File Syntax · 17
Configuration Files · 22
Connector · 4, 6
Connectors and Cabling · ii, 4
CPU · 4 Create Handle · 13

D

Destroy Handle · 13
Device Action · 13
Device Error Messages · 31
Device Present · 13
Device Prim Tag · 13
DLL · 11
Download APL · 13

E

Error Message Descriptions · 28, 32
Error Messages · 27

F

Finish · 13 Flash
EPROM · 4, 10
flash memory · 7, 8

G

Generate Unique Addresses · 13
Get Attr · 13
Get Available Interfaces · 13
Get Bus Error · 14
Get Bus Error Description · 14
Get Device Error · 14
Get Device Error Description · 14
Get I/O Descriptor · 14
Get I/O Type · 14
Get Info · 14
Get Privilege · 14
Get Returncode Description · 14
Get Supported Bus Type · 14
Get Supported Channel · 14
Get Supported Privilege · 15
Get Supported Speed · 15

H

Hardware Initialization · ii, 7
Hierarchy · 20
Host System Requirements · 10

I

Information Messages · 26
Init · 15
Initialization Error Message · 27
Initialize Error · 27
INPUT · 20
Internet · 2
Is Analog · 15
iTool1 · 7

M

Multi-Errors · 15
Multi-Read Input · 15
Multi-Read Output · 15
Multi-Write Output · 15

O

optical isolator · 4
Order Guide · 4
OUTPUT · 20

P

Parameter · 26
PC Card Layout · ii, 4, 5
PC Interface Card Specifications · 10
PC to PCI Card Interface · ii, 4
PCI Bus Requirements · 10
PCI expansion slot · 6
Pin Assignments · 5
Program · 8

R

Read Input · 15
Replacement_Address · 19

S

SDS.LOG · 25
SDSID · 7
SDSIF DLL · 11
Service · 2
Set Attr · 15
Software Installation · 8
Special Character · 17
SRAM · 4, 10
Start Up · 17
System Info · 15

T

Termination Cap · 5
Termination resistor · 5
Termination Resistors · ii, 5
troubleshooting tools · 8

V

Verify · 8

W

Warranty · 2
Watchdog Control · 15
Watchdog Status · 16
Watchdog Sustain · 16
Write Output · 16