ADAM-5000 Series

RS-485 Based Data Acquisition and Control System User’s Manual

Copyright Notice

Acknowledgment

ADAM is a trademark of Advantech Co., Ltd. IBM and PC are trademarks of International Business Machines Corporation.

CE Nonification

The ADAM-5000 series developed by Advantech Co., Ltd. has passed the CE test for environmental specifications. Test conditions for passing included the equipment being operated within an industrial enclosure, using shielded twisted-pair RS-485 cables and having SFC-6 sleeve core clamps added to the power cable and the RS-485 cable. In order to protect the ADAM-5000 system from being damaged by ESD (Electrostatic Discharge) and EMI leakage, we strongly recommend the use of CE-compliant industrial enclosure products, shielded twisted-pair RS-485 cables, and core clamps.

Edition 7.3
Aug. 2007

A Message to the Customer…..

Advantech Customer Services

Each and every Advantech product is built to the most exacting specifications to ensure reliable performance in the unusual and demanding conditions typical of industrial environments. Whether your new Advantech equipment is destined for the laboratory or the factory floor, you can be assured that your product will provide the reliability and ease of operation for which the name Advantech has come to be known.

Your satisfaction is our number one concern. Here is a guide to Advantech's customer services. To ensure you get the full benefit of our services, please follow the instructions below carefully.

Technical Support

We want you to get the maximum performance from your products. So if you run into technical difficulties, we are here to help. For most frequently asked questions you can easily find answers in your product documentation. These answers are normally a lot more detailed than the ones we can give over the phone.

So please consult this manual first. If you still can't find the answer, gather all the information or questions that apply to your problem and, with the product close at hand, call your dealer. Our dealers are well trained and ready to give you the support you need to get the most from your Advantech products. In fact, most problems reported are minor and are able to be easily solved over the phone. In addition, free technical support is available from Advantech engineers every business day. We are always ready to give advice on application requirements or specific information on the installation and operation of any of our products.

Product Warranty

Advantech warrants to you, the original purchaser, that each of its products will be free from defects in materials and workmanship for one year from the date of purchase.

This warranty does not apply to any products which have been repaired or altered by other than repair personnel authorized by Advantech, or which have been subject to misuse, abuse, accident or improper installation. Advantech assumes no liability as a consequence of such events under the terms of this Warranty. Because of Advantech’s high quality-control standards and rigorous testing, most of our customers never need to use our repair service. If an Advantech product ever does prove defective, it will be repaired or replaced at no charge during the warranty period. For out-of-warranty repairs, you will be billed according to the cost of replacement materials, service time and freight. Please consult your dealer for more details.

If you think you have a defective product, follow these steps:

1. Collect all the information about the problem encountered (e.g. type of PC, CPU speed, Advantech products used, other hardware and software used etc.). Note anything abnormal and list any on-screen messages you get when the problem occurs.

2. Call your dealer and describe the problem. Please have your manual, product, and any helpful information readily available.

3. If your product is diagnosed as defective, you have to request an RAM number. When requesting an RMA (Return Material Authorization) number, please access ADVANTECH's RMA website: http://www.advantech.com.tw/rma. If the web sever is shut down, please contact our office directly. You should fill in the "Problem Repair Form", describing in detail the application environment, configuration, and problems encountered. Note that error descriptions such as "does not work" and "failure" are so general that we are then required to apply our internal standard repair process.

4. Carefully pack the defective product, a completely filled-out Repair and Replacement Order Card and a photocopy of dated proof of purchase (such as your sales receipt) in a shippable container. A product returned without dated proof of purchase is not eligible for warranty service.

5. Write the RMA number visibly on the outside of the package and ship it prepaid to your dealer.

Contents

Chapter 1 Introduction 1-1

1.1 Overview...1-2

1.2 System Configuration..1-3

1.3 A Few Steps to a Successful System........................... 1-4

Chapter 2 Installation Guideline.................................2-1

2.1 General 2-2

2.2 Module Installation 2-6

2.3 I/O Slots and I/O Channel Numbering 2-6

2.4 Mounting.. 2-7

2.5 Wiring and Connections 2-9

Chapter 3 ADAM-5000 System…..…………………….... 3-1

3.1 Overview 3-2

3.2 Major Features of the ADAM-5000 System 3-2

3.3 System Setup 3-6

3.4 Technical Specifications of the ADAM-5000 ……..... 3-7

Chapter 4 I/O Modules………..…………………………… 4-1

Chapter 5 Software Utilities………..………………………. 5-1

5.1 ADAM-4000 and ADAM-5000 Windows Utility….. 5-2

5.1.1 Overview 5-3

5.1.2 Save Function 5-3

5.1.3 COM Port Settings 5-4

5.1.4 Search Connected modules 5-5

5.1.5 Terminal Emulation 5-7
5.1.6 Data Scope 5-8
5.1.7 Saving a Module's Configuration to File 5-10
5.1.8 Load Module's Configuration File 5-12
5.1.9 Module Configuration 5-14
5.1.10 Module Calibration 5-15
5.1.11 Data Input and Output 5-19
5.1.12 Alarm Settings 5-21
5.1.13 Download Procedure 5-22
5.2 DLL (Dynamic Link Library) Driver 5-25
5.3 DDE (Dynamic Data Exchange) Server 5-25

Chapter 6 Command Set……………………………………….6-1
6.1 Introduction 6-2
6.2 Syntax 6-2
6.3 CPU Command Set.. 6-4
6.4 ADAM-5013 RTD Input Command Set ……............... 6-19
6.5 Analog Input Command Set 6-37
6.6 ADAM-5017H Analog Input Command Set …......... 6-57
6.7 Analog Input Alarm Command Set 6-71
6.8 Analog Output Command Set 6-90
6.9 Digital Input/Output Command Set 6-107
6.10 ADAM-5080 Counter/Frequency Command Set...... 6-115

Chapter 7 Troubleshooting…………..……………………….7-1
7.1 Hardware Diagnosis.. 7-2
7.2 Software Diagnosis 7-2
7.3 System Indicators 7-3
7.4 Communication Problems 7-5
7.5 I/O Module Troubleshooting 7-6

Appendix A Quick Start Example.................................... A-1

A.1 System Requirements to Setup an ADAM-5000 System…. A-2

A.2 Basic Configuration Hook-up A-5

A.3 Baud Rate and Checksum A-8

A.4 A Distributed ADAM-5000 Network System Hook-up A-11

Appendix B Data Format and I/O Ranges........................ B-1

B.1 Analog Input Formats B-2

B.2 Analog Input Ranges - ADAM-5017, 5018 and 5018P...... B-4

B.3 Analog Input Ranges of ADAM-5017H and 5017UH.. B-7

B.4 Analog Output Formats B-8

B.5 Analog Output Ranges B-8

B.6 ADAM-5013 RTD Input Format and Ranges B-9

Appendix C RS-485 Network....................................... C-1

C.1 Basic Network Layout C-3

C.2 Line Termination C-6

C.3 RS-485 Data Flow Control C-9

Appendix D How To Use Checksum Features.................... D-1

D.1 Checksum Enable/Disable D-2

Appendix E ADAM-4000/5000 System Grounding Installation… E-1

E. 1 Power Supplies For relevant wiring issues, please refer to the following scheme:............................E-2

E.2 Grounding Installation E-2

E.3 External DI, DO, AI, AO Wiring Reference E-3

E.4 Requirements for RS-485 signal wires E-3

E.5 Grounding reference (Ground bar for the factory, environment should have a standard resistance below W)…………….E-5

E.6 Some Suggestions on Wiring Layout E-6

Appendix F Grounding Reference…………………………….…. F-1

F.1 Grounding F-3

F.2 Shielding F-9

F.3 Noise Reduction Techniques................................... F-14

F.4 Check Point List F-15

Appendix G Grounding Reference…………………………….…G-1

G.1 The ADAM-5000 series main system's DIP Switch Setting.. G-2

G.2 Modbus Commands for 5000 Series G-3

G.3 Modbus Address Mapping G-5

G.4 Address mapping of ADAM-5080 For ADAM-5000 ModBus.. G-8

Figures

Figure 1-1 ADAM-5000 Configurations…………………….………….1-3

Figure 2-1 ADAM-5000 Diagnostic indicators…………….………..2-3
Figure 2-2 ADAM-5000 Network address DIP switch.………….2-4
Figure 2-3 Module alignment and installation………………..…..2-6
Figure 2-4 ADAM-5000 Panel mounting.………………………..2-7
Figure 2-5 ADAM-5000 Rail mounting.……………….……..2-8
Figure 2-6 ADAM-5000E Rail mounting..………………………..2-9
Figure 2-7 ADAM-5000 Wiring and connections..……………….2-10
Figure 2-8 Built-in Communication Ports for Diagnostic………..2-13
Figure 2-9 Flexible Communication Port Function Connection….2-14

Figure 3-1 Function block diagram..…………………….…..3-8

Figure 5-1 Display the connected module……………...…..5-4
Figure 5-2 Save the information of connected modules to txt file....5-4
Figure 5-3 Setup options……………………………..…..5-4
Figure 5-4 Checksum function enabled…………..……..…..5-8
Figure 5-5 The connection for the Data Scope function……..…..5-8
Figure 5-6 Monitor the issing commands from PC#1…………....5-9
Figure 5-7 Zero Calibration……………………………...…..5-16
Figure 5-8 Execute Zero Calibration…………………..…..5-16
Figure 5-9 Span Calibration………………………..…..…..5-16

Figure 5-10 Execute Span Calibration…………………….…….5-17
Figure 5-11 CJC Calibration…………………………….…….5-17
Figure 5-12 Execute CJC Calibration………………….…….5-18
Figure 5-13 RTD Module Calibration…………………….…….5-18
Figure 5-14 Analog Output Calibration………………….…….5-18

Figure 6-1 Baud rate codes……………………….…………..…….6-6
Figure 6-2 Analog module error codes……………….…..…….6-18
Figure 6-3 Data format for 8-bit parameters……..……….…….6-38
Figure 6-2 Data format of 8-bit parameters……..……….…….6-92

Figure A-1 Power supply connections…………..……….…….A-4
Figure A-2 ADAM-5000 system hook-up and configuration..…….A-6
Figure A-3 Grounding the INIT* terminal……..……….…….A-10
Figure A-4 ADAM-5000 network system hook-up…..………..A-11

Figure C-1 Daisy chaining…………………..…………..…….C-3
Figure C-2 Star structure ……..……………………..…….C-4
Figure C-3 Random structure…………………….………..…….C-5
Figure C-4 ADAM-4000 and ADAM-5000 in a network..………..C-6
Figure C-5 Signal distortion……………………………..…….C-7
Figure C-6 Termination resistor location…………….…….C-8
Figure C-7 RS-485 data floe control with RTS…..……………...C-9

1

Introduction

Introduction

1.1 Overview

The ADAM-5000 series is a complete product line that provides a wide variety of features in a data acquisition and control application. It includes 4 I/O-slots ADAM-5000/485 and 8 I/O-slots ADAM-5000E. They are remotely controlled by the host computer through a set of commands and transmitted in a RS-485 network. The system kernel is small, but offers many good features to the users. The modulardesign also provides more flexibility in the system configuration. The following is a summary of the major ADAM-5000 system components.

ADAM-5000 System Kernel

The ADAM-5000/485 system kernel includes a CPU card, a power regulator, a 4-slot base, a built-in RS-232 communication port and one built-in RS-485 communication port. The ADAM-5000E system includes all of the above components, except it has an 8-slot base. Details of the system kernel features and more are covered in Chapter 3.

I/O Configuration

The ADAM-5000/485 CPU can support up to 64 I/O points with the 4-slot base currently available.The ADAM-5000E CPU can support up to 128 I/O points with the 8-slot base currently available. These points can be assigned as input or output points.

I/O Module

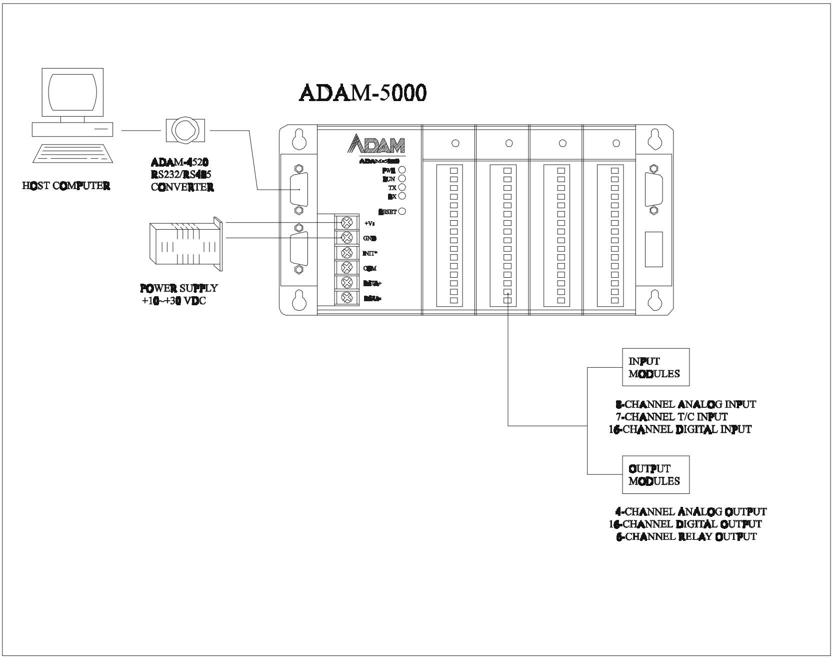
The ADAM-5000 series has a complete range of I/O modules for your applications. A full range of digital modules which support 10 to 30 V_{DC} and relay outputs are offered. The analog modules provide 16-bit resolution and programmable input and output signal ranges (including bipolar).

Software Utility

Based on the Modbus standard, the ADAM-5000 and ADAM-5000E firmware is a built-in Modbus/RTU . Therefore, Advantech provides the necessary OPC Server, and Windows Utility for users for client data for the ADAM-5000 and ADAM-5000E. Users can configure this DA&C system via Windows Utility; integrate with HMI software package via Modbus/RTU driver or Modbus/RTU OPC Server. and the DDE (Dynamic Data Exchange) server provides links to popular Windows packages such as Intouch, FIX DMACS, Advantech GeniDAQ, etc.

1.2 System Configuration

The following diagram shows the system configurations possible with the ADAM-5000.



Note: To avoid system over heating, only four ADAM-5024 are allowed to be installed on ADAM-5000E.

1.3 A Few Steps to a Successful System

Step 1:Review the installation Guideline

You should always make safety your first priority in any system application. Chapter 2 provides several guidelines that will help provide a safer, more reliable system.

Step 2: Understand the System Kernel

The system module is the heart of ADAM-5000 system. Make sure you take time to understand the various features and setup requirements.

Step 3: Understand the I/O System Configurations It is important to understand how your I/O modules can be configured. It is also important to understand how the system power budget is calculated. This can affect your I/O configuration.

Step 4: Understand the Utility Software

Before you begin to link your applications in your host computer with the ADAM-5000 systems, it is very helpful to understand how the DOS and Windows utility software helps you configure your ADAM-5000.

Step 5: Review the Programming Concepts

All control systems differ in some areas. The ADAM-5000 system allows you to develop your applications in DOS or Windows.It provides an ASCII command set, DLL (Dynamic Library Link) and DDE (Dynamic Data Exchange) server to you.

Step 6: Understand the Troubleshooting Procedures

Many things can be happened on the factory floor: switches fail, the power supply is incorrect, etc. In most cases, the majority of the troubleshooting time is spent trying to locate the problems.The ADAM-5000 system has some built-in features that help you quickly identify problems.

2

Installation Guideline

Installation Guideline

2.1 General

Environmental Specifications

The following table lists the environmental specifications that general ly apply to the ADAM-5000 system (System kernel and I/O modules).

Specification	Rating
Storage temperature	-13 to 185°F (-25 to 85°C)
Ambient operating temperature	14 to 158°F (-10 to 70°C)
Ambient humidity*	5 to 95%, non-condensing
Atmosphere	No corrosive gases

***** Equipment will operate below 30% humidity. However, static electricity problems occur much more frequently at lower humidity levels. Make sure you take adequate precautions before you touch the equipment. Consider using ground straps, antistatic floor cover- ings, etc. if you use the equipment in low humidity environments.

Power Requirements

Although the ADAM-5000 systems are designed for standard industrial unregulated 24 V_{DC} power supply, they accept any power unit that supplies within the range of +10 to +30V_{DC}. The power supply ripple must be limited to 100 mV peak-to-peak, and the immediate ripple voltage should be maintained between +10 and +30 V_{DC}.

Diagnostic Indicators

Diagnostic indicators are located on the front panel of the ADAM system. They show both normal operation and system status in your remote I/O system. The indicators are:

- System status (PWR, RUN)
- Communication status (TX, RX)
- I/O module status

A complete description of the diagnostic indicators and how to use them for troubleshooting is explained in Chapter 7.

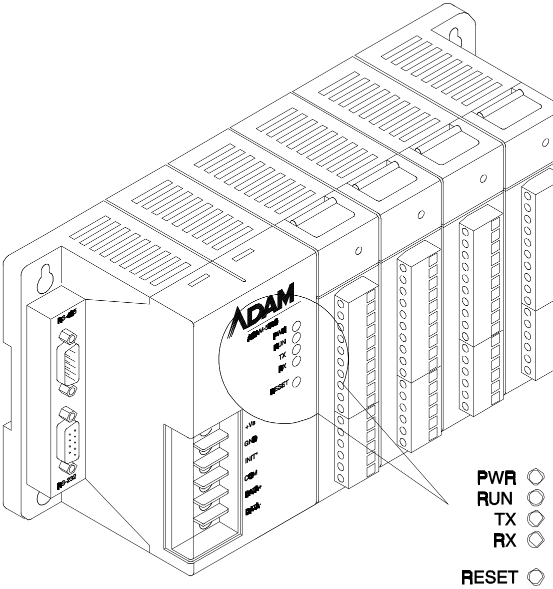


Figure 2-1 *ADAM-5000 Diagnostic indicators*

Setting the Network Address Switch

Set DIP switch 8 to OFF to install Avvantech Protocol , Set the network address using the 8-pin DIP switch. Valid settings range from 0 to 127 (00h to 7Fh) where ON in any of the 8 DIP switch positions equates to a binary 1, and OFF equates to a binary 0.

Note: 00h is special for initial setting. If 00h is setting by user, the baud rate setting will be fixed to 9600 bps. It is recommand to setting the range from 1 to 127 (01h to 7Fh).

Setting the Network Address Switch

Set DIP switch 8 to ON to install Modbus function , Set the Modbus address using the 8-pin DIP switch. Valid settings range from 0 to 127 (00h to 7Fh) where ON in any of the 8 DIP switch positions equates to a binary 1, and OFF equates to a binary 0.

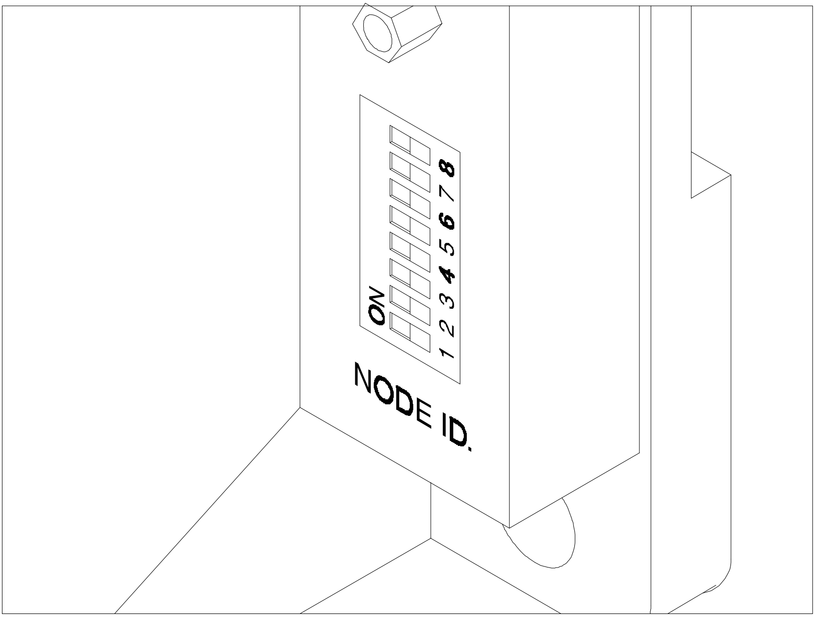
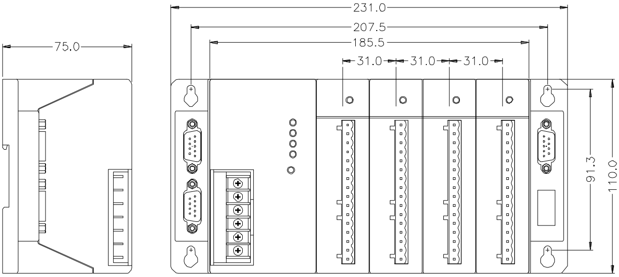


Figure 2-2 *ADAM-5000 Network address DIP switch*

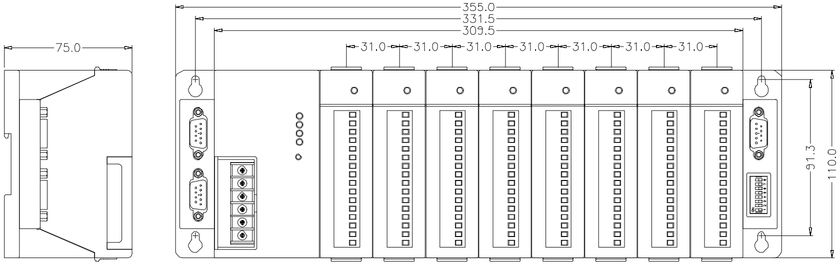
Dimensions and Weights(ADAM-5000)

The following diagrams show the dimensions of the system unit and an I/O unit of the ADAM-5000. All dimensions are in millimeters.



Dimensions and Weights(ADAM-5000E)

The following diagrams show the dimensions of the system unit and the I/O unit of the ADAM-5000E. All dimensions are in millimeters.



Module	5000/485	5000E	5013	5017	5017H	5018	5024	5050
Weight	470g	525g	45g	79g	45g	72g	75g	63g

Module	5051	5051D	5052	5056	5056D	5060	5068	5080
Weight	65g	45g	68g	68g	45g	85g	65g	52g

2.2 Module Installation

When inserting modules into the system, align the PC board of the module with the grooves on the top and bottom of the system. Push the module straight into the system until it is firmly seated in the backplane connector. Once the module is inserted into the system, push in the retaining clips (located at the top and bottom of the module) to firmly secure the module to the system.

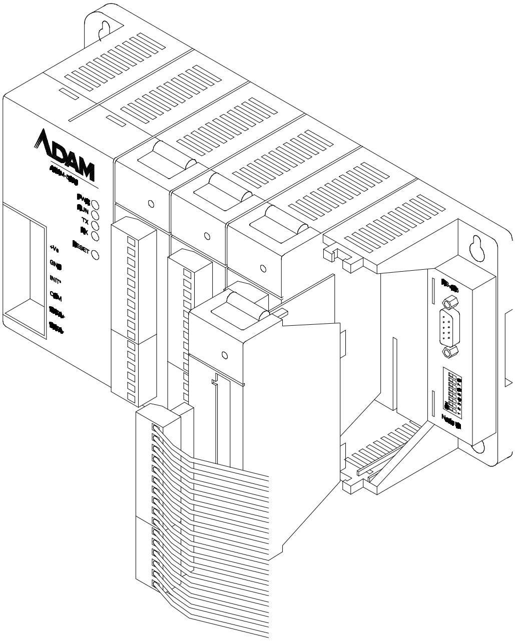


Figure 2-3 *Module alignment and installation*

2.3 I/O Slots and I/O channel Numbering

The ADAM-5000/485 system each provides 4 slots for use with I/O modules. The I/O slots are numbered 0 thru 3, and the channel numbering of any I/O module in any slot starts from 0. The ADAM-5000E system each provides 8 slots for use with I/O modules. The slots are numbered 0 thru 7. For example, ADAM-5017 is a 8-channel analog input module, its channel numbering is 0 thru 7.

2.4 Mounting

The ADAM-5000 system can be installed on a panel or DIN rail.

Panel Mounting

Mount the system on the panel horizontally to provide proper ventilation. You cannot mount the system vertically, upside down or on a flat horizontal surface. A standard #7 tating screw (4mm diameter) should be used.

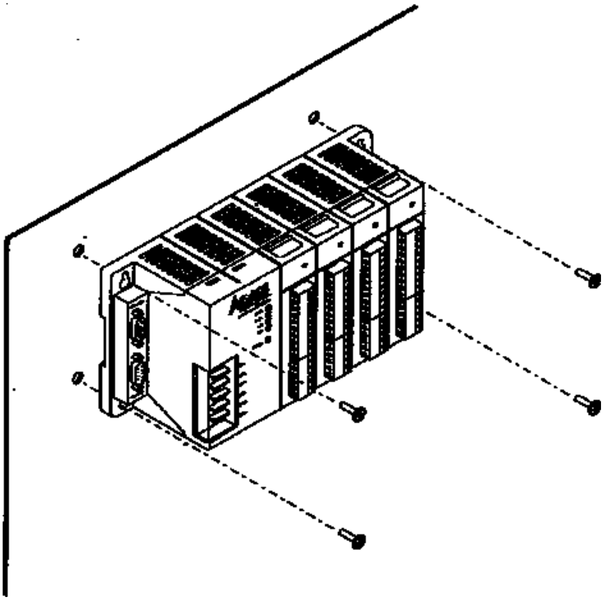


__Figure 2-4__ ADAM-5000 panel mounting

DIN Rail Mounting

The system can also be secured to the cabinet by using mounting rails. If you mount the system on a rail, you should also consider using end brackets on each end of the rail. The end brackets help keep the system from sliding horizontally along the rail. This helps minimize the possibility of accidentally pulling the wiring loose. If you examine the bottom of the system, you will notice two small retainingclips. To secure the system to a DIN rail, place the system onto the rail and gently push up on the retaining clips. The clips lock the system on the rail. To remove the system, pull down on the retaining clips, lift up on the base slightly, and pull it away from the rail.

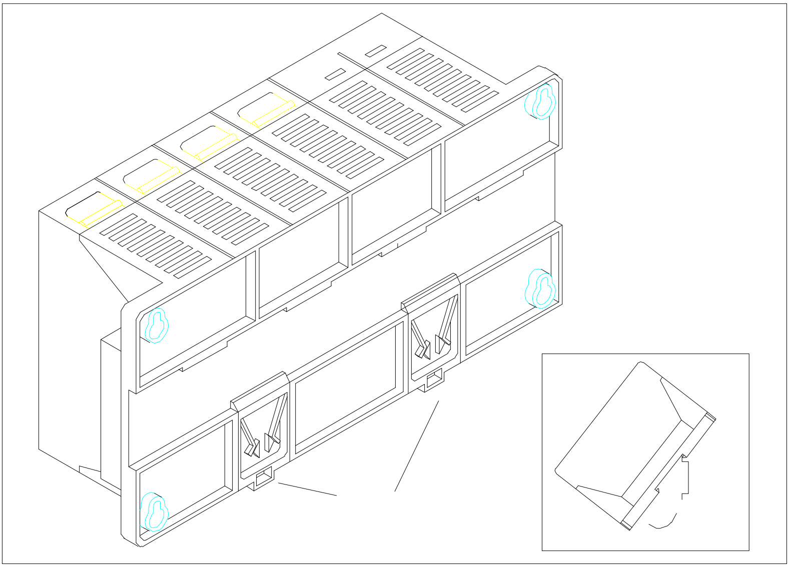


Figure 2-5 *ADAM-5000Rail mounting*

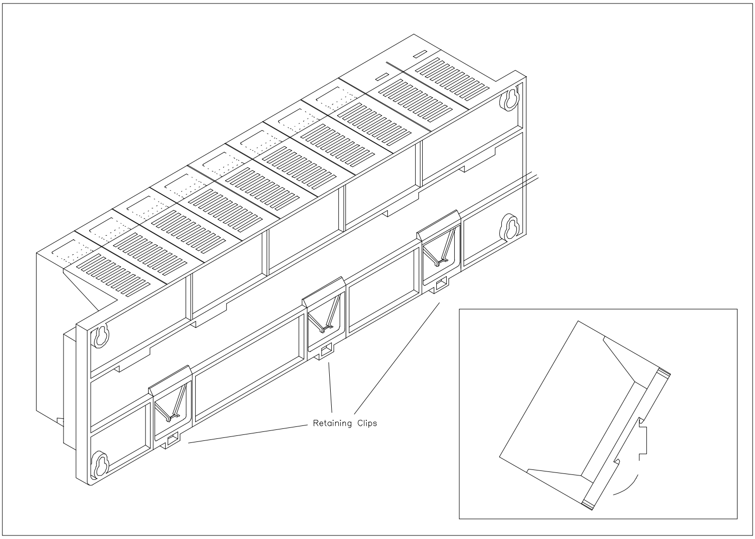


Figure 2-6 *ADAM-5000E Rail mountings*

2.5 Wiring and Connections

This section provides basic information on wiring the power supply and I/O units, and on connecting the network.

DC Power Supply Unit Wiring

Be sure that the DC power supply voltage remains within the allowed fluctuation range of between 10 to 30 V_{DC}. Terminals $+V_S$ and GND are for power supply wiring.

Note: The wire(s) used should be at least 2mm^2.

INIT* is used for changing baud rate and checksum. COM is provided as reference to the RS-485 ground signal. DATA+ and DATA- are provided for the RS-485 twisted pair connection.

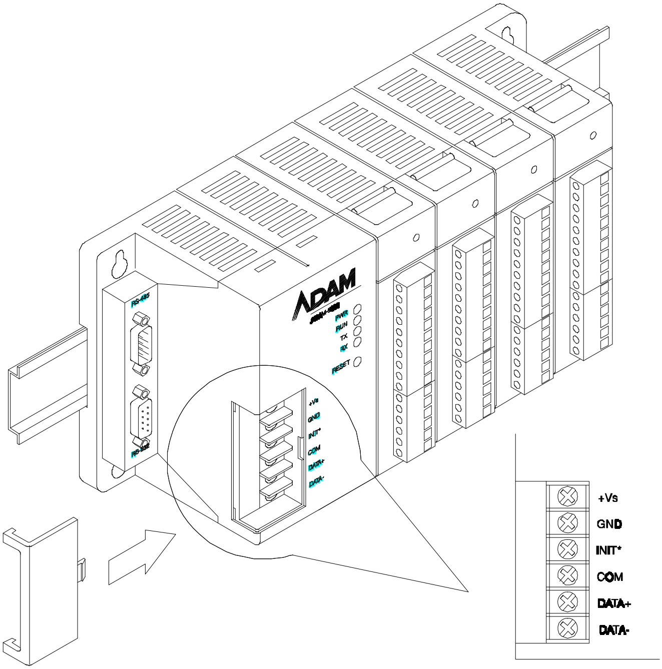


Figure 2-7 *ADAM-5000 Wiring Connections*

I/O Module Wiring

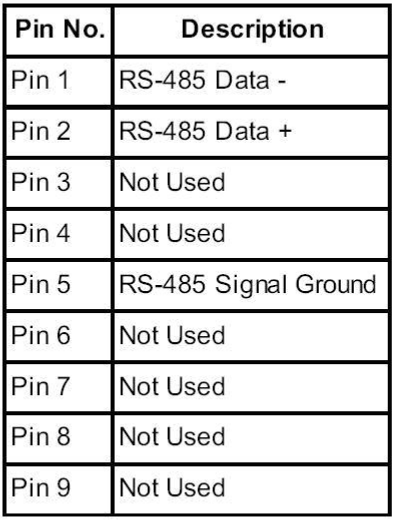
The system uses plug-in screw terminal blocks for the interface between I/O module and field devices. The following information must be considered when connecting electrical devices to I/O modules.

1. The terminal block accepts 0.5 mm^2 to 2.5 mm^2 wires
2. Always use a continuous length of wire, do not combine wires to attain needed length
3. Use the shortest possible wire length
4. Use the wire trays for routing where possible

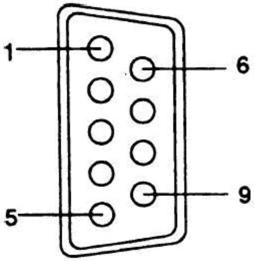
5. Avoid running wires near high energy wiring
6. Avoid running input wiring in close proximity to output wiring where possible
7. Avoid creating sharp bends in the wires

RS-485 Port Connection

There is a pair of DB9 ports in the ADAM-5000 system. The ports are designed to link the RS-485 through a cable to a network in a system. The pin assignment of the port is as follows:



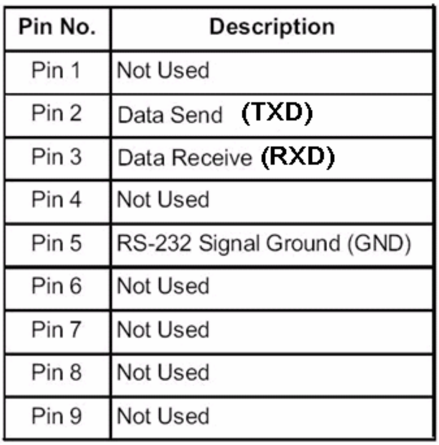
Pin No.	Description
Pin 1	RS-485 Data -
Pin 2	RS-485 Data +
Pin 3	Not Used
Pin 4	Not Used
Pin 5	RS-485 Signal Ground
Pin 6	Not Used
Pin 7	Not Used
Pin 8	Not Used
Pin 9	Not Used



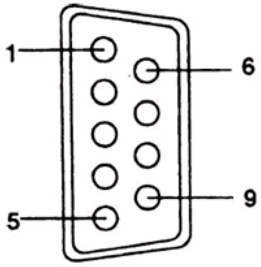
Note: The wiring of the RS-485 should be through a **twisted** pair. To reduce electrical noise, it should be twisted as tightly as possible

RS-232 Port Connection

The RS-232 port is designed for field configuration and diagnostics. Users may connect a notebook PC to the RS-232 port to configure or troubleshoot your system in the field. Further, the ADAM-5000 system can also be configured as the slave of the host computer through this port connection. The pin assignment of the port is as follows:



Pin No.	Description
Pin 1	Not Used
Pin 2	Data Send **(TXD)**
Pin 3	Data Receive **(RXD)**
Pin 4	Not Used
Pin 5	RS-232 Signal Ground (GND)
Pin 6	Not Used
Pin 7	Not Used
Pin 8	Not Used
Pin 9	Not Used



Build-in Communication Ports for Diagnostic Connection (ADAM-5000E only)

The Built-in Communication Ports for Diagnostic Function enables users to perform a quick diagnostics to locate where the system is at fault.

This Diagnostic Function requires the RS-485 port of ADAM-5000E to be connected to COM1 of host PC, and the RS-232 port of ADAM-5000E to COM2 of the previous host PC or other PCs. Then you should install software such as ComWatch or Hyperterminal and so on to monitor the commands that are being issued and the subse- quent responses from connected modules.

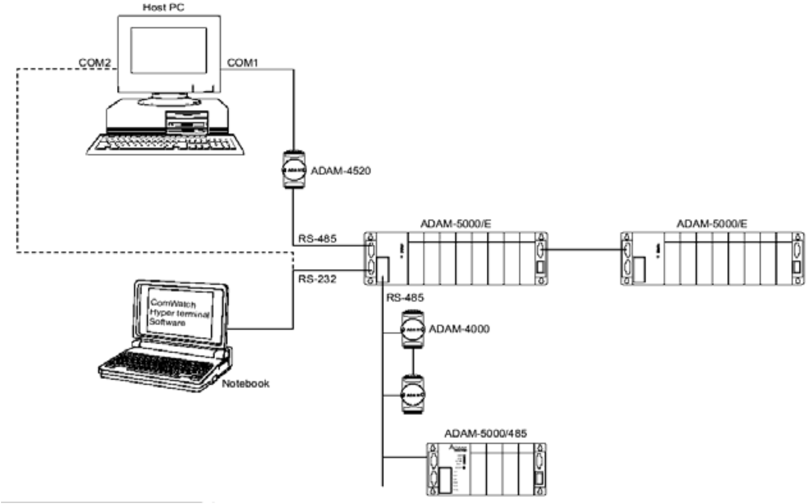


Figure 2-8 *Build-in Communication Ports for Diagnostic Connection*

Flexible Communication Port Function Connection(ADAM-5000E only)

The Flexible Communication Port Function prevents ADAM-5000E from system glitches due to communication line problems.

This function enables simultaneous connections via COM1 and COM2 port of your host PC to the RS-232 and RS-485 port of ADAM-5000E specifically. While working in conjunction with specific HMI software (e.g. AFX, FIX) that offers *COM Port Backup Function*, ADAM-5000E can circumvent failed communication on one port by switching to another available port to continue program execution.

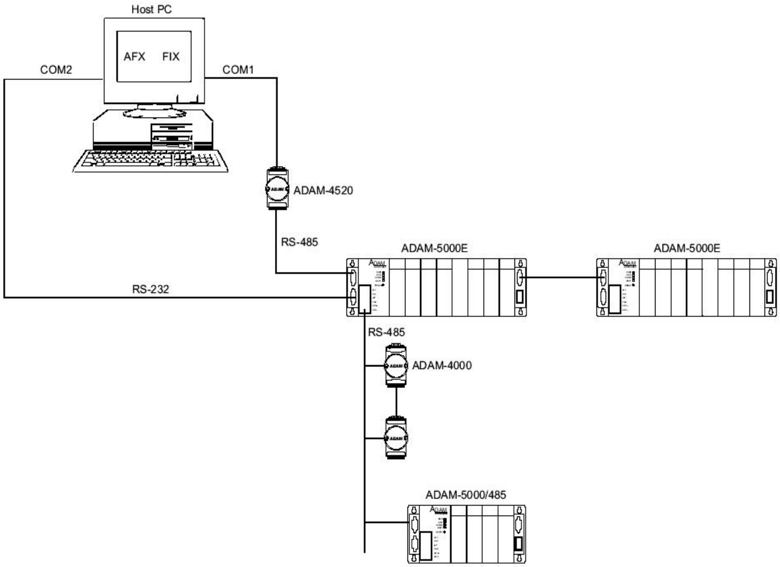


Figure 2-9 *Flexible Communication Port Function Connection*

3

ADAM-5000 System

3.1 Overview

The ADAM-5000 series is a data acquisition and control system which can control, monitor and acquire data through multichannel I/O modules. Encased in rugged industrial grade plastic bases, the systems provide intelligent signal conditioning, analog I/O, digital I/O, RS-232 and RS-485 communication. The ADAM-5000/485 can handle up to any 4 combinations of I/O modules (64 I/O points), while the ADAM-5000E can handle up to 8 combinations of I/O modules (128 I/O points). The systems communicate with their controlling host over a multi-drop RS-485 network.

3.2 Major Features of the ADAM-5000 System

The ADAM-5000 system consists of two major parts: the system kernel and I/O modules. The system kernel includes a CPU card, power regulator, 4-slot base, 8-slot base, built-in RS-232 communication port, and a pair of built-in RS-485 ports. It also offers the following major features:

The CPU's Basic Functions

The CPU is the heart of the system and has the following basic functions:

- Data acquisition and control for all I/O modules in the system
- Linearization of T/C (Thermocouple)
- Communication software and command set
- Calibration software and command set
- Alarm monitoring
- Management of the EEPROM device that holds the system parameters
- Data transformation
- Diagnosis

Diagnosis

There are 4 LEDs (indicated as PWR, RUN, TX and RX) to provide visual information on the general operation of the ADAM-5000 system. The LEDs also indicate the error status when the ADAM-5000 system performs the self test. Besides the LED indica- tors, the system also offers software diagnosis via the RS-232 port. For details, refer to Chapter 7.

3-Way Isolation and Watchdog Timer

Electrical noise can enter a system in many different ways. It may enter through an I/O module, a power supply connection or the communication ground connection. The ADAM-5000 system provides isolation for I/O modules (3000 V_{DC}), communication connection (2500 V_{DC}) and communication power connection (3000 V_{DC}). The 3-way isolation design prevents ground loops and reduces the effect of electrical noise to the system. It also offers better surge protection to prevent dangerous voltages or spikes from harming your system. The system also provides a Watchdog timer to monitor the microprocessor. It will automatically reset the microprocessor in ADAM-5000 system if the system fails.

Remote Software Configuration and Calibration

The ADAM-5000 system merely issues a command from the host computer, you can change an analog input module to accept several ranges of voltage input, current input, thermocouple input or RTD input. With the exception of system node address, all the parameters including speed, parity, HI and LO alarm, and calibration parameters setting may be set remotely. Remote configuration can be done by using either the provided menu-based software or the command set's configuration and calibration commands. By storing configuration and calibration parameters in a nonvolatile EEPROM, the systems are able to retain these parameters in case of power failure.

Flexible Alarm Setting

The ADAM-5000 system provides a flexible alarm setting method via an utility software (ADAM.EXE) between analog input modules and digital output modules. The user may configure a point of any digital output module plugged into any slot as the High alarm or Low alarm output of a channel of an analog input module. The relationship and their High/Low alarm limits may be downloaded into the system‘s EEPROM by the host computer.

The alarm functions can be enabled or disabled remotely. When the alarm function is enabled, the user may select whether the digital output is triggered. If the digital outputs are enabled, they are used to indicate the High and Low Alarm state. The High and Low alarm states can be read at any time by the host computer.

Every A/D conversion will be followed by a comparison with the High and Low limit. When the input value is over the High limit or below the Low limit, the High or Low alarm state is set to ON.

There are two alarm mode options: Momentary and Latching.

If the alarm is in Latching mode, the alarm will stay on even when the input value returns within limits. An alarm in Latching mode can be turned OFF by issuing a Clear Alarm command from the host computer. A Latching alarm is cleared by the microprocessor when the opposite alarm is set

For example, the alarm is in latching mode and the High alarm is turned ON. When the module receives a value that is lower than the Low alarm limit, the microprocessor will clear the High alarm and turn the Low alarm ON.

When the alarm is in Momentary mode, the alarm will be turned ON when the input value is outside of alarm limits and OFF while the input value remains within alarm limits. The arrangement of coupling High and Low alarm states with digital outputs may be utilized to build ON/ OFF controllers that can operate without host computer involvement.

Connectivity and Programming

ADAM-5000 systems can connect to and communicate with all computers and terminals. They use either RS-232 or RS-485 transmission standards and communicate with ASCII format commands. However, users can only select and use one communication port at any time. All communications to and from the system are performed in ASCII, which means that ADAM-5000 systems can be programmed in virtually any high-level language. The details of all commands will be covered in Chapter 6.

Flexible Communication Connection

ADAM-5000's built-in RS-232/485 conversion capability enables users to freely choose either RS-232 port or RS-485 port to connect with host PC. When user select either port to connect with their host PC, the other port could be utilized according to their specific needs.

For example, if RS-232 port is selected for connection with host PC, the RS-485 port can be used for connection with DA&C modules (such as ADAM-5000/485, ADAM-5000, ADAM-4000 modules). Thus users save extra costs for another RS-232/485 conversion device (e.g. ADAM-4520).

Or if users select RS-485 port for host PC connection, the RS-232 port can then have different usage such as described in the following sections (see Built-in Communication Ports for Field Diagnostics and Flexible Communication Port)

Built-in Communication Ports for Diagnostics (ADAM-5000E only)

When users utilize application software to control their system, ADAM-5000E can provide another port to let user monitor at any time the communication quality and condition as a reference to maintenance and test. When error occurs in the system, users can perform a quick diagnostics to locate the fault. A considerable amount of troubleshooting efforts can be saved. For example, using popular ComWatch software, users can watch the current execution and response of a certain command. It is very convenient to identify whether it is communication or the hardware product that is causing the problem.

Flexible Communication Port(ADAM-5000E only)

ADAM-5000E provides a further application. Users can simultaneously connect COM1 and COM2 of host PC to RS-232 and RS-485 port of ADAM-5000E. When host PC issues a command through one of the COM ports but receives no response (the other port will serve as backup), the COM Port Backup Function of HMI software (e.g. AFX, FIX) will automatically switch to another COM port to continue program execution without undue influence on your system. Probability of a system crash has thus minimized.

3.3 System Setup

A Single System Setup thru the RS-232 Port

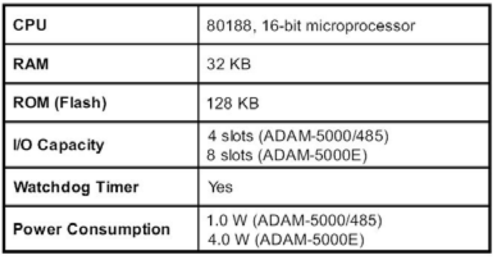
If users would like to use a PC to locally control and monitor a simple application, the ADAM-5000 system provides up to 64 points or 128 points and front-end wiring through the RS-232 port to the host computer.

A Distributed I/O Setup thru the RS-485 Network

The RS-485 network provides lower-noise sensor readings as the systems can be placed much closer to the source. Up to 256 ADAM-5000 systems may be connected to an RS-485 multi-drop network by using the ADAM-4510/4510S RS-485 repeaters, extending the maximum communication distance to 4,000 ft. The host computer is connected to the RS-485 network from one of its COM ports through the ADAM-4520/4522 RS-232/RS-485 converter.

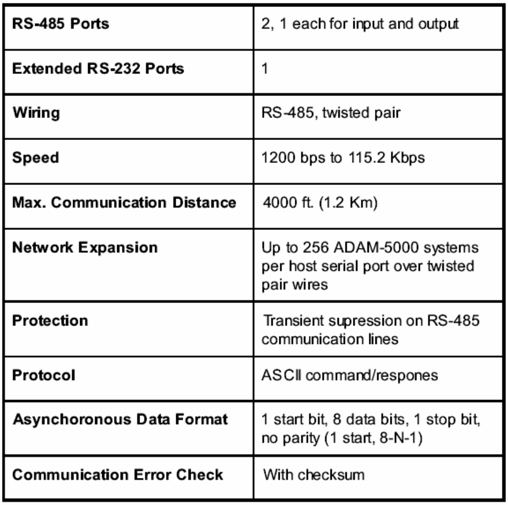
To boost the network's throughput, the ADAM-4510/4510S RS-485 repeaters use a logical RTS signal to manage the repeater's direction. Only two wires are needed for the RS-485 network: DATA+ and DATA-. Inexpensive, shielded twisted-pair wiring is employed.

3.4 Technical Specification of the ADAM-5000 Processor



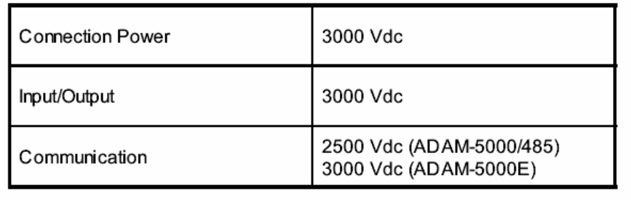
CPU	80188, 16-bit microprocessor
RAM	32 KB
ROM (Flash)	128 KB
I/O Capacity	4 slots (ADAM-5000/485) 8 slots (ADAM-5000E)
Watchdog Timer	Yes
Power Consumption	1.0 W (ADAM-5000/485) 4.0 W (ADAM-5000E)

Communication



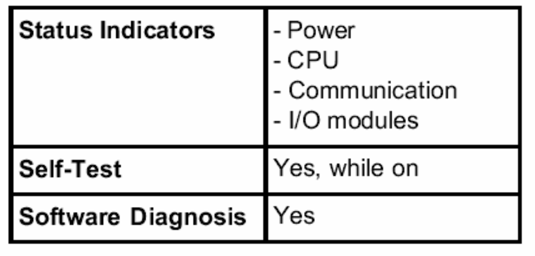
RS-485 Ports	2, 1 each for input and output
Extended RS-232 Ports	1
Wiring	RS-485, twisted pair
Speed	1200 bps to 115.2 Kbps
Max. Communication Distance	4000 ft. (1.2 Km)
Network Expansion	Up to 256 ADAM-5000 systems per host serial port over twisted pair wires
Protection	Transient supression on RS-485 communication lines
Protocol	ASCII command/respones
Asynchoronous Data Format	1 start bit, 8 data bits, 1 stop bit, no parity (1 start, 8-N-1)
Communication Error Check	With checksum

Isolation



Connection Power	3000 Vdc
Input/Output	3000 Vdc
Communication	2500 Vdc (ADAM-5000/485) 3000 Vdc (ADAM-5000E)

Diagnosis



Status Indicators	- Power - CPU - Communication - I/O modules
Self-Test	Yes, while on
Software Diagnosis	Yes

Basic Function Block Diagram

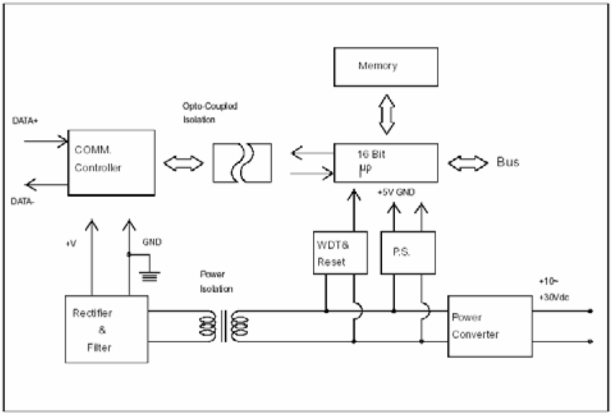


Figure 3.1 *Function block diagram*

4

I/O modules

This manual introduces the detail specifications functions and application wiring of each ADAM-5000 I/O modules. To organize an ADAM-5510 Series Controller, you need to select I/O modules to interface the main unit with field devices or processes that you have previously determined. Advantech provides 19 types of ADAM-5000 I/O modules for various applications so far. Following table is the I/ O modules support list we provided for user's choice.

More detailed specification and user's guides, please refer the user's manual of ADAM-5000 IO Module. It had integrated and collected this information.

Module	Name	Specification	Reference
Analog I/O	ADAM-5013	3-ch. RTD input	Isolated
	ADAM-5017	8-ch. AI	Isolated
	ADAM-5017H	8-ch. High speed AI	Isolated
	ADAM-5017UH	8-ch. Ultra High speed AI	Isolated
	ADAM-5018	7-ch. Thermocouple input	Isolated
	ADAM-5024	4-ch. AO	Isolated
Digital I/O	ADAM-5050	7-ch. D I/O	Non-isolated
	ADAM-5051	16-ch. DI	Non-isolated
	ADAM-5051D	16-ch. DI w/LED	Non-isolated
	ADAM-5051S	16-ch. Isolated DI w/LED	Isolated
	ADAM-5052	8-ch. DI	Isolated
	ADAM-5055S	16-ch. Isolated DI/O w/LED	Isolated
	ADAM-5056	16-ch. DO	Non-isolated
	ADAM-5056D	16-ch. DO w/LED	Non-isolated
	ADAM-5056S	16-ch. Isolated DO w/LED	Isolated
	ADAM-5056SO	16-ch. Iso. DO w/LED (source)	Isolated
Relay Output	ADAM-5060	6-ch. Relay output	Isolated
	ADAM-5068	8-ch. Relay output	Isolated
	ADAM-5069	8-ch. Relay output	Isolated
Counter/Frequency	ADAM-5080	4-ch. Counter/Frequency	Isolated
Serial I/O	ADAM-5090	4-port RS232	Non-isolated

Table 4-1: I/O Module Support List

5

Software Utilities

There is a software utility available to the ADAM-5000 systems. The Windows utility software helps you to configure your ADAM-5000. A DLL (Dynamic Link Library) driver is provided to write Windows applications, and a DDE (Dynamic Data Exchange) server is a service that links the ADAM-5000 systems to popular Windows packages such as Intouch, FIX DMACS, ONSPEC, Genesis and Excel.

5.1 ADAM-4000 and ADAM-5000 Windows Utility

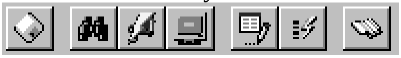
The ADAM-4000 and 5000 Windows Utility offers a graphical interface that helps you configure the ADAM-4000 and ADAM-5000 DA&C Modules. This windows utility makes it very convenient to monitor your Data Acquisition and Control system. The following guidelines will give you some brief instructions on how to use the utility.

- Overview
- COM port settings
- Search connected modules
- Terminal emulation
- Data Scope
- Save module's configuration to file
- Load module's configuration files to configure module
- Module configuration
- Module calibration
- Data input and output
- Alarm settings
- Download procedure

5.1.1 Overview

Main Menu

The window utility consists of a toolbar on the top and a display area that shows forth the relevant information about the connected modules. The utility's main toolbar is as shown below:



The main toolbar buttons are shortcuts to some commonly used menu items:

Save: Saves the connected module to PC.

Save the information of all connected modules to .txt file .By doing this; users can keep track of every different setting environment.

Search: Search for the address of connected modules on network.

Terminal: Issue commands and receives response.

Data Scope: Display the current data.

Save Configuration: Saves the configuration of selected module into txt file.

Load Configuration: Download the previous configuration file

Help: Display the Online Help for the ADAM utility.

5.1.2 Save Function

Save the settings of current module (e.g. Baud rate, Address, Modules Name) to txt file.

Example:

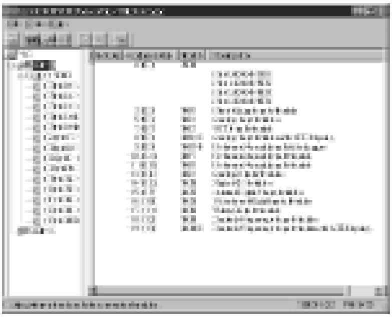


Figure 5-1 *Display the connected module*

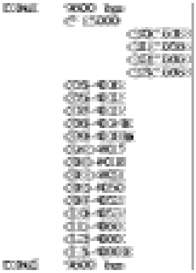


Figure 5-2 *Save the information of connected modules to txt file*

5.1.3 COM Port Settings

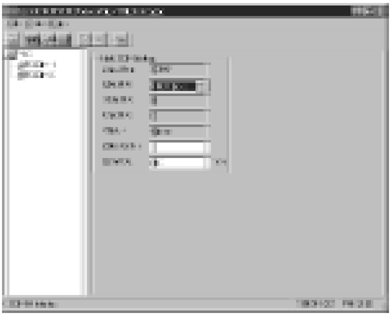


Figure 5-3 *Setup options*

Baud rate:

The communication speed (baud rate) can be configured from 1200 bps to 115.2 Kbps.

Prefix Char:

The Prefix Char is added to each ADAM command as follows:
[Prefix Char] + [ADAM Command]

Note: This is a special command only for ADAM-4521, ADAM-4541 and ADAM-4550.

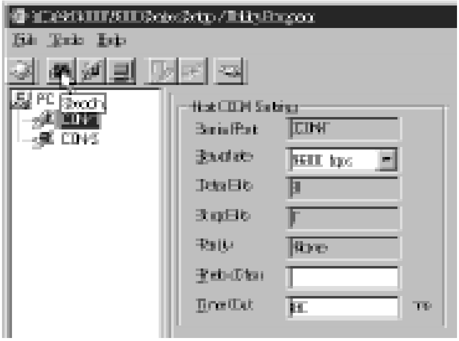
Timeout:

Timeout means the time limit for waiting a response after the system has issued a command. If no response has been received when timeout has passed, we’ll see the “Timeout!” message on the screen.

5.1.4 Search connected modules

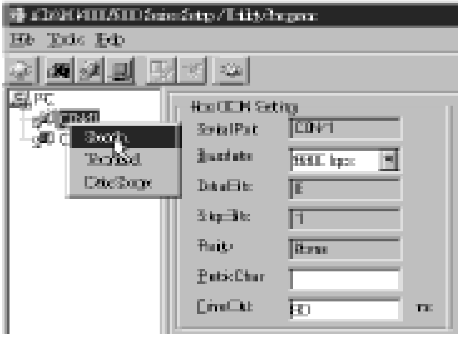
When you use the Search command, it will search for any connected modules on network and display their data. There are three ways to search for:

1. Click the Toolbar button:

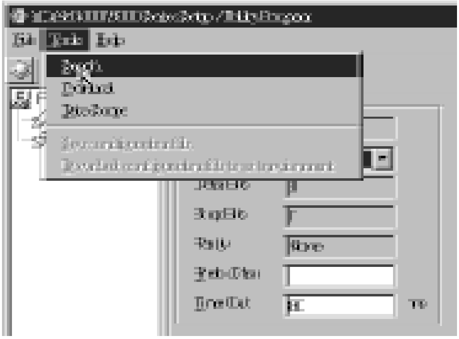


Host COM Setting
Serial Port: COM1
Baudrate: 9600 bps
Data Bits: 8
Stop Bits: 1
Parity: None
Prefix Char:
Time Out: 80 ms

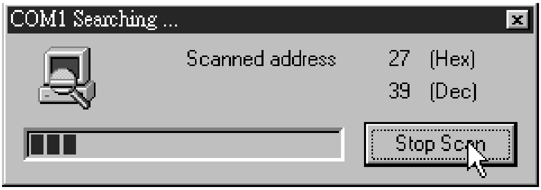
2. Click the right mouse button:



3. Click the Tools menu and choose the Search command:



4. The connected modules on network are currently being searched:


COM1 Searching ...

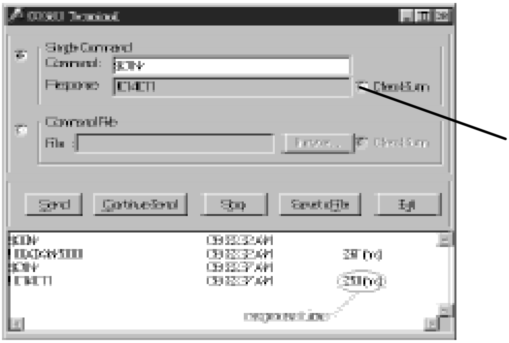
Scanned address 27 (Hex)

39 (Dec)

5.1.5 Terminal Emulation

You can issue commands and receive response by clicking the Terminal button. There are two ways to issue commands:

1. Issue single command:



Enable or Disable

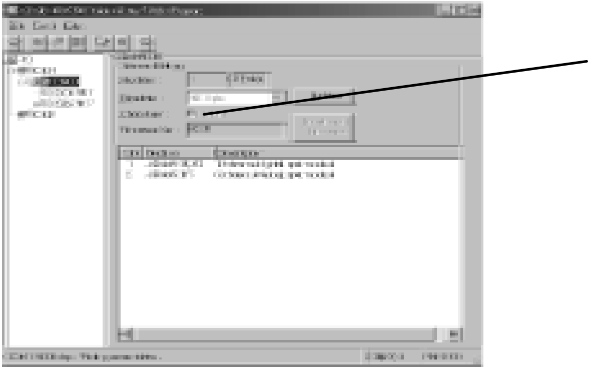
2. Batch command

Users can compose a sequence of commands and save them into a .txt file. Just click the Browse button to list all the .txt files available and select the file for continuous execution of the batch of commands therein.

3.

Exit Back to the main menu.

Note: If you select the checksum function on previous main menu, you have to select the checksum function in this menu.



Enable or Disable

__Figure 5-4__ Checksum function enabled

5.1.6 Data Scope

Data Scope enables you to monitor the issue of commands and the responses on another connected PC within your system. The following example illustrates the working connection for the Data Scope function:

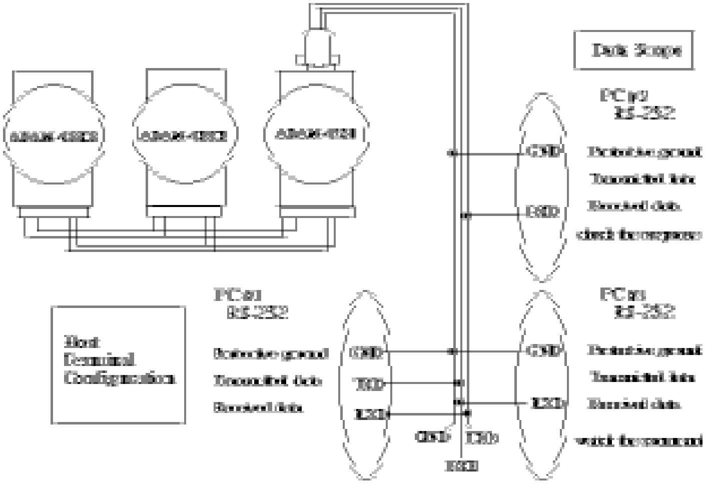
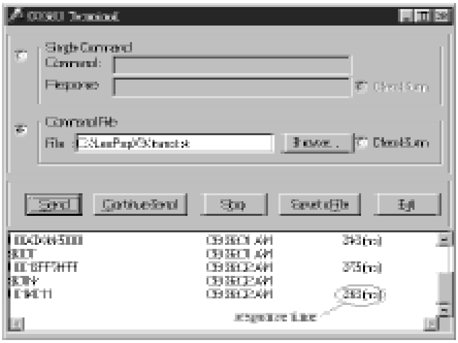


__Figure 5-5__ The connection for Data Scope function

When you issue commands from PC#1, you will get response.



Send : Send single command or batch command.

Continue Send : Send a single command or batch command repeatedly.

Stop : Stop issuing commands.

Save to File : Save history of the terminal emulation to txt file.

On PC#3, you can observe all commands issued from PC#1. Meanwhile, you can also observe all responses received at PC#2.

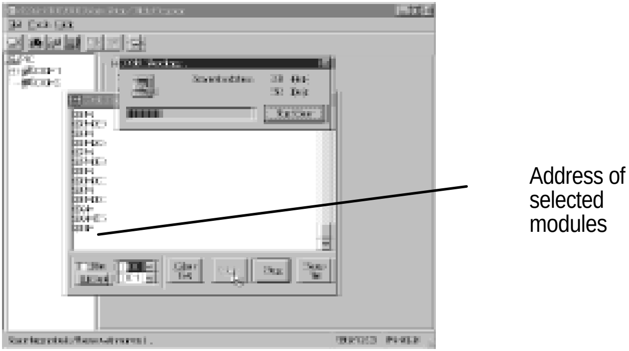


***Figure 5-4** Checksum function enabled*

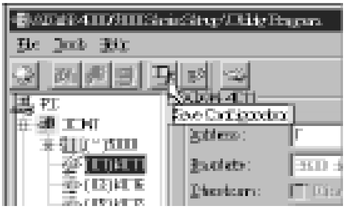
When your system is connected with multiple ADAM-4000 or ADAM-5000 modules, just click the addresses of the modules to see relevant information (multiple selection from 00 to FF is allowed). Then check the Filter option and click Update button to see relevant information of the modules. Note that the information about other unselected modules won't show forth.

5.1.7 Saving module's Configuration to file

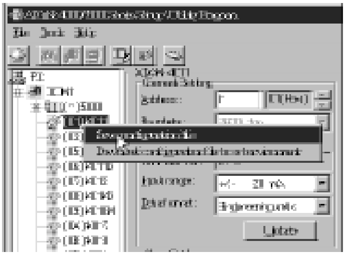
- Save the input range, baud rate, data format, checksum status and/ or integration time and Alarm Status for a specified analog input module.
- Save the output range, baud rate, data format, checksum status and slew rate for a specified analog output module.
- Save the baud rate and checksum status for a digital I/O module.
- Save the input mode, baud rate, checksum status and/or frequency gate time, input signal mode, gate mode, alarm status, etc. for a specified counter/frequency module.

There are three ways to save a configuration file:

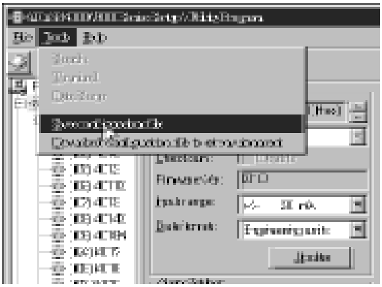
1. Click the Toolbar button



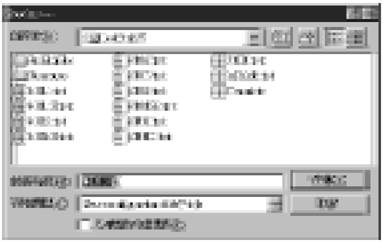
2. Click the right mouse button



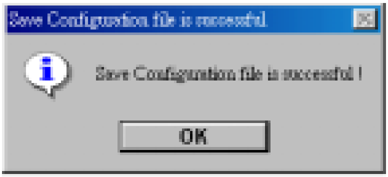
3. Click the Tools menu. Choose the “Save Configuration file” command



and then specify the file name.



The configuration file is now saved.



Save Configuration file is successful

Save Configuration file is successful !

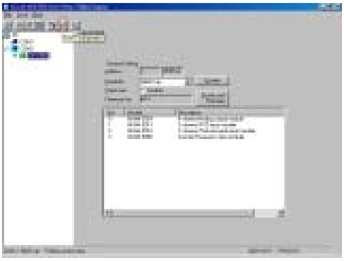
5.1.8 Load Module's Configuration File

- Reload previous settings. Sets the input range, baud rate, data format, checksum status and/or integration time and alarm status for a specified analog input module.
- Sets the output range, baud rate, data format, checksum status and slew rate for a specified analog output module.
- Sets the baud rate and checksum status for a digital I/O module.
- Sets the input mode, baud rate, checksum status and/or frequency gate time, input signal mode, gate mode, alarm status, etc. for a specified counter/frequency module.

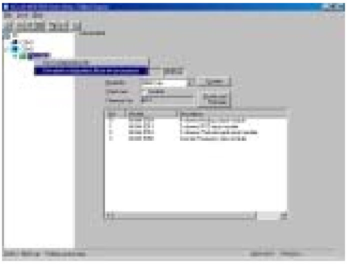
Note: *Baud rate and checksum can only be changed in the INIT* state. Changed settings can only take effect after a module is rebooted.*

There are three ways to load a configuration file:

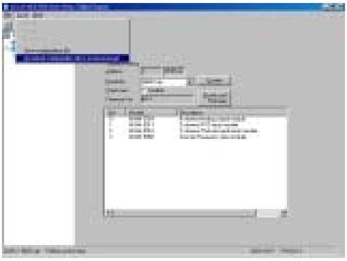
1. Click the Toolbar button:



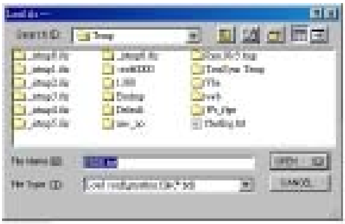
2. Click the right mouse button:



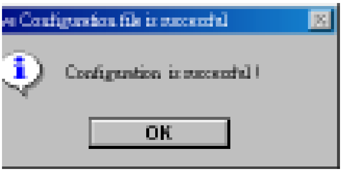
3. Click the Tools menu and choose Download configuration file to set the environment command:



4. Choose the file name:



The configuration file is now loaded.



...Configuration file is successful

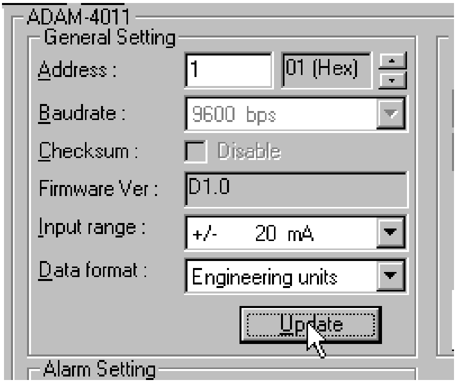
Configuration is successful !

5.1.9 Module Configuration

- Sets the input range, baud rate, data format, checksum status, and/or integration time for a specified analog input module.
- Sets the output range, baud rate, data format, checksum status and slew rate for a specified analog output module.
- Sets the baud rate and checksum status for a digital I/O module.
- Sets the input mode, baud rate, checksum status and/or frequency gate time for a specified counter/frequency module.

Note: Baud rate and Checksum can only be changed in the INIT* state. Changed settings only take effect after a module is rebooted.

For Example: The configuration of ADAM-4011



Address : 1 01 (Hex)

Baudrate : 9600 bps

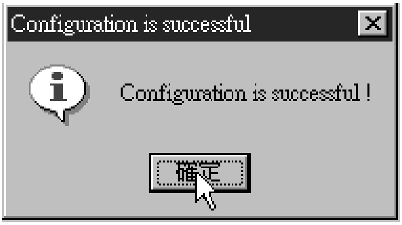
Checksum : □ Disable

Firmware Ver : D1.0

Input range : +/- 20 mA

Data format : Engineering units

- **Address:** Represents the address of the module. The Range is from 0 to 255.
- **Baudrate:** Represents the baud rate.
- **Checksum:** Represents the checksum status, i.e., Disabled/ Enabled.
- **Firmware Ver:** Represents the version of firmware.
- **Input range:** Represents the input range of modules. You can refer to Chapter 4.
- **Data format:** Represents the data format (e.g. engineering format). You can refer to Chapter 4.


Configuration is successful

Configuration is successful !

5.1.10 Module Calibration

Calibration is to adjust the accuracy of ADAM module. There are several modes for module's calibration: Zero calibration and span calibration. Only analog input and output modules can be calibrated. Note: The calibrating function supports ADAM-5013/5017/5017H/ 5018/5024.

Zero Calibration:

(1). Apply power to the module and let it warm up for about 30 minutes.

(2). Make sure that the module is correctly installed and is properly configured for the input range you want to calibrate.

(3). Use a precision voltage source to apply a calibration voltage to the modules' terminals of the specific channel.

(4). Click the "Zero Calibration" button. See Figure 5-7

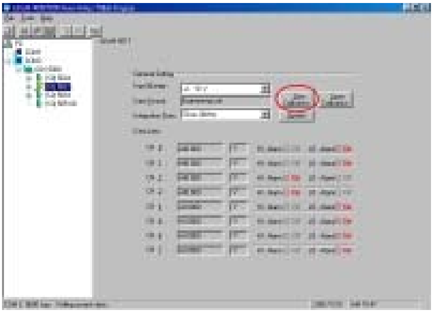
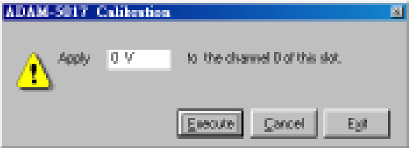


***Figure 5-7** Zero Calibration*

(5). Click the Execute button to begin the calibration



ADAM-5017 Calibration

Apply 0 V to the channel 0 of this slot.

***Figure 5-8** : Execute Zero Calibration*

Span Calibration:

(1). Use a precision voltage source to apply a calibration voltage to the modules’ terminals of the specific channel.

(2). Click the “Span Calibration” button. See Figure 5-9

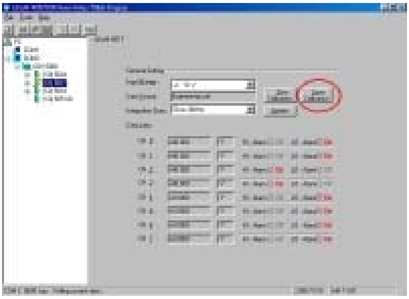
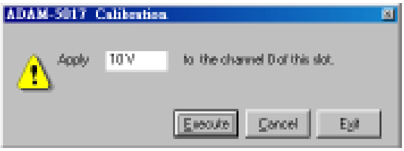


***Figure 5-9** : Span Calibration*

(3). Click the Execute button to begin the calibration



ADAM-5017 Calibration

Apply 10 V to the channel 0 of this slot.

Figure 5-10 *: Execute Span Calibration*

CJC Calibration:

CJC (cold junction sensor) calibration only applies to the ADAM-5018

(1). Prepare a voltage source which is accurate to the mV level.

(2). Run the zero calibration and span calibration function.

(3). Use a temperature emulation device (such as Micro-10) to send a temperature signal to the ADAM module and then compare this signal with the reading from the ADAM module. If the reading value is different from the signal, adjust the CJC value to improve it.

(4). Click the “CJC Calibration” button. See Figure 5-11.

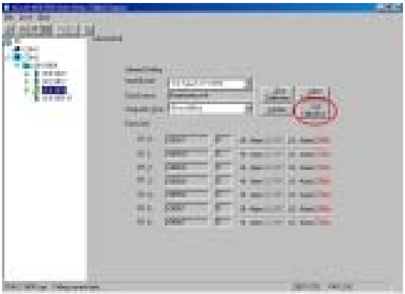
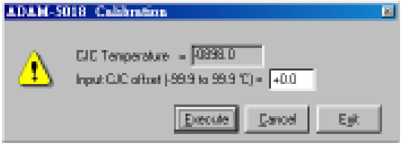


Figure 5-11 *: CJC Calibration*

.

(5). Click the Execute button to begin the calibration



ADAM-5018 Calibration

CJC Temperature = 0898.0

Input CJC offset (-99.9 to 99.9 °C) = +0.0

***Figure 5-12** : Execute CJC Calibration*

Analog Input Resistance Calibration:

RTD sensor calibration only applies to the ADAM-5013

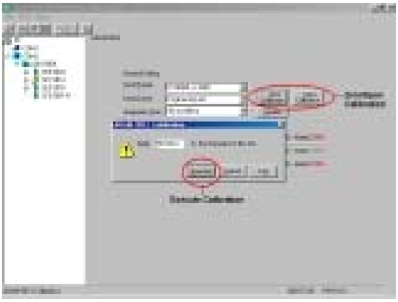


***Figure 5-13** : RTD Module Calibration*

Analog Output Calibration:

4~20 mA: ADAM 5024

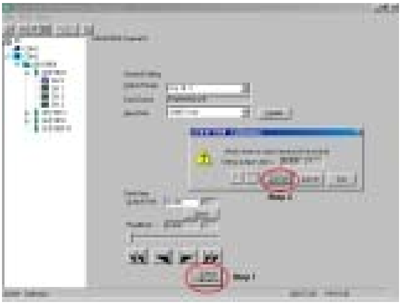


***Figure 5-14** : Analog Output Calibration*

5.1.11 Data Input and Output

Analog Input Module with Digital Output

- The function can only be used when the alarm status is “Disable”.

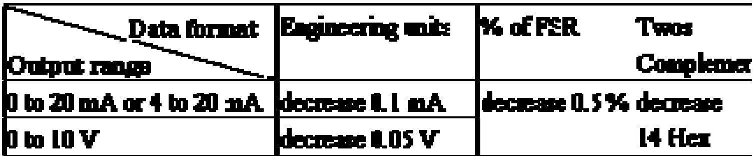
Digital Output Module

- Click the item to turn it on or off.

Analog Output Module

Send Enter a value that users want to get

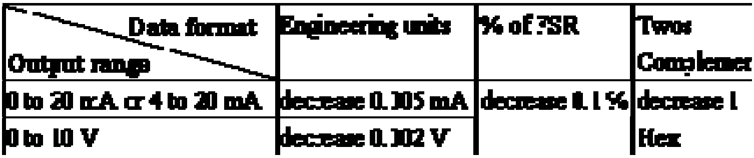
- ◀◀
- Fast Decrease



Data format Output range	Engineering units	% of FSR	Twos Complemen
0 to 20 mA or 4 to 20 mA	decrease 0.1 mA	decrease 0.5 %	decrease 14 Hex
0 to 10 V	decrease 0.05 V		

◀

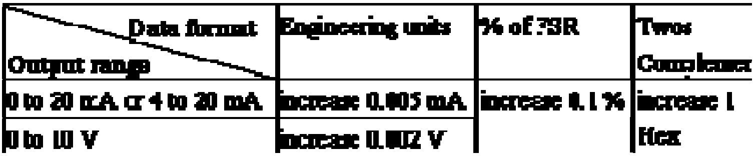
decrease



Data format Output range	Engineering units	% of FSR	Twos Complemen
0 to 20 mA or 4 to 20 mA	decrease 0.005 mA	decrease 0.1 %	decrease 1 Hex
0 to 10 V	decrease 0.002 V		

▶

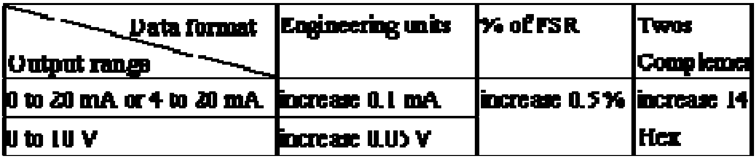
increase



Data format Output range	Engineering units	% of FSR	Twos Complemen
0 to 20 mA or 4 to 20 mA	increase 0.005 mA	increase 0.1 %	increase 1 Hex
0 to 10 V	increase 0.002 V		

▶▶

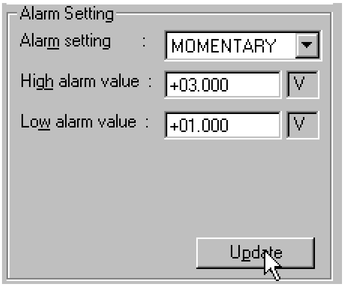
- fast increase



Data format Output range	Engineering units	% of FSR	Twos Complemen
0 to 20 mA or 4 to 20 mA	increase 0.1 mA	increase 0.5 %	increase 14 Hex
0 to 10 V	increase 0.05 V		

5.1.12 Alarm Setting

- Set the alarm status, high alarm value, low alarm value, and then click the *Update* button.



Alarm Setting

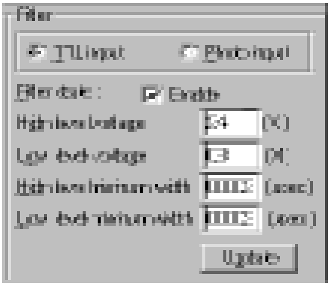
Alarm setting : MOMENTARY

High alarm value : +03.000 V

Low alarm value : +01.000 V

- **Alarm setting:** Disables or enables the alarm either in Latching or Momentary mode.
- **High alarm value:** Downloads the high alarm limit value into the module. The format is always in engineering units.
- **Low alarm value:** Downloads the low alarm limit value into the module. The format is always in engineering units.

Digital Filter



Filter

● TTL input ○ Photo input

Filter state : ☑ Enable

High level voltage 2.4 (V)

Low level voltage 0.8 (V)

High level minimum width [illegible] (usec)

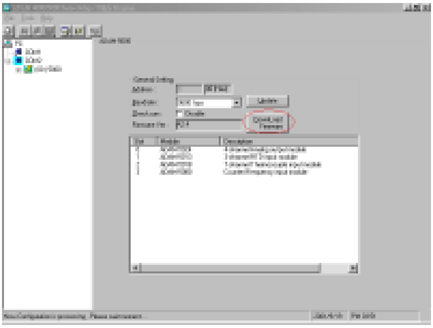
Low level minimum width [illegible] (usec)

- **High level voltage:** Set the high trigger level for non-isolated input signals. The range is from 0.1 V to 5.0 V. This high trigger level must be higher than the low trigger level at all times.

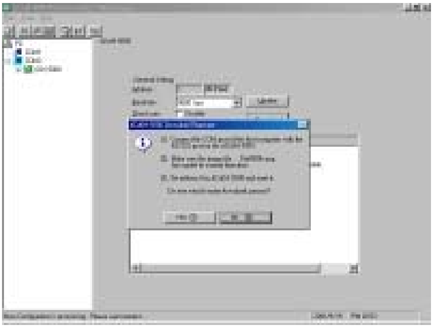
- **Low level voltage:** Set the low trigger level for non-isolated input signals. The range is from 0.1 V to 5.0 V.
- **High level minimum width:** Set the minimum width at high level. The unit is µsec (microseconds) and its resolution is 1 µsec. Users can set value from 2 to 65535.
- **Low level minimum width:** Set the minimum width at low level. The unit is µsec (microseconds) and its resolution is 1 µsec. Users can set value from 2 to 65535.

5.1.13 Download Procedure

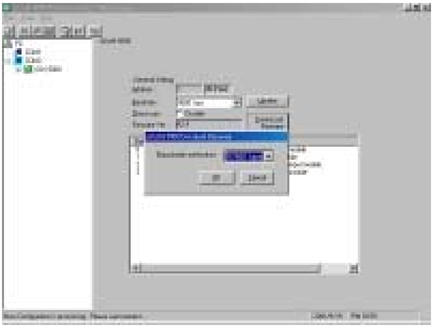
1. Click the Download Firmware button.



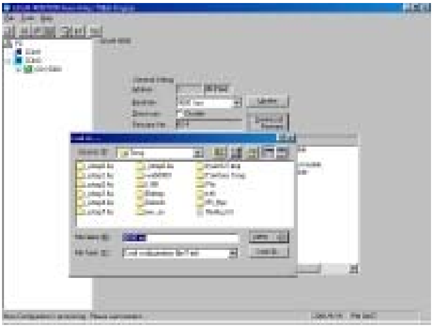
2. Click Yes.



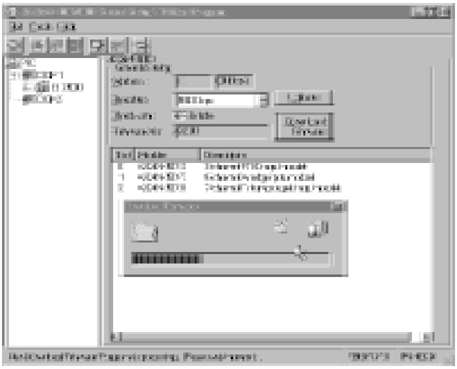
3. Choose the baud rate.



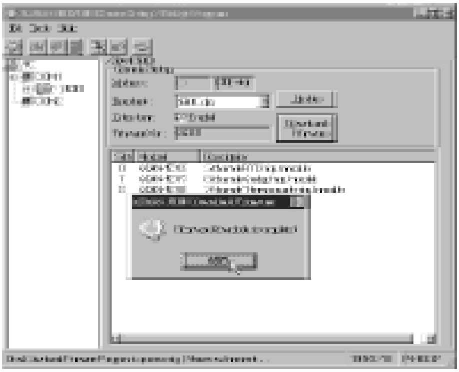
4. Choose Download file.



Firmware downloads in progress.



Firmware downloads complete.



NOTICE: THE FIRMWARE UPGRAD OPERATION IS ONLY USED BY RS-232 PORT.

5.2 DLL (Dynamic Link Library) Driver

The ADAM-5000 API Dynamic Link Library (DLL) enables you to quickly and easily write Windows applications for ADAM-5000 systems. The library supports both C++ and Visual Basic. Since ADAM-5000 systems communicate with a host computer through the host‘s COM port, no additional driver (DRV or VxD) needs to be installed. The DLL includes all necessary function calls to utilize the ADAM-5000 systems to their fullest extent.

Together with the DLL driver you‘ll find the source code of a Visual Basic example on your diskette. The example provides several control windows to communicate with all types of ADAM-5000 modules. You can customize the source code to create your own tailor-made
ADAM-5000 setup program or monitoring system.

For details on the ADAM-5000 function calls refer to the Help file included on the ADAM-5000 API diskette.

5.3 DDE (Dynamic Data Exchange) Server

The ADAM-5000 DDE server takes advantage of DDE, a built-in Windows communication service. The DDE server acquires data from the ADAM-5000 systems and passes it to your application program via the hot link (DDE). The software can also pass control and configuration commands to the ADAM-5000 systems using the DDE protocol. You can now use ADAM-5000 systems with most Windows- based data acquisition software that supports DDE. Examples include Intellution's FIX DMACS, Wonderware’s InTouch, ONSPEC, Paragon and Excel.

For details on the ADAM-5000 DDE server refer to the DDE server manual for the ADAM-5000.

6

Command Set

Command Set

6.1 Introduction

To avoid communication conflicts when several devices try to send data at the same time, all actions are instigated by the host computer. The basic form is a command/response protocol with the host initiating the sequence.

When systems are not transmitting they are in listen mode. The host issues a command to a system with a specified address and waits a certain amount of time for the system to respond. If no response arrives, a time-out aborts the sequence and returns control to the host. Changing ADAM-5000/5000E system's configuration might require the system to perform auto calibration before changes can take effect. This is especially true when changing the range as the system has to perform all stages of auto calibration that it performs when booted. When this process is underway, the system does not respond to any other commands. The command set includes information on the delays that might occur when systems are reconfigured.

6.2 Syntax

Command Syntax: [delimiter character][address][slot][channel] [command][data][checksum][carriage return]

Every command begins with a delimiter character. There are four valid characters: $, #, % and @.

The delimiter character is followed by a two character address (hexadecimal) that specifies the target system. The two characters following the address specify the module slot and channel.

Depending on the command, an optional data segment may follow the command string. An optional two character checksum may also be appended to the command string. Every command is terminated with a carriage return (cr).

Note: *All commands should be issued in UPPERCASE characters only!*

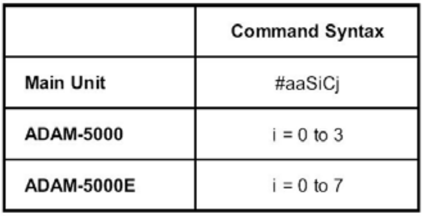
The command set is divided into the following five categories:

- CPU Command Set
- Analog Input Command Set
- Analog Input Alarm Command Set
- Analog Output Modules Command Set
- Digital I/O Modules Command Set

Every command set category starts with a command summary of the particular type of module, followed by datasheets that give detailed information about individual commands.

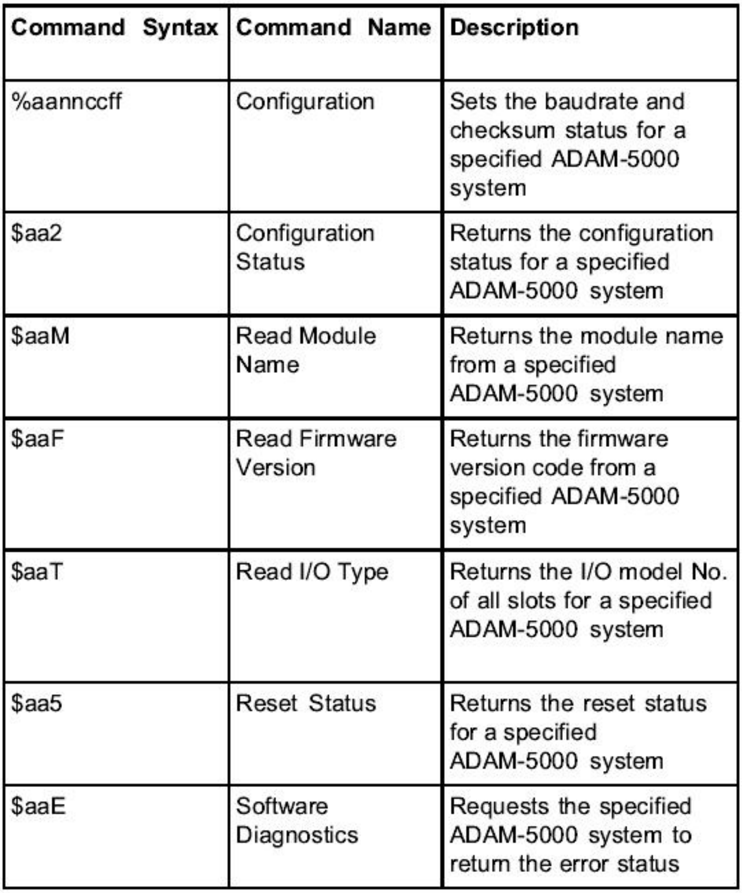
Although commands in different subsections sometime share the same format, the effect they have on a certain module can be completely different than that of another. Therefore, the full command set for each type of module is listed along with a description of the effect the command has on the given module.

Note: *Before setting commands, the user needs to know the type of main unit being used. If ADAM-5000/485 is being used, the "i" in Si can be set at 0 to 3. If ADAM-5000E is being used, the "i" in Si can be set at 0 to 7. This is illustrated in the table below:*



	Command Syntax
Main Unit	#aaSiCj
ADAM-5000	i = 0 to 3
ADAM-5000E	i = 0 to 7

6.3 CPU Command Set



Command Syntax	Command Name	Description
%aannccff	Configuration	Sets the baudrate and checksum status for a specified ADAM-5000 system
$aa2	Configuration Status	Returns the configuration status for a specified ADAM-5000 system
$aaM	Read Module Name	Returns the module name from a specified ADAM-5000 system
$aaF	Read Firmware Version	Returns the firmware version code from a specified ADAM-5000 system
$aaT	Read I/O Type	Returns the I/O model No. of all slots for a specified ADAM-5000 system
$aa5	Reset Status	Returns the reset status for a specified ADAM-5000 system
$aaE	Software Diagnostics	Requests the specified ADAM-5000 system to return the error status

%aanncсff

Name Configuration

Description Sets baud rate and checksum status for a specified ADAM-5000 system.

Syntax **%aannccff(cr)**

% is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to configure.

nn is reserved for system use. Its default value is 00h.

cc represents the baud rate code.

ff is a hexadecimal number that equals the 8-bit parameter representing checksum status. The sixth bit represents the checksum status; 1 means enabled while 0 means disabled. The other bits are not used and are set to 0.

(cr) is the terminating character, carriage return (0Dh).

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid parameter was entered or if the INIT* terminal was not grounded when attempting to change baud rate or checksum settings. There is no response if the module detects a syntax error, communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was in- valid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

%aannccff

(cr) is the terminating character, carriage return (0Dh).

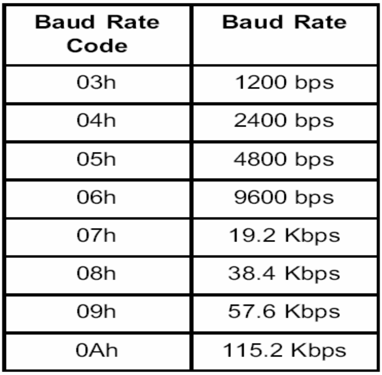
Example command: **%23000A40(cr)**

response: **!23(cr)**

The ADAM-5000 system with address 23h is configured to a baud rate of 115.2 Kbps and with checksum generation or validation.

The response indicates that the command was received. Wait 7 seconds to let the new configuration setting take effect before issuing a new command to the system.

Note: *All configuration parameters can be changed dynamically, except checksum and baud rate parameters. They can only be altered when the INIT* terminal is grounded.*



Baud Rate Code	Baud Rate
03h	1200 bps
04h	2400 bps
05h	4800 bps
06h	9600 bps
07h	19.2 Kbps
08h	38.4 Kbps
09h	57.6 Kbps
0Ah	115.2 Kbps

Figure 6-1 *Baud rate codes*

$aa2

Name Configuration Status

Description Returns the configuration status for a specified system module.

Syntax **$aa2(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

2 is the Configuration Status command.

(cr) is the terminating character, carriage return (0Dh).

Response **!aaccff(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was in- valid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

cc represents the baud rate code.

ff is a hexadecimal number that equals the 8-bit parameter representing checksum status. The sixth bit represents the checksum status; 1 means enabled while 0 means disabled. The other bits are not used and are set to 0.

(cr) is the terminating character, carriage return (0Dh).

$aa2

(See also the **%aannccff** configuration command)

Example command: **$452(cr)**

response: **!450600(cr)**

The command requests the ADAM-5000 system at address 45h to send its configuration status.

The ADAM-5000 system at address 45h responds with a baud rate of 9600 bps and with no checksum function or checksum generation.

$aaM

Name Read Module Name

Description Returns the module name from a specified ADAM-5000 system.

Syntax **$aaM(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

M is the Module Name command.

(cr) is the terminating character, carriage return (0Dh).

Response **!aa5000(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was in- valid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

#aaF

Example command: **$15M(cr)**

response: **!155000(cr)**

The command requests the system at address 15h to send its module name.

The system at address 15h responds with module name **5000** indicating that there is an ADAM-5000 at address 15h.

$aaF

Name Read Firmware Version

Description Returns the firmware version code from a specified ADAM-5000 system.

Syntax **$aaF(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

F is the Firmware Version command.

(cr) is the terminating character, carriage return (0Dh).

Response **!aa(version)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(version) represents the firmware version of the ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

$aaF

Example command: **$17F(cr)**

response: **!17A1.06(cr)**

The command requests the system at address 17h to send its firmware version.

The system responds with firmware version **A1.06.**

$aaT

Name	Read I/O Type
Description	Returns the I/O module no. of all slots for a specified ADAM-5000 system.

Syntax **$aaT(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

T is the I/O Module Types command.

(cr) is the terminating character, carriage return (0Dh).

Response **!aabbccddee(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was in- valid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

bb, cc, dd, ee represent the I/O Module No. of all slots from slot 0 thru 3 of the ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

$aaT

Example command: **$12T(cr)**

response: **!1218245160(cr)**

The command requests the ADAM-5000 system at address 12h to send all existing I/O module numbers. The system at address 12h responds with I/O module numbers 18, 24, 51 and 60 in slots 0-3. This means that the ADAM-5000 system contains an ADAM-5018, ADAM-5024, ADAM-5051 and ADAM-5060 in slots 0 thru 3.

$aa5

Name Reset Status

Description Checks the reset status of the addressed ADAM-5000 system to see whether it has been reset since the last Reset Status command was issued to the ADAM-5000 system.

Syntax **$aa5(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system whose Reset Status is to be returned.

5 is the Reset Status command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aas(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

s represents the Status bit that is returned by the ADAM-5000 system. If s=1, the system has been reset or powered up since the last time it was issued a Reset Status command. If s=0, the system has not been reset.

(cr) is the terminating character, carriage return (0Dh)

$aa5

Example command: **$395(cr)**

response: **!391(cr)**

The ADAM-5000 system at address 39h was reset or powered up since the last Reset Status command was issued.

$aaE

Name Software Diagnostics

Description Requests the specified ADAM-5000 system to return the error status

Syntax **$aaE(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

E is Software Diagnostics command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aabbccddee(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

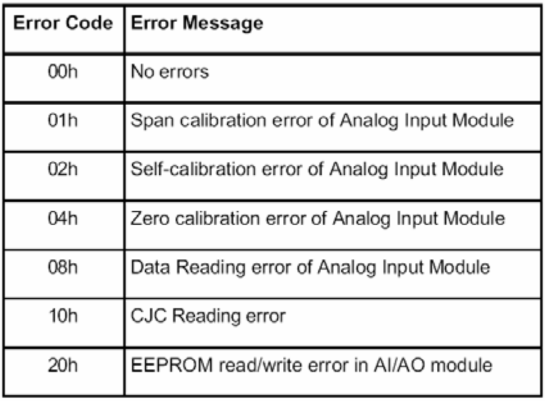
! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

bbccddee are hexadecimal numbers representing the error code from slot 0 thru slot 3 of the system.

(cr) is the terminating character, carriage return (0Dh)



Error Code	Error Message
00h	No errors
01h	Span calibration error of Analog Input Module
02h	Self-calibration error of Analog Input Module
04h	Zero calibration error of Analog Input Module
08h	Data Reading error of Analog Input Module
10h	CJC Reading error
20h	EEPROM read/write error in AI/AO module

Figure 6-2 *Analog module error codes*

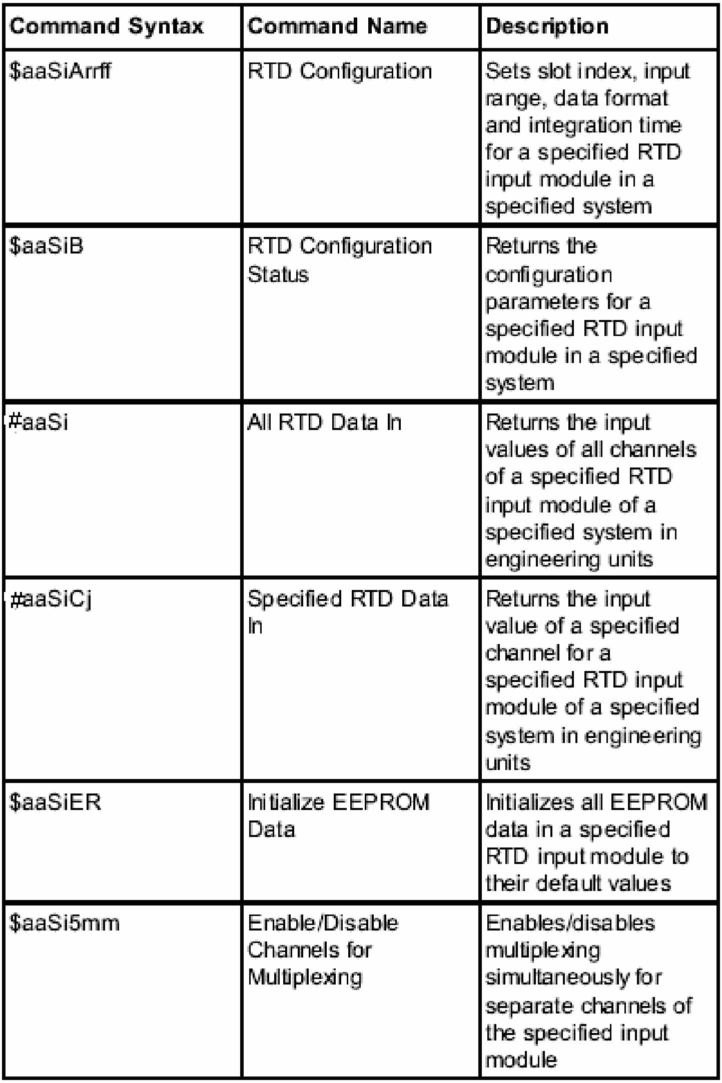
Example: command: **$01E(cr)**

response: **!0100000001**

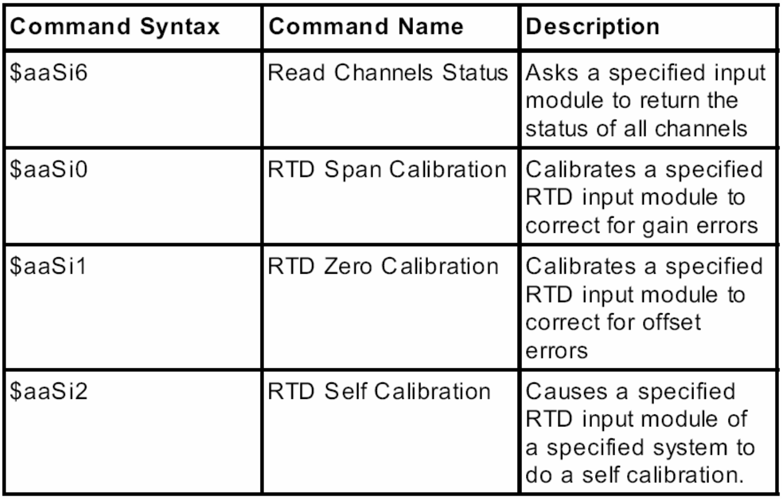
The command diagnoses the system at address 01h and responds with its error status code.

The system responds that the module in slot 3 has a span calibration error.

6.4 ADAM-5013 RTD Input Command Set



Command Syntax	Command Name	Description
$aaSiArrff	RTD Configuration	Sets slot index, input range, data format and integration time for a specified RTD input module in a specified system
$aaSiB	RTD Configuration Status	Returns the configuration parameters for a specified RTD input module in a specified system
#aaSi	All RTD Data In	Returns the input values of all channels of a specified RTD input module of a specified system in engineering units
#aaSiCj	Specified RTD Data In	Returns the input value of a specified channel for a specified RTD input module of a specified system in engineering units
$aaSiER	Initialize EEPROM Data	Initializes all EEPROM data in a specified RTD input module to their default values
$aaSi5mm	Enable/Disable Channels for Multiplexing	Enables/disables multiplexing simultaneously for separate channels of the specified input module



Command Syntax	Command Name	Description
$aaSi6	Read Channels Status	Asks a specified input module to return the status of all channels
$aaSi0	RTD Span Calibration	Calibrates a specified RTD input module to correct for gain errors
$aaSi1	RTD Zero Calibration	Calibrates a specified RTD input module to correct for offset errors
$aaSi2	RTD Self Calibration	Causes a specified RTD input module of a specified system to do a self calibration.

Note: *The ADAM-5013 module also has "Alarm Setting" functions. The alarm command set for the ADAM-5013 is the same as that for the ADAM-5017, ADAM-5017H, and the ADAM-5018. Please refer to pages 6-71 to 6-89 for this set of commands.*

$aaSiArrff

Name RTD Configuration

Description Sets slot index, input range, data format and integration time for a specified RTD input module in a specified system.

Syntax **$aaSiArrff(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to configure.

Si identifies the desired slot i (i:0to3).

A represents the I/O module configuration command.
rr represents the 2-character hexadecimal code of the input range. (See Appendix B)

ff is a hexadecimal number that equals the 8-bit parameter representing data format. Bits 0 and 1 represent data format. Bit 7 represents integration time. The layout for the 8-bit parameter is shown in Figure 6-3 (See page 6-
38). The other bits are not used and are set to 0. (cr) is the terminating character, carriage return (0Dh).

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal

$aaSiArrff

address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

Example command: **$35S3A2000(cr)**

response: **!35(cr)**

The RTD input module in slot 3 of the ADAM-5000 system at address 35h is configured to an RTD type Pt -100 to 100° C, engineering unit data format, and integration time 50ms (60Hz). The response indicates that the command has been received.

$aaSiB

Name RTD Configuration Status

Description Returns the configuration parameters for a specified RTD input module in a specified system.

Syntax **$aaSiB(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the desired slot i (i:0to3)

B represents the configuration status command

(cr) is the terminating character, carriage return (0Dh).

Response. **!aarrff(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

rr represents the 2-character hexadecimal code of the input range. (See Appendix B)

ff is a hexadecimal number that equals the 8-bit parameter representing data format. Bits 0 and 1 represent data format. Bit 7 represents integration time (See RTD Configuration Command $aaSiArrff).

$aaSiB

(cr) is the terminating character, carriage return (0Dh).

Example command: **$35S3B(cr)**

response: **!352000(cr)**

The RTD input module in slot 3 of the ADAM-5000 system at address 35h responds with an RTD type Pt -100 to 100° C, engineering unit data format, and integration time 50ms (60Hz).

$aaSi

Name All RTD Data In

Description Returns the input values of all channels of a specified RTD input module in a specified system in engineering units only.

Syntax **$aaSi(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si is the I/O slot of the ADAM-5000 system you want to read.

(cr) is the terminating character, carriage return (0Dh).

Response. **>(data)(data)(data)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

> delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

(data) is the input value in engineering units of the interrogated module of the specified system. The (data) from all channels is shown in sequence from 0 to 2. If (data)=“ ”, it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh).

$aaSi

Example command: **$35S3(cr)**

response: **>+80.01 +20.00 -40.12(cr)**

The command requests the RTD input module in slot 3 of the ADAM-5000 system at address 35h to return the input values of all channels.

The RTD input module responds with input values of all channels in sequence from 0 to 2 : +80.01° C, +20.00° C, -40.12° C.

#aaSiCj

Name Specified RTD Data In

Description Returns the input value of a specified channel for a specified RTD input module of a specified system in engineering units only.

Syntax **#aaSiCj(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

SiCj identifies the desired slot i (i:0 to 3) and the desired channel j (j:0 to 2) of the module you want to interrogate.

(cr) is the terminating character, carriage return (0Dh).

Response. >**(data)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

> delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(data) is the input value in engineering units of the specified channel for the specified RTD input module of the specified system. If (data)=“ ”, it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh).

#aaSiCj

Example command: **$35S3C0(cr)**

response: **>+80.01(cr)**

The command requests the RTD input module in slot 3 of the ADAM-5000 system at address 35h to return the input value of channel 0. The RTD input module responds that the input value of channel 0 is +80.01° C.

$aaSiER

Name Initialize EEPROM Data

Description Initializes all EEPROM data in a specified analog input module to their default values. This command is sent following a failed attempt to calibrate a module (the module shows no effect from an attempted calibration). Following initialization, the problem module should readily accept calibration.

Syntax **$aaSiER(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot in which you wish to initialize all EEPROM data.

ER represents the initialize EEPROM data command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSi5mm

Name Enable/Disable Channels for multiplexing

Description Enables/Disables multiplexing for separate channels of the specified input module

Syntax **$aaSi5mm(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot of the system.

5 represents the enable/disable channels command.

mm are two hexadecimal values. Each value is interpreted by the module as 4 bits. The first 4-bit value is 0. The second 4-bit value represents the status of channels 0 to 3. A value of 0 means the channel is disabled, while a value of 1 means the channel is enabled. (See the Read Channel Status Command $aaSi6).

Note: Bit 4 can not enable a channel in the ADAM-5013 since the module is physically limited to 3 channels.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was

$aaSi5mm

invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

Example command: **$00S1501(cr)**

response: **!00(cr)**

The command enables/disables the channels of the analog input module in slot 1 of the system at address 00h. Hexadecimal 0 is a fixed value. Hexadecimal 1 equals binary 0001, which enables channel 0 and disables channels 1 and 2.

$aaSi6

Name Read Channels Status

Description Asks a specified input module to return the status of all channels

Syntax **$aaSi6(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the I/O slot of the system you want to read channels status. The channel status defines whether a channel is enabled or disabled.

6 represents the read channels status command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aamm(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

mm are two hexadecimal values. Each value is interpreted as 4 bits. The first 4-bit value is 0. The second 4-bit value represents the status of channels 0-3. A value of 0 means the channel is disabled, while a value of 1 means the channel is enabled.

$aaSi6

(cr) is the terminating character, carriage return (0Dh)

Example command: **$00S16(cr)**

response: **!0001(cr)**

The command asks the analog input module in slot 1 of the system at address 00h to send the status of its input channels. The analog input module responds that channel 0 of its multiplex channels is enabled, the others are disabled (01h equals 0000 and 0001).

$aaSi0

Name RTD Span Calibration

Description Calibrates a specified RTD input module of a specified system to correct for gain errors.

Syntax **$aaSi0(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system which contains the RTD module.

Si identifies the slot i (i:0 to 3) containing the RTD module to be calibrated.

0 represents the span calibration command.

(cr) is the terminating character, carriage return (0Dh).

Response. **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

> delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

$aaSi1

Name RTD Zero Calibration

Description Calibrates a specified RTD input module of a specified system to correct for offset errors.

Syntax **$aaSi1(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system which contains the module which is to be calibrated.

Si identifies the slot i (i:0 to 3) containing the RTD module to be calibrated.

1 represents the zero calibration command.

(cr) is the terminating character, carriage return (0Dh).

Response. **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

$aaSi2

Name RTD Self Calibration

Description Causes a specified RTD input module of a specified system to do a self calibration.

Note: This command is for use when RTD Zero and Span calibration commands have been tried and had no effect. A user first issues an RTD self calibration command, and then issues zero and span calibration commands.

Syntax **$aaSi2(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system which contains the module to be calibrated.

Si identifies the desired slot i (i:0 to 3) containing the module to be calibrated.

2 represents the self calibration command.

(cr) is the terminating character, carriage return (0Dh).

Response. **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

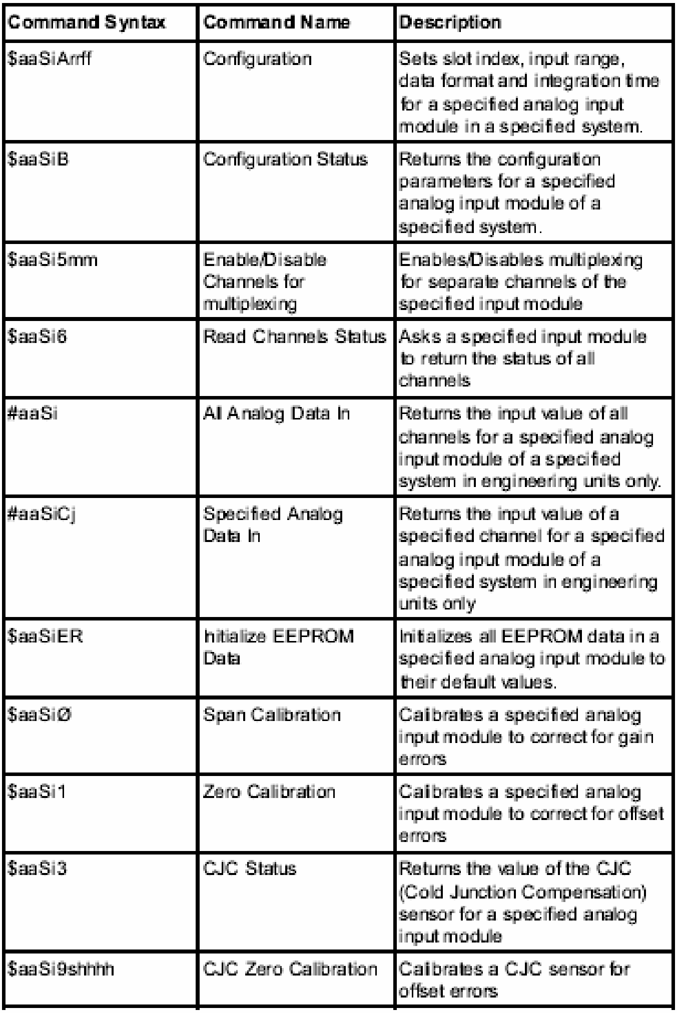
! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

6.3 Analog Input Command Set



Command Syntax	Command Name	Description
$aaSiArrff	Configuration	Sets slot index, input range, data format and integration time for a specified analog input module in a specified system.
$aaSiB	Configuration Status	Returns the configuration parameters for a specified analog input module of a specified system.
$aaSi5mm	Enable/Disable Channels for multiplexing	Enables/Disables multiplexing for separate channels of the specified input module
$aaSi6	Read Channels Status	Asks a specified input module to return the status of all channels
#aaSi	All Analog Data In	Returns the input value of all channels for a specified analog input module of a specified system in engineering units only.
#aaSiCj	Specified Analog Data In	Returns the input value of a specified channel for a specified analog input module of a specified system in engineering units only
$aaSiER	Initialize EEPROM Data	Initializes all EEPROM data in a specified analog input module to their default values.
$aaSiØ	Span Calibration	Calibrates a specified analog input module to correct for gain errors
$aaSi1	Zero Calibration	Calibrates a specified analog input module to correct for offset errors
$aaSi3	CJC Status	Returns the value of the CJC (Cold Junction Compensation) sensor for a specified analog input module
$aaSi9shhhh	CJC Zero Calibration	Calibrates a CJC sensor for offset errors

Note: See pages 71-89 for Analog Input Alarm Command Set.

$aaSiArrff

Name Configuration

Description Sets slot index, input range, data format and integration time for a specified analog input module in a specified system.

Syntax **$aaSiArrff(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to configure.

Si identifies the I/O slot you want to configure.

A is I/O module configuration command.

rr represents the 2-character hexadecimal code of the input range. (See Appendix B)

ff is a hexadecimal number that equals the 8-bit parameter representing data format. Bits 0 and 1 represent data format. Bit 7 represents integration time. The layout of the 8-bit parameter is shown in Figure 6-3. The other bits are not used and are set to 0.

(cr) is the terminating character, carriage return (0Dh)

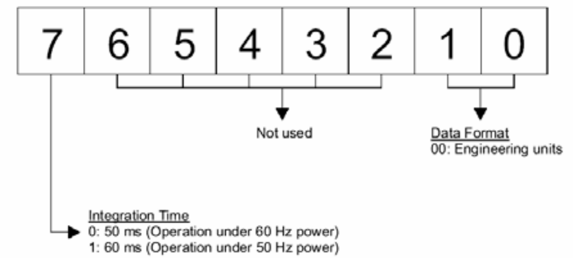

7
6
5
4
3
2
1
0
Not used
Data Format
00: Engineering units
Integration Time
0: 50 ms (Operation under 60 Hz power)
1: 60 ms (Operation under 50 Hz power)

***Figure 6-3** Data format for 8-bit parameters*

$aaSiArrff

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

Example command: **$35S3A0000(cr)**

response: **!35(cr)**

The analog input module in slot 3 of the ADAM-5000 system at address 35h is configured to an input range ±15mV, engineering units data format, and integration time 50ms (60Hz).

The response indicates that the command has been received.

Note: *An analog input module requires a maximum of 7 seconds to perform auto calibration and ranging after it is reconfigured. During this time span, the module cannot be addressed to perform any other actions.*

$aaSiB

Name Configuration Status

Description Returns the configuration status parameters for a specified analog input module of a specified system.

Syntax **$aaSiB(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the I/O slot you want to read.

B is configuration status command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aarrff(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

rr represents the 2-character hexadecimal code of the input range.

ff is a hexadecimal number that equals the 8-bit parameter representing data format. Bit 0 and 1 represent data format. Bit 7 represents integration time. (See Configuration Command $aaSiArrff).

$aaSiB

(cr) is the terminating character, carriage return (0Dh)

Example command: **$26S1B**

response: **!260000**

The ADAM-5018 analog input module in slot 1 of the ADAM-5000 system at address 26h responds with an input range ±15mV, engineering units data format, and integration time 50ms (60Hz).

$aaSi5mm

Name Enable/Disable Channels for multiplexing

Description Enables/Disables multiplexing for separate channels of the specified input module

Syntax **$aaSi5mm(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot of the system.

5 identifies the enable/disable channels command.

mm are two hexadecimal values. Each value is interpreted as 4 bits. The first 4-bit value represents the status of channels 4-7, the second 4 bit value represents the status of channels 0-3. A value of 0 means the channel is disabled, while a value of 1 means the channel is enabled. (See the Read Channel Status Command $aaSi6)

Note: Bit 7 cannot be enabled in the ADAM-5018 / 5018P since the module is physically limited to 7 channels.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

$aaSi5mm

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

Example command: **$00S1581(cr)**

response: **!00(cr)**

The command enables/disables channels of the analog input module in slot 1 of the system at address 00h. Hexadecimal 8 equals binary 1000, which enables channel 7 and disables channels 4, 5 and 6. Hexadecimal 1 equals binary 0001, which enables channel 0 and disables channels 1, 2 and 3.

$aaSi6

Name Read Channels Status

Description Asks a specified input module to return the status of all channels

Syntax **$aaSi6(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the I/O slot of the system you want to read channels status. The channel status defines whether a channel is enabled or disabled.

6 is the read channels status command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aamm(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

mm are two hexadecimal values. Each value is interpreted as 4 bits. The first 4-bit value represents the status of channels 4-7, the second 4 bits represents the status of

$aaSi6

channels 0-3. A value of 0 means the channel is disabled, while a value of 1 means the channel is
enabled.

(cr) is the terminating character, carriage return (0Dh)

Example command: **$02S16(cr)**

response: **!02FF(cr)**

The command asks the analog input module in slot 1 of the system at address 02h to send the status of its input channels. The analog input module responds that all its multiplex channels are enable (FF equals 1111 and 1111).

#aaSi

Name All Analog Data In

Description Returns the input value of all channels for a specified analog input module of a specified system in engineering unit only.

Syntax **#aaSi(cr)**

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si is the I/O slot of ADAM-5000 system you want to read.

(cr) is the terminating character, carriage return (0Dh)

Response **>(data) (data) (data) (data) (data) (data) (data) (data) (cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

> is a delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

(data) is the input value in engineering units of a channel in the interrogated module of the specified system. The (data) from all channels is shown in sequence from 7 to 0. If (data) = " ", it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh)

$aaSi

Example command: **#12S1(cr)**

response: **+1.4567 +1.4852 +1.4675 +1.4325 +1.4889 +1.4235 +1.4787 +1.4625(cr)**

The command requests the analog input module in slot 1 of the ADAM-5000 system at address 12h to return the input values of all channels.

The analog input module responds that input values of all channels are in sequence from 7 to 0: +1.4567, +1.4852, +1.4675, +1.4325, +1.4889, +1.4235, +1.4787 and +1.4625.

#aaSiCj

Name Specified Analog Data In

Description Returns the input value of a specified channels for a specified analog input module of a specified system in engineering unit only.

Syntax **#aaSiCj(cr)**

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the I/O slot you want to interrogate.

Cj identifies the channel you want to read.

(cr) is the terminating character, carriage return (0Dh)

Response **>(data)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

> is a delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

(data) is the input value in engineering units of the specified channel for a specified analog input module of the specified system. If (data) = “   ”, it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh)

#aaSiCj

Example command: **#22S2C2(cr)**

response: **>+1.4567**

The command requests the analog input module in slot 2 of the ADAM-5000 system at address 22h to return the input value of channel 2.

The analog input module responds that the input value of channel 2 is +1.4567.

$aaSiER

Name Initialize EEPROM data

Description Initializes all EEPROM data in a specified analog input module to their default values. This command is sent following a failed attempt to calibrate a module (the module shows no effect from an attempted calibration). Following initialization, the problem module should readily accept calibration.

Syntax **$aaSiER(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot for which you wish to initialize all EEPROM data.

ER is Initialize all EEPROM data command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiØ

Name Span Calibration

Description Calibrates a specified analog input module to correct for gain errors

Syntax **$aaSiØ(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system which is to be calibrated.

Si identifies the I/O slot which is to be calibrated.

Ø represents the span calibration command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

Note: In order to successfully calibrate an analog input module's input range, a proper calibration input signal should be connected to the analog input module before and during the calibration process. (See also Chapter 4, Section 4.5 on Calibration)

$aaSi1

Name Zero Calibration

Description Calibrates a specified analog input module to correct for offset errors

Syntax **$aaSi1(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system which is to be calibrated.

Si identifies the I/O slot which is to be calibrated.

1 represents the zero calibration command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

Note: In order to successfully calibrate an analog input module's input range, a proper calibration input signal should be connected to the analog input module before and during the calibration process. (See also Chapter 4, Section 4.5 on Calibration)

$aaSi3

Name	CJC Status Command (ADAM-5018 / 5018P only)
Description	Returns the value of the CJC (Cold Junction Compensation) sensor for a specified analog input module
Syntax	**$aaSi3(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot which contains the CJC Status you wish to retrieve.

3 is CJC Status command.

(cr) is the terminating character, carriage return (0Dh)

Response >**(data)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

>delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(data) is the value that is retrieved by the module by reading its CJC sensor. The data format, in degrees Celsius, consists of a “+” or “-” sign followed by five decimal digits and a fixed decimal point. The resolution of the data is 0.1°C.

(cr) is the terminating character, carriage return (0Dh)

$aaSi3

Example command: **$09S13(cr)**

response: **>+0036.8(cr)**

The command requests the analog input module in slot 1 of the ADAM-5000 system at address 09h to read its CJC sensor and return the data. The analog input module responds with 36.8°C.

$aaSi9shhh

Name CJC Zero Calibration (ADAM-5018 / 5018P only)

Description Calibrates an analog input module to adjust for offset errors of its CJC (Cold Junction Compensation) sensor

Syntax **$aaSi9shhhh(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot which contains the CJC Status you wish to retrieve.

9 is CJC Status command.

s sign, + or -, indicates whether to increase or decrease the CJC offset value.

hhhh is a four character hexadecimal "count" value. Each count equals approximately 0.009°C. The value can range from 0000 to FFFF.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSi9shhhh

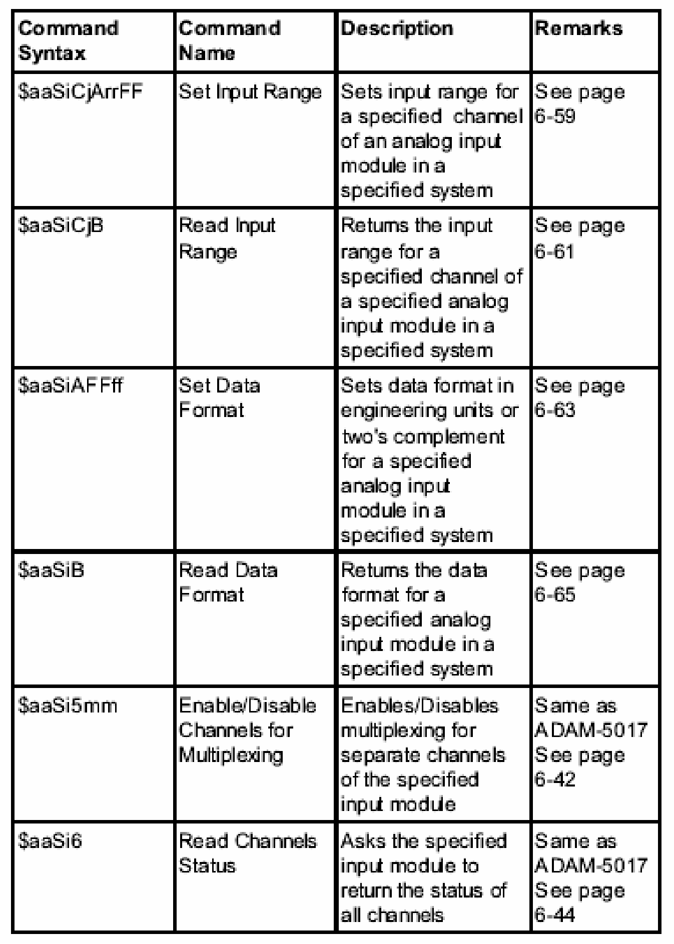
Example command: **$07S29+0042(cr)**

response: **!07(cr)**

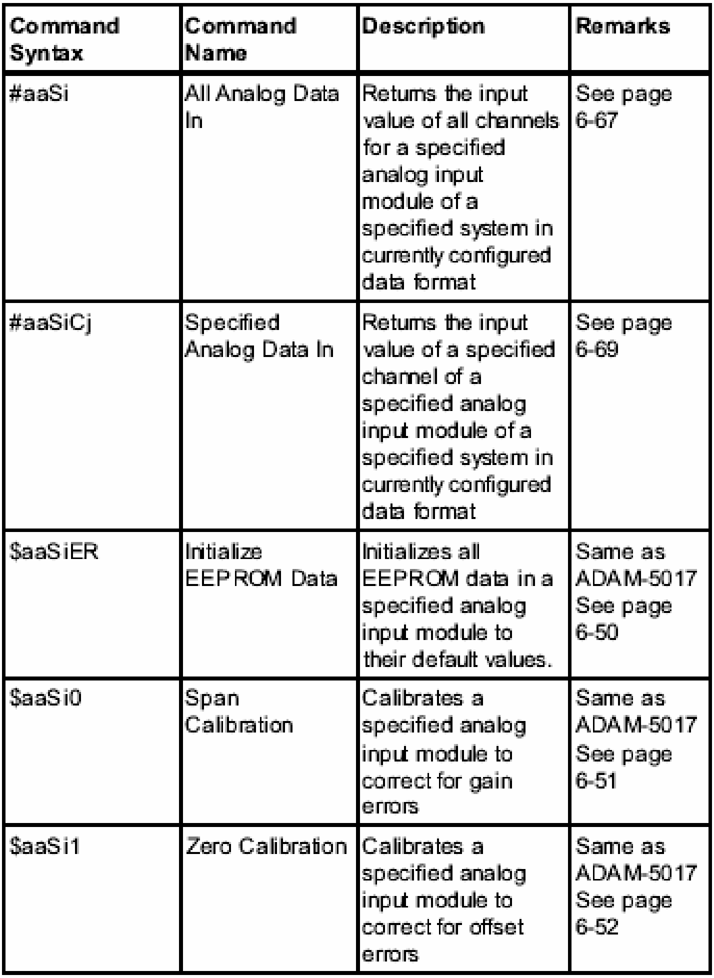
The command increases the CJC offset value of the analog input module in slot 2 of the system at address 07h with 66 counts (42 hex) which equals about 0.6°C.

Note: *An analog input module requires a maximum of 2 seconds to perform auto calibration and ranging after it receives a CJC Calibration command. During this interval, the module cannot be addressed to perform any other actions.*

6.6 ADAM-5017H/ADAM-5017UH Analog Input Command Set



Command Syntax	Command Name	Description	Remarks
$aaSiCjArrFF	Set Input Range	Sets input range for a specified channel of an analog input module in a specified system	See page 6-59
$aaSiCjB	Read Input Range	Returns the input range for a specified channel of a specified analog input module in a specified system	See page 6-61
$aaSiAFFff	Set Data Format	Sets data format in engineering units or two's complement for a specified analog input module in a specified system	See page 6-63
$aaSiB	Read Data Format	Returns the data format for a specified analog input module in a specified system	See page 6-65
$aaSi5mm	Enable/Disable Channels for Multiplexing	Enables/Disables multiplexing for separate channels of the specified input module	Same as ADAM-5017 See page 6-42
$aaSi6	Read Channels Status	Asks the specified input module to return the status of all channels	Same as ADAM-5017 See page 6-44



Command Syntax	Command Name	Description	Remarks
#aaSi	All Analog Data In	Returns the input value of all channels for a specified analog input module of a specified system in currently configured data format	See page 6-67
#aaSiCj	Specified Analog Data In	Returns the input value of a specified channel of a specified analog input module of a specified system in currently configured data format	See page 6-69
$aaSiER	Initialize EEPROM Data	Initializes all EEPROM data in a specified analog input module to their default values.	Same as ADAM-5017 See page 6-50
$aaSi0	Span Calibration	Calibrates a specified analog input module to correct for gain errors	Same as ADAM-5017 See page 6-51
$aaSi1	Zero Calibration	Calibrates a specified analog input module to correct for offset errors	Same as ADAM-5017 See page 6-52

Note: The ADAM-5017H / 5017UH module also has "Alarm Setting" functions. The alarm command set for the ADAM-5017UH is the same as that for the ADAM-5013, ADAM-5017, ADAM-5017H and ADAM-5018. Please refer to pages6-71 to 6-89 for this set of commands.

$aaSiCjrrFF

Name Set Input Range

Description Sets the input range for a specified channel of a specified analog input module in a specified system.

Syntax **$aaSiCjArrFF**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to configure.

SiCj identifies the slot i (i:0 to 3) of the ADAM-5000 system and the channel j (j:0 to 7) of the ADAM-5017H / 5017UH whose range you want to set.

A represents the set input range command.

rr represents the 2-character hexadecimal code of the input range. (See Appendix B)

Note: Each channel in a ADAM-5017H / 5017UH module may be set to a different range, but the data formats of all channels in this module must be the same.

(cr) is the terminating character, carriage return (0Dh).

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal

$aaSiCjrrFF

address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

Example command: **$35S3C1A0bFF(cr)**

response: **!35(cr)**

Channel 1 of the ADAM-5017H / 5017UH module in slot 3 of the ADAM-5000 system at address 35h is set to the input range 0-20 mA, engineering unit data format. The response indicates that the command has been received as a valid command.

$aaSiCjB

Name Read Input Range

Description Returns the input range in engineering units for a specified channel of a specified analog input module in a specified system.

Syntax **$aaSiCjB**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

SiCj identifies the slot i (i:0 to 3) of the ADAM-5000 system and the channel j (j:0 to 7) of the ADAM-5017H / 5017UH module you want to interrogate.

B represents the read input range command.

(cr) is the terminating character, carriage return (0Dh).

Response **!aarr00(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

rr represents the 2-character hexadecimal code of the input range. (See Appendix B)

(cr) is the terminating character, carriage return (0Dh).

$aaSiCjB

Example command: **$35S3C1B(cr)**

response: **!350b00(cr)**

Channel 1 of the ADAM-5017H / 5017UH module in slot 3 of the ADAM-5000 system at address 35h responds with an input range 0-20 mA, engineering unit data format.

$aaSiAFFff

Name Set Data Format

Description Sets the data format in engineering units or in two's complement format for a specified analog input module in a specified system.

Syntax **$aaSiAFFff**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to configure.

Si identifies the I/O slot of the ADAM-5000 system containing the ADAM-5017H module you want to configure.

AFF represents the set data format command.

ff represents the 2-character hexadecimal code of the data format. 00 is for engineering unit format. 02 is for two’s complement format.

Note: Each channel in an ADAM-5017H / 5017UH module may be set to a different range, but the data formats of all channels in this module must be the same.

(cr) is the terminating character, carriage return (0Dh).

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was

$aaSiAFFff

invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

Example command: **$35S3AFF00(cr)**

response: **!35(cr)**

The data format of the ADAM-5017H / 5017UH module in slot 3 of the ADAM-5000 system at address 35h is configured for engineering unit format. The response indicates that the command has been received as a valid command.

$aaSiB

Name Read Data Format

Description Returns the data format for a specified analog input module in a specified system.

Syntax **$aaSiB**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the I/O slot of the ADAM-5000 system containing the ADAM-5017H / 5017UH module you want to interrogate.

B represents the read data format command.

(cr) is the terminating character, carriage return (0Dh).

Response **!aaFFff(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

ff represents the 2-character hexadecimal code of the data format. 00 is for engineering unit format. 02 is for two's complement format.

(cr) is the terminating character, carriage return (0Dh).

$aaSiB

Example command: **$35S3B(cr)**

response: **!35FF00(cr)**

The ADAM-5017H / 5017UH module in slot 3 of the ADAM-5000 system at address 35h responds that it is configured for engineering unit data format.

#aaSi

Name All Analog Data In

Description Returns the input value of all channels for a specified analog input module of a specified system in engineering units or two’s complement data format

Syntax **#aaSi**

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the I/O slot (i:0 to 3) of ADAM-5000 system you want to read.

(cr) is the terminating character, carriage return (0Dh).

Response **!(data)(data)(data)(data) (data)(data)(data)(data)(cr)** if the command is valid. (Engineering Unit Data Format)

!(dddd)(dddd)(dddd)(dddd)(dddd)(dddd)(dddd)(dddd)(cr) if the command is valid. (Two’s Complement Data Format)

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

(data) is the input value in engineering units of the interrogated module of the specified system. The (data) from all channels is shown in sequence from 7 to 0. If (data)=” “, it means the channel is invalid.

(dddd) is the input value in two’s complement format of

#aaSi

the interrogated module of the specified system. The (dddd) from all channels is shown in sequence from 7 to 0. If (dddd)=” “, it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh).

Example command: **#35S3(cr)**

response: **+6.000 +7.000 +8.125 +4.250 +10.000 +8.500 +7.675 +5.445 (cr)**

The command requests the ADAM-5017H / 5017UH module in slot 3 of the ADAM-5000 system at address 35h to return the input values of all channels.

The analog input module responds with the input values of all channels, in sequence from 0 to 7: **+6.000, +7.000, +8.125, +4.250, +10.000, +8.500, +7.675, +5.445.**

#aaSiCj

Name Specified Analog Data In

Description Returns the input value of a specified channel of a specified analog input module in a specified ADAM-5000 system in engineering units or two’s complement data format

Syntax **#aaSiCj(cr)**

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to configure.

Si identifies the I/O slot (i:0 to 3) of ADAM-5000 system you want to read.

Cj identifies the channel you want to read.

(cr) is the terminating character, carriage return (0Dh).

Response **!(data)(cr)** if the command is valid. (Engineering Unit Data Format)

!(dddd)(cr) if the command is valid. (Two’s Complement Data Format)

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

(data) is the input value in engineering units of the specified channel of the specified analog input module.

#aaSiCj

If (data)=" ", it means the channel is invalid.

(dddd) is the input value in two's complement format of the specified channel of the specified module. If (dddd)=" ", it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh).

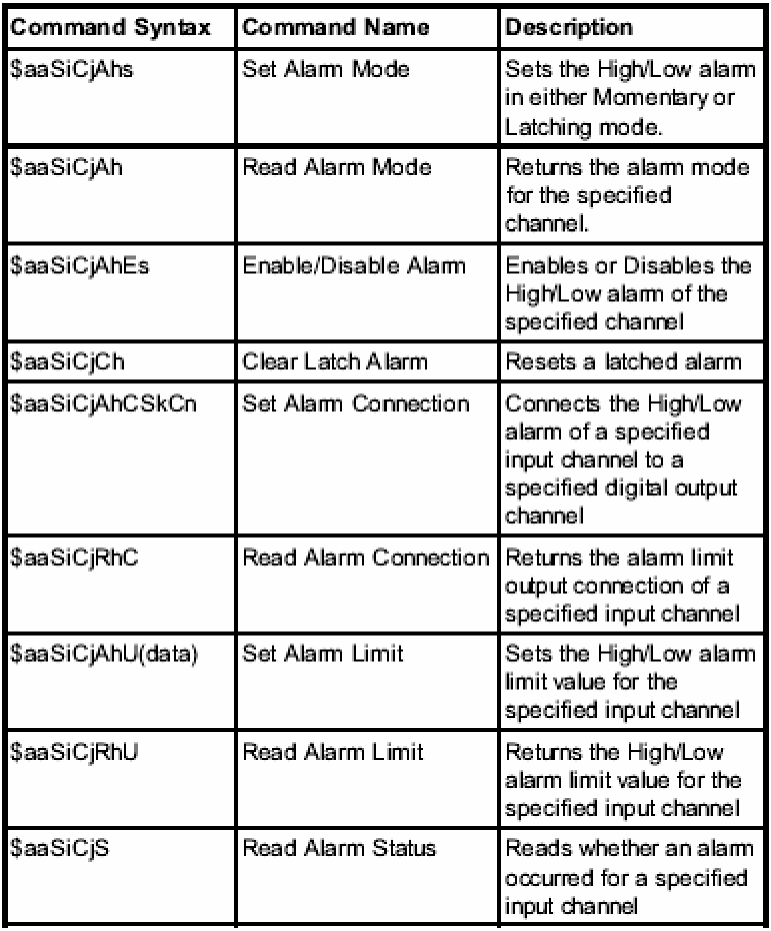
Example command: **#35S3C2(cr)**

response: **+9.750 (cr)**

The command requests the ADAM-5017H / 5017UH module in slot 3 of the ADAM-5000 system at address 35h to return the input value of channel 2.

The analog input module responds that the input value of channel 2 is **+9.750.**

6.7 Analog Input Alarm Command Set



Command Syntax	Command Name	Description
$aaSiCjAhs	Set Alarm Mode	Sets the High/Low alarm in either Momentary or Latching mode.
$aaSiCjAh	Read Alarm Mode	Returns the alarm mode for the specified channel.
$aaSiCjAhEs	Enable/Disable Alarm	Enables or Disables the High/Low alarm of the specified channel
$aaSiCjCh	Clear Latch Alarm	Resets a latched alarm
$aaSiCjAhCSkCn	Set Alarm Connection	Connects the High/Low alarm of a specified input channel to a specified digital output channel
$aaSiCjRhC	Read Alarm Connection	Returns the alarm limit output connection of a specified input channel
$aaSiCjAhU(data)	Set Alarm Limit	Sets the High/Low alarm limit value for the specified input channel
$aaSiCjRhU	Read Alarm Limit	Returns the High/Low alarm limit value for the specified input channel
$aaSiCjS	Read Alarm Status	Reads whether an alarm occurred for a specified input channel

Note: *This command set applies to the ADAM-5013, ADAM-5017, ADAM-5017H , ADAM5017UH , ADAM-5018P and the ADAM-5018P.*

$aaSiCjAhs

Name Set Alarm Mode

Description Sets the High/Low alarm of the specified input channel in the addressed ADAM-5000 system to either Latching or Momentary mode.

Syntax **$aaSiCjAhs(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of anADAM-5000 system.

SiCj identifies the desired slot i (i : 0 to 3) and the desired channel j (j : 0 to 7).

Ahs is the Set Alarm Mode command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

s indicates alarm mode and can have the value M = Momentary mode, L = Latching mode

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjAhs

Example command: **$03S0C1AHL(cr)**

response: **!03(cr)**

Channel 1 of slot 0 in the ADAM-5000 system at address 03h is instructed to set its High alarm in Latching mode.

The module confirms that the command has been received.

$aaSiCjAh

Name Read Alarm Mode

Description Returns the alarm mode for the specified channel in the specified ADAM-5000 system.

Syntax **$aaSiCjAh(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i (i : 0 to 3) and the desired channel j (j : 0 to 7).

Ah is the Read Alarm Mode command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aas(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

s indicates alarm mode and can have the value M = Momentary mode, L = Latching mode

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjAh

Example command: **$03S0C1AL(cr)**

response: **!03M(cr)**

Channel 1 of slot 0 in the ADAM-5000 system at address 03h is instructed to return its Low alarm mode. The system responds that it is in Momentary mode.

$aaSiCjAhEs

Name Enable/Disable Alarm

Description Enables/Disables the High/Low alarm of the specified input channel in the addressed ADAM-5000 system

Syntax **$aaSiCjAhEs(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i (i : 0 to 3) and the desired channel j (j : 0 to 7).

AhEs is the Set Alarm Mode command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

s indicates alarm enable/disable and can have the value E = Enable, D = Disable

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjAhEs

Example command: **$03S0C1ALEE(cr)**

response: **!03(cr)**

Channel 1 of slot 0 in the ADAM-5000 system at address 03h is instructed to enable its Low alarm function.

The module confirms that its Low alarm function has been enabled.

Note: An analog input module requires a maximum of 2 seconds after it receives an Enable/Disable Alarm command to let the setting take effect. During this interval, the module cannot be addressed to perform any other actions.

$aaSiCjCh

Name Clear Latch Alarm

Description Sets the High/Low alarm to OFF (no alarm) for the specified input channel in the addressed ADAM-5000 system

Syntax **$aaSiCjCh(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i (i : 0 to 3) and the desired channel j (j : 0 to 7).

Ch is the Clear Latch Alarm command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjCh

Example command: **$03S0C1CL(cr)**

response: **!03(cr)**

Channel 1 of slot 0 in the ADAM-5000 system at address 03h is instructed to set its Low alarm state to OFF.

The system confirms it has done so accordingly.

$aaSiCjAhSkCn

Name Set Alarm Connection

Description Connects the High/Low alarm of the specified input channel to the specified digital output in the addressed ADAM-5000 system

Syntax **$aaSiCjAhCSkCn(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i (i : 0 to 3) and the desired analog input channel j (j : 0 to 7).

AhC is the Set Alarm Connection command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

SkCn identifies the desired slot k (k : 0 to 3) and the desired digital output point n (n : 0 to F). To disconnect the digital output, k and n should be set as '*'.
(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjAhCSkCn

Example command: **$03S0C1ALCS1C0(cr)**

response: **!03(cr)**

Channel 1 of slot 0 in the ADAM-5000 system at address 03h is instructed to connect its Low alarm to the digital output of point 0 of slot 1 in the same ADAM-5000 system.

The system confirms it has done so accordingly.

$aaSiCjRhc

Name Read Alarm Connection

Description Returns the High/Low alarm limit output connection of a specified input channel in the addressed ADAM-5000 system

Syntax **$aaSiCjRhC(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i (i : 0 to 3) and the desired analog input channel j (j : 0 to 7).

RhC is the Read Alarm Connection command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aaSkCn(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

SkCn identifies the desired slot k (k : 0 to 3) and the desired digital output point n (n : 0 to F) to which the input alarm is connected. If the values of k and n are ‘*’, the analog input has no connection with a digital output point.

$aaSiCjRhC

(cr) represents terminating character, carriage return (0Dh)

Example command: **$03S0C1RLC(cr)**

response: **!03S1C0(cr)**

Channel 1 of slot 0 in the ADAM-5000 system at address 03h is instructed to read its Low alarm output connection.

The system responds that the Low alarm output connects to the digital output at point 0 of slot 1 in the same ADAM-5000 system.

$aaSiCjAhU(data)

Name Set Alarm Limit

Description Sets the High/Low alarm limit value for the specified input channel of a specified ADAM-5000 system.

Syntax **$aaSiCjAhU(data)(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i (i : 0 to 3) and the desired analog input channel j (j : 0 to 7).

AhU is the Set Alarm Limit command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(data) represents the desired alarm limit setting. The format is always in engineering units.

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjAhU(data)

Example command: **$03S0C1AHU+080.00(cr)**

response: **!03(cr)**

Channel 1 of slot 0 in the ADAM-5000 system at address 03h is configured to accept type-T thermocouple input. The command will set its High alarm limit to +80°C.

The system confirms the command has been received.

Note: An analog input module requires a maximum of 2 seconds after it receives a Set Alarm Limit command to let the settings take effect. During this interval, the module cannot be addressed to perform any other actions.

$aaSiCjRhU

Name Read Alarm Limit

Description Returns the High/Low alarm limit value for the specified input channel in the addressed ADAM-5000 system

Syntax **$aaSiCjRhU(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i (i : 0 to 3) and the desired analog input channel j (j : 0 to 7).

RhU is the Read Alarm Limit command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(data)(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(data) represents the desired alarm limit setting. The format is always in engineering units.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjRhU

Example command: **$03S0C1RHU(cr)**

response: **!03+2.0500(cr)**

Channel 1 of slot 0 in the ADAM-5000 system at address 03h is configured to accept 5V input. The command instructs the system to return the High alarm limit value for that channel.

The system responds that the High alarm limit value in the desired channel is 2.0500 V.

$aaSiCjS

Name Read Alarm Status

Description Reads whether an alarm occurred for the specified input channel in the specified ADAM-5000 system

Syntax **$aaSiCjS(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i (i : 0 to 3) and the desired analog input channel j (j : 0 to 7).

S is the Read Alarm Status command.

(cr) represents terminating character, carriage return (0Dh)

Response **!aahl(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

h represents the status of High alarm. ‘1’ means the High alarm occurred, ‘0’ means it did not occur.

l represents the status of Low alarm. ‘1’ means the Low alarm occurred, ‘0’ means it did not occur.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjS

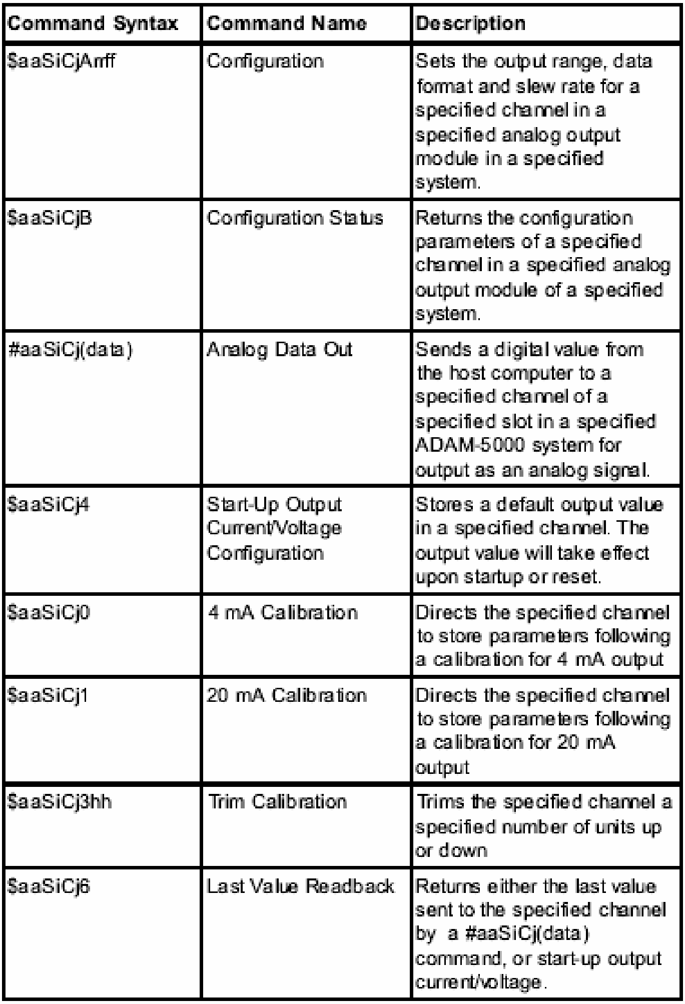
Example command: **$03S0C1S(cr)**

response: **!0301(cr)**

The command instructs the system at address 03h to return its alarm status for channel 1 of slot 0.

The system responds that a High alarm has not occurred and that a Low alarm has occurred.

6.8 Analog Output Command Set



Command Syntax	Command Name	Description
$aaSiCjArrff	Configuration	Sets the output range, data format and slew rate for a specified channel in a specified analog output module in a specified system.
$aaSiCjB	Configuration Status	Returns the configuration parameters of a specified channel in a specified analog output module of a specified system.
#aaSiCj(data)	Analog Data Out	Sends a digital value from the host computer to a specified channel of a specified slot in a specified ADAM-5000 system for output as an analog signal.
$aaSiCj4	Start-Up Output Current/Voltage Configuration	Stores a default output value in a specified channel. The output value will take effect upon startup or reset.
$aaSiCj0	4 mA Calibration	Directs the specified channel to store parameters following a calibration for 4 mA output
$aaSiCj1	20 mA Calibration	Directs the specified channel to store parameters following a calibration for 20 mA output
$aaSiCj3hh	Trim Calibration	Trims the specified channel a specified number of units up or down
$aaSiCj6	Last Value Readback	Returns either the last value sent to the specified channel by a #aaSiCj(data) command, or start-up output current/voltage.

$aaSiCjArrff

Name Configuration

Description Sets the output range, data format and slew rate for a specified channel of a specified analog output module in a specified system.

Syntax **$aaSiCjArrff(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to configure.

SiCj identifies the I/O slot i (i : 0 to 3) and the channel j (j : 0 to 3) of the module you want to configure.

A is I/O module configuration command.

rr represents the 2-character hexadecimal code of the output range. (See Appendix B)

ff is a hexadecimal number that equals the 8-bit parameter representing the status of data format and slew rate. Bits 0 and 1 represent data format. Bits 2,3,4,5 represent slew rate. The layout of the 8-bit parameter is shown in Figure 6-4. The other bits are not used and are set to 0.

(cr) is the terminating character, carriage return (0Dh)

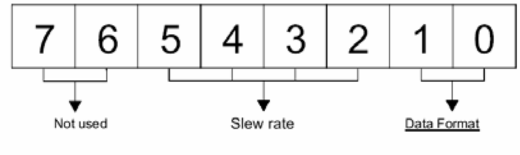


Figure 6-4 *Data format for 8-bit parameters*

$aaSiCjArrff

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

Example command: **$35S3C0A3110(cr)**

response: **!35(cr)**

The analog output channel 0 in slot 3 of the ADAM-5000 system at address 35h is configured to an output range 4 to 20mA, engineering units data format, and a slew rate of 1.0mA/sec.

The response indicates that the command has been received.

Note: *An analog output module requires a maximum of 20 milliseconds to perform auto calibration and ranging after it is reconfigured. During this time span, the module cannot be address to perform any other actions.*

$aaSiCjB

Name Configuration Status

Description Returns the configuration parameters of a specified channel in a specified analog output module of a specified system.

Syntax **$aaSiCjB(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

SiCj identifies the I/O slot i (i : 0 to 3) and the channel j (j : 0 to 3) you want to read.

B is configuration status command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aarrff(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

rr represents the 2-character hexadecimal code of the output range.

ff is a hexadecimal number that equals the 8-bit parameter representing the status of data format and slew rate.

$aaSiCjB

Bits 0 and 1 represent data format. Bits 2, 3, 4 and 5 represent slew rate. The other bits are not used and are set to 0. (See Configuration command $aaSiCjArrff)

(cr) is the terminating character, carriage return (0Dh)

Example command: **$24S1C1B**

response: **!243210**

The analog output channel 1 in slot 1 of the ADAM-5000 system at address 24h responds with an output range 0 to 10V, engineering units data format, and a slew rate of 1.0mA/sec.

$aaSiCj(data)

Name Analog Data Out

Description Sends a digital value from the host computer to a specified channel of a specified slot in a specified ADAM-5000 system for output as an analog signal. Upon receipt, the analog output module in the specified slot will output an analog signal corresponding to the digital value received.

Syntax **#aaSiCj(data)(cr)**

# is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i (i : 0 to 3) and the channel j (j : 0 to 3) of the analog output module that is to output an analog signal.

(data) is a digital value incoming to the module, which corresponds to the desired analog output value (always in engineering units) to be output from the module. The analog value output will depend on the module's range configuration. (See also Appendix B, Data Formats and I/O Ranges)

(cr) is the terminating character, carriage return (0Dh)

Response **>(cr)** if the command is valid.

?aa(cr) if a value was sent that is out of range. Note that when the analog output module receives such a value, it will try to use a value that is close to the one received, but within the module‘s configured range.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

> is a delimiter character indicating a valid command was received.

#aaSiCj(data)

? delimiter character indicating the command was invalid.

(cr) is the terminating character, carriage return (0Dh)

Example command: **#33S1C115.000(cr)**

response: **>(cr)**

The command instructs the module in slot 1 of the ADAM-5000 system at address 33h to output a value of 15 mA from it's channel 1. The module should be an analog output module with it's channel 1 configured for a range of 0-20 mA or 4-20 mA. If it is an analog output module configured for the range 0-10 V, it's output value will be 10 V and the response will be **?33(cr)**.

$aaSiCj4

Name Start-Up Output Current/Voltage Configuration

Description Stores a default output value in a specified channel. The output value will take effect upon startup or reset.

Syntax **$aaSiCj4(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i (i : 0 to 3) and the channel j (j : 0 to 3) of the module you want to set.

4 is the Start-Up Output Current/Voltage Configuration command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCj4

Example command: **$0AS1C14(cr)**

response: **!0A(cr)**

Presume the present output value of channel 1 of slot 1 in the ADAM-5000 system at address 0Ah is 9.4 mA. The command tells the analog output module to store the present output value in its non-volatile memory. When the system is powered up or reset, its default output value will be 9.4 mA.

The response from the ADAM-5000 system at address 0Ah indicates the command has been received.

Note: *An analog output module requires a maximum of 6 milliseconds after it receives a Startup Output Current/Voltage Configuration command to let the settings take effect. During this interval, the module cannot be addressed to perform any other actions.*

$aaSiCj0

Name 4 mA Calibration

Description Directs the specified channel to store parameters following a calibration for 4 mA output

Syntax **$aaSiCj0(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i (i : 0 to 3) and the channel j (j : 0 to 3) of the module you want to calibrate.

0 is the 4 mA calibration command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

Note: Before issuing the 4 mA Calibration command, the analog output module should be trimmed to the correct value using the Trim Calibration command. Either a mA meter or a resistor and voltmeter should

$aaSiCj0

be connected to the module's output. (See also the analog output module's Trim Calibration command in Chapter 4, Section 4.5, Analog Output Module Calibration for a detailed description.)

$aaSiCj1

Name 20 mA Calibration

Description Directs the specified channel to store parameters following a calibration for 20 mA output

Syntax **$aaSiCj1(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i (i : 0 to 3) and the channel j (j : 0 to 3) of the module you want to calibrate.

1 is the 20 mA calibration command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

Note: Before issuing the 20 mA Calibration command, the analog output module should be trimmed to the correct value using the Trim Calibration command. Either a mA meter or a resistor and voltmeter should

$aaSiCj1

be connected to the module's output. (See also the analog output module's Trim Calibration command in Chapter 4, Section 4.5, Analog Output Module Calibration for a detailed description.)

$aaSiCj3hh

Name Trim Calibration

Description Trims the specified channel a specified number of units up or down

Syntax **$aaSiCj3hh(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i (i : 0 to 3) and the channel j (j : 0 to 3) of the module you want to calibrate.

3 is the trim calibration command.

hh is the 2-character twos complement hexadecimal value that represents the number of counts by which to increase or decrease the output current. Each count equals approximately 1.5µA. Values range from 00 to 5F and from A1 to FF (hexadecimal), where 00 represents 0 counts, 5F represents +95 counts, A1 represents -95 counts and FF represents -1 counts. Negative values decrease and positive numbers increase the output current according to the number of counts.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

$aaSiCj3hh

(cr) is the terminating character, carriage return (0Dh)

Example command: **$07S1C2314(cr)**

response: **!07(cr)**

The command tells channel 2 of the analog output module in slot 1 of the ADAM-5000 system at address 07h to increase its output value by 20 (14h) counts which is approximately 30 µA.

The analog output module confirms the increase.

Note: *In order to perform a Trim Calibration, either a mA meter or a resistor and voltmeter should be connected to the module's output prior to calibration. (See also the 4 mA and 20 mA Calibration commands of the analog output module's command set. Refer also to Chapter 4, Section 4.5, Analog Output Calibration for a detailed description.)*

$aaSiCj6

Name Last Value Readback

Description Returns either the last value sent to the specified channel by a #aaSiCj(data) command, or the start-up output current/voltage.

Syntax **$aaSiCj6(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i (i : 0 to 3) and the channel j (j : 0 to 3) for the module you want to return a prior value.

6 is the last value readback command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(data)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(data) is the value that is returned by the analog output module. The format of the data depends on the module‘s configurated data format.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCj6

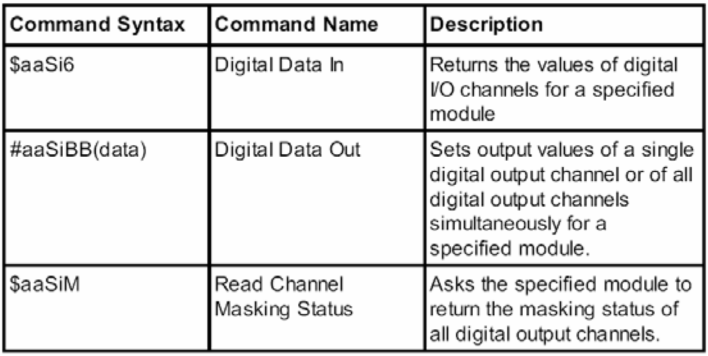
Example command: **$0AS2C16(cr)**

response: **!0A03.000(cr)**

The command tells channel 1 of the analog output module in slot 2 of the ADAM-5000 system at address 0Ah to return the last output value it received from an Analog Data Out command, or its start-up output current /voltage.

The analog output module returns the value 3.000 mA (this assumes that the module was configured for the range 0-20 mA).

6.9 Digital Input/Output



Command Syntax	Command Name	Description
$aaSi6	Digital Data In	Returns the values of digital I/O channels for a specified module
#aaSiBB(data)	Digital Data Out	Sets output values of a single digital output channel or of all digital output channels simultaneously for a specified module.
$aaSiM	Read Channel Masking Status	Asks the specified module to return the masking status of all digital output channels.

$aaSi6

Name Digital Data In

Description **This** command requests that the specified module in an ADAM-5000 system at address **aa** return the status of its digital input channels and a read back value of its digital output channels.

Syntax **$aaSi6(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot of the system you want to read.

6 is the Digital Data In command.

(cr) is the terminating character, carriage return (0Dh)
Response **!aa(datainput)(datainput)00(cr)** if the command is valid.

(ADAM-5051)

!aa(datainput)(dataoutput)00(cr) if the command is valid.

(ADAM-5055S)

!aa(dataoutput)(dataoutput)00(cr) if the command is valid.

(ADAM-5056)

!aa(dataoutput)0000(cr) if the command is valid.

(ADAM-5060, ADAM-5068, ADAM-5069)

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communi cation error or if the specified address does not exist.

Note: ! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(datainput) a 2-character hexadecimal value representing the input values of the digital input module.

(dataoutput) a 2-character hexadecimal value which is the read back of a digital output channel or relay.

(cr) is the terminating character, carriage return (0Dh)

Example command: **$33S26(cr)**

response: **!33112200(cr)**

The command asks the digital input module in slot 2 of the ADAM-5000 system at address 33h to return the values of all of its channels.

The first 2-character portion of the response indicates the address of the ADAM-5000 system.

The second 2-character (high byte) portion of the response, value 11h (00010001), indicates that digital input channels 8 and 12 are ON, channels 9, 10, 11, 13, 14 and 15 are OFF.

The third 2-character (low byte) portion of the response, value 22h (00100010), indicates that digital input channels 1 and 5 are ON, and channels 0, 2, 3, 4, 6 and 7 are OFF.

$aaSiBB(data)

Name Digital Data Out

Description This command either sets a single digital output channel or sets all digital output channels

Syntax #aaSiBB(data)(cr)

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the slot (i:0 to 7) of the ADAM-5000 system which contains the module whose output values you want to set.

BB is used to indicate which channel(s) either single or all will be set.

- Writing to all channels (write a byte): both characters should be equal to zero (**BB=00**).

- Writing to a single channel (write a bit): first character is 1, second character indicates channel number which can range from 0h to Fh. The ADAM-5056 can range from 0h to Fh, and the ADAM-5055S/5060/5068/5069 can range from 0h to 7h).

(data) is the hexadecimal representation of the digital output value(s).

- When writing to a single channel (bit), the first character is always 0. The value of the second character is either 0 or 1.

- When writing to all channels (byte), the 2 or 4-characters are significant. The digital equivalent of these hexadecimal characters represent the channels' status. Note that the numbers of channels on the ADAM-5056 and ADAM-5055S/5060/5068/5069 differ.

- A 4-character hexadecimal value is used to set the channels, from 15 thru 0, of the ADAM-5056.

- A 2-character hexadecimal value is used to set the channels, from 5 thru 0, of the ADAM-5060. Bits 6 and 7 always default to 0 in the ADAM-5060.

- A 2 character hexadecimal value is used to set the channels, from 7 thru 0, of the ADAM-5055S/5068/5069.

Response **>(cr)** if the command was valid.

?aa(cr) if an invalid command has been issued.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

> delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system that is responding.

(cr) is the terminating character, carriage return (0Dh)

Example command: **#15S11201(cr)**

response: **>(cr)**

An output bit with value 1 is sent to channel 2 of a digital output module in slot 1 of the ADAM-5000 system at address 15h - either ADAM-5056 or ADAM-5055S/5060/5068/5069. Channel 2 of the digital output module is set to ON.

command: **#14S1001234(cr)**

response: **>(cr)**

An output byte with value 1234h (0001001000110100) is sent to the digital output module (ADAM-5056) in slot 1 of the ADAM-5000 system at address 14h. Channels 2, 4, 5, 9 and 12 will be set to ON, and all other channels are set to OFF.

command: **#15S0003A(cr)**

response: **>(cr)**

An output byte with value 3Ah (00111011) is send to the digital output module (ADAM-5060) in slot 0 of the ADAM-5000 system at address 15h. Channels 0, 1, 3, 4 and 5 will be set to ON while channel 2 is set to OFF. Bits 6 and 7 are not used and always default to 0.

Note: *If any channel of the digital output module is configured as the output for an analog input alarm, it cannot be reconfigured via digital output commands. Channels used for analog input alarms always have a higher priority.*

$aaSiM

Name Read Channel Masking Status

Description Ask the specified module to return the masking status of digital output channels

Syntax $aaSiM(cr)

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot of the system you want to read.

M is Channel Masking Status command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(data)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system that is responding.

(data) is the hexadecimal value representing the status of all digital output channels. A value of 1 means the channel is masked, while a value of 0 means the channel is valid.

- A 4-character value represents the output channels in sequence from 15 thru 0 in an ADAM-5056 module.

- A 2-character value represents the output channels in sequence from 5 thru 0 in an ADAM-5060 module.

- A 2-character value represents the output channels in sequence from 7 thru 0 in ADAM-5055S/5068 module. Each bit represents a channel.

(cr) is the terminating character, carriage return (0Dh)

Example command: **$19S1M(cr)**

response: **!191322(cr)**

The command asks the digital output module in slot 1 of the ADAM-5000 system at address 19h to return the masking status of all of its channels.

The first 2-character portion of the response indicates the address of the ADAM-5000 system.

The second 2-characters portion of the response, value 13h (00010011), indicates that digital output channels 8, 9 and 12 are masked, while channels 10, 11, 13, 14 and 15 are valid.

The third 2-character portion of the response, value 22h (00100010), indicates that digital output channels 1 and 5 are masked, while channels 0, 2, 3, 4, 6 and 7 are valid.

$AASi7 (ADAM-5050 only)

Name Read 5050 channel status

Description The command requests to read 5050 channel status.

Syntax **$AASi7(cr)**

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot i (i : 0 to 3).

7 is the command for the last value readback.

Response **!AAXXXX** if the command is valid.

?AA(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

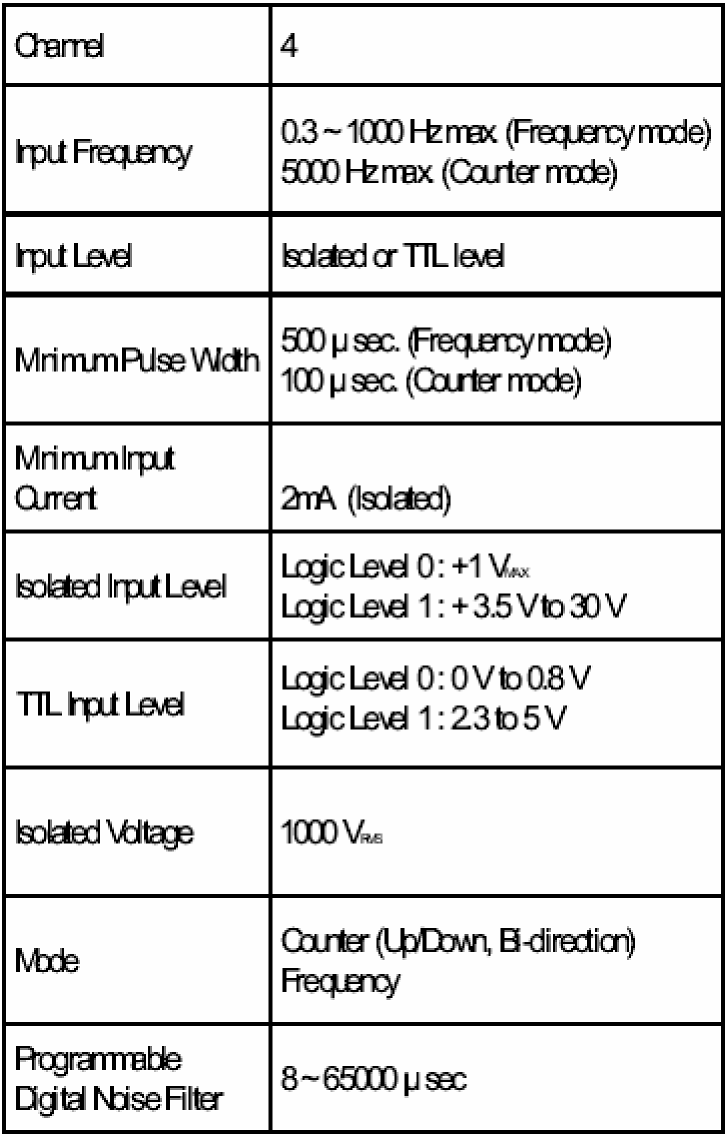
? delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

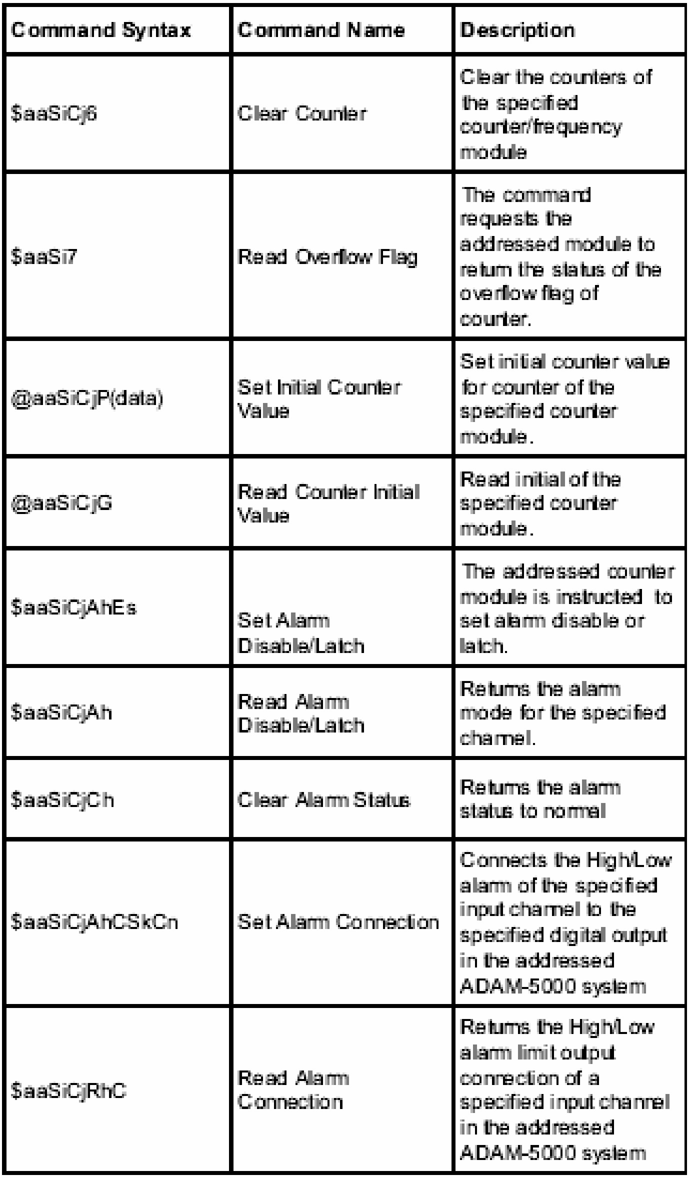
XXXX is the number of overflow for a specified channel. (**XXXX** represents 0~3 channels, each of which is represented by one **XX**).

(cr) is the terminating character, carriage return (0Dh)

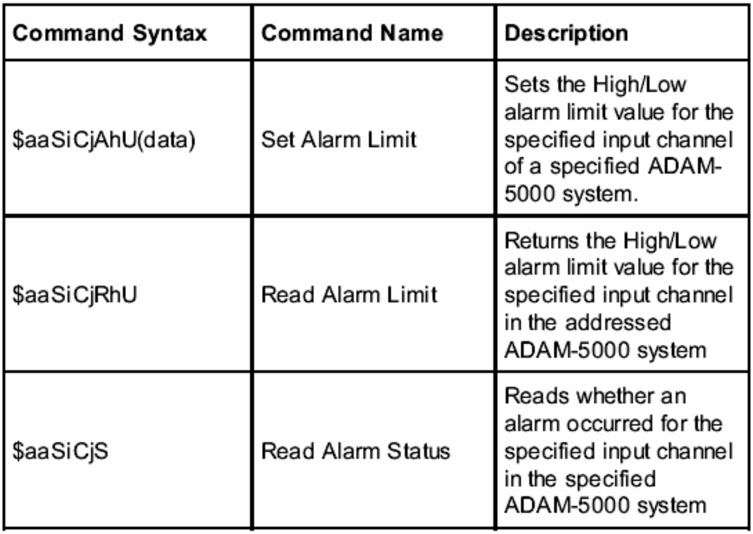
6.10 ADAM-5080 Counter/Frequency Command Set



Channel	4
Input Frequency	0.3 ~ 1000 Hz max. (Frequency mode) 5000 Hz max. (Counter mode)
Input Level	Isolated or TTL level
Minimum Pulse Width	500 µ sec. (Frequency mode) 100 µ sec. (Counter mode)
Minimum Input Current	2mA (Isolated)
Isolated Input Level	Logic Level 0 : +1 V_{MAX} Logic Level 1 : + 3.5 V to 30 V
TTL Input Level	Logic Level 0 : 0 V to 0.8 V Logic Level 1 : 2.3 to 5 V
Isolated Voltage	1000 V_{RMS}
Mode	Counter (Up/Down, Bi-direction) Frequency
Programmable Digital Noise Filter	8 ~ 65000 µ sec



Command Syntax	Command Name	Description
$aaSiCj6	Clear Counter	Clear the counters of the specified counter/frequency module
$aaSi7	Read Overflow Flag	The command requests the addressed module to return the status of the overflow flag of counter.
@aaSiCjP(data)	Set Initial Counter Value	Set initial counter value for counter of the specified counter module.
@aaSiCjG	Read Counter Initial Value	Read initial of the specified counter module.
$aaSiCjAhEs	Set Alarm Disable/Latch	The addressed counter module is instructed to set alarm disable or latch.
$aaSiCjAh	Read Alarm Disable/Latch	Returns the alarm mode for the specified channel.
$aaSiCjCh	Clear Alarm Status	Returns the alarm status to normal
$aaSiCjAhCSkCn	Set Alarm Connection	Connects the High/Low alarm of the specified input channel to the specified digital output in the addressed ADAM-5000 system
$aaSiCjRhC	Read Alarm Connection	Returns the High/Low alarm limit output connection of a specified input channel in the addressed ADAM-5000 system



Command Syntax	Command Name	Description
$aaSiCjAhU(data)	Set Alarm Limit	Sets the High/Low alarm limit value for the specified input channel of a specified ADAM-5000 system.
$aaSiCjRhU	Read Alarm Limit	Returns the High/Low alarm limit value for the specified input channel in the addressed ADAM-5000 system
$aaSiCjS	Read Alarm Status	Reads whether an alarm occurred for the specified input channel in the specified ADAM-5000 system

$aaT

Name Read Module Name

Description Returns the module name from a specified ADAM-5000 system.

Syntax **$aaT(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

T is the command for reading Module Name.

(cr) is the terminating character, carriage return (0Dh).

Response **!aaFFFFFFFF(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

FFFFFFFF indicates the I/O slot which ADAM-5080 module is in.

(cr) is the terminating character, carriage return (0Dh).

$aaT

Example command: $25T(cr)

Response !25FF80FFFF(cr)

ADAM-5080 is plug in slot 1 and the command requests the system at address 25h to send its module name.

$aaF

Name Read Firmware Version

Description Returns the firmware version code from a specified ADAM-5000 system.

Syntax **$aaF(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

F is the command for reading Firmware Version.

(cr) is the terminating character, carriage return (0Dh).

Response **!aa(version)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(version) represents the firmware version of the ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

$aaF

Eample command: **$18F(cr)**

response: **!18A2.3(cr)**

The command requsets the system at address 18h to send its firmware version.

The system responds with firmware version A2.3.

$aaSiArrff

Name Set Configuration

Description Set slot index and counter mode.

Syntax **$aaSiArrff(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to configure.

Si identifies the I/O slot **i** you want to configure.

A is command for setting I/O module configuration.

rr indicates which mode is.
rr=00 represents Bi-direction counter mode.
rr=01 represenrs UP/DOWN counter mode.
rr=02 represents Frequency mode.

ff indicates which format is
ff=**00** represents the engineer format.
ff=02 represents the hexdecimal format.

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiArrff

Example command: **$24S1A0002(cr**)

response: **!24(cr)**

The ADAM-5080 in Slot 1 of ADAM-5000 system at address 24h is in Bi-direction mode and configured for hexdecimal format.

$aaSiB

Name Read Configuration.

Description The command requests the Configuration of slot

Syntax **$aaSiB(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the desired slot i

B represents the configuration status command

(cr) is the terminating character, carriage return (0Dh).

Response. **!aarrff(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command is received.

? delimiter character indicating the command is invalid.
aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

rr=00 represents Bi-direction counter mode.
rr=01 represenrs UP/DOWN counter mode.
rr=02 represents Frequency mode.

ff indicates which format is
ff=**00** represents the engineer format.
ff=02 represents the hexdecimal format.

(cr) is the terminating character, carriage return (0Dh).

$aaSiB

Example command: **$35S3B(cr**)

response: **!350100(cr)**

The ADAM-5080 in Slot 3 of ADAM-5000 system at address 35h responds that it is configured in UP/DOWN counter mode and for engineering unit data format.

#aaSi

Name Read All Channel Counter (Frequency) Data

Description Return the input value of all channels for the specified input module for a specified system in engineering unit only.

Syntax **#aaSi(cr)**

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si is the I/O slot of ADAM-5000 system you want to read.

(cr) is the terminating character, carriage return (0Dh)

Response **>(data) (data) (data) (data) (cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

> is a delimiter character.

? is a delimiter character indicating the command being invalid.

(data) is the input value in engineering units of the interrogated module of the specified system. If the numbers of (data) are ten ,counter/frequency mode is in decimal format.If the numbers of (data) are eight,counter/frequency mode is in hexdecimal format. If (data) = “ ”, it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh).

$aaSi

Example command: **#16S2(cr**)

response:

If the response you got is in Counter mode, you'll see one similar to the example below:

>12354580132675213069345214630518321065 49(cr)

What you see here is actually the input values of all channels that is returned from slot 2 of the ADAM-5000 system at address 16h.

As all 4 values are concatenated into one numerical string such as above, we can still easily discern the values of 4 channels specifically as:

1235458013, 2675213069, 3452146305 and 1832106549

If the response is

>0e88fa63c33697b52a68d61fe2ca6915(cr)

The command requests the module in slot 2 of the ADAM-5000 system at address 16h to return the input values of all channels.

The module response that input values if all channels are hexadecimal:

0e88fa63,c33697b5,2a68d61f,e2ca6915

#aaSi

However, if the response is in frequency mode, you'll see one similar to the example below:

>0000098700000006490000000762000000011600(cr)

As all 4 values are concatenated into one numerical string such as above, we can still easily discern the values of 4 channels specifically as:

0000098700,0000064900,0000076200,0000011600

What you see here is actually the input values of all channels returned from slot 2 of the ADAM-5000 system at address 16h and in decimal format. However, it is not the actual frequency.

Each actual frequency can be obtained by dividing the response value by 100. Therefore, taking an example of the value above, the actual frequency should be:

actual frequency = 98700/100 = 987

If the response is:

>0000F10000020000000310000000DD400(cr)

The command requests the module in slot 2 of the ADAM-5000 system at address 16h to return the input values of all channels.

The module response that input values if all channels are hexadecimal:

0000F100,00020000,00031000,000DD400

The actual frequency can be obtained by transferring hexadecimal format to decimal format .Then divide the response value by 100. Therefore, taking an example of the value above, the actual frequency should be:

F100 (hexadecimal)=24100 (decimal)

actual frequency = 24100/100 = 241

$aaSiCj

Name Read One Channel Counter (Frequency) Data

Description The command will return the input value from one of the four channels of a specified module.

Syntax **#aaSiCj(cr)**

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the I/O slot you want to interrogate.

Cj identifies the channel you want to read.

(cr) is the terminating character, carriage return (0Dh)

Response **>(data)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

> is a delimiter character.

? delimiter character indicating the command was invalid.

(data) is the input value in engineering units of the interrogated module of the specified system. If the numbers of (data) are ten, counter/frequency mode is in decimal format. If the numbers of (data) are eight
, counter/frequency mode is in hexadecimal format. If (data) = “ ”, it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh)

#aaSiCj

Example command: **$35S3C2(cr**)

response: **>0000000451(cr)**

The command requests the ADAM-5080 module in slot 3 of the ADAM-5000 system at address 35h to return the input value of channel 2.

The counter module responds that the input value of channel 2 is 451.

$aaSiØ(data)

Name Set Digital filter Scale

Description Set the filter seconds to start to measure the input signal.

Syntax **$aaSiØ(data)(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system which is to be calibrate.

Si identifies the sepcified slot.

Ø is the command for setting digital filter scale.

(data) represents filter secends from 8µs~65000 µs. Be aware that (data) has 5 characters.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiØ(data)

Example command: **$26S3000765(cr**)

response: **!26(cr)**

The ADAM-5080 in slot 3 of the ADAM-5000 system at address 26h need 765µ seconds to start to measure the input.

$aaSiØ

Name Read Digital filter scale

Description Read the filter seconds to start to measure the input signal.

Syntax **$aaSiØ(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system which is to be calibrate.

Si identifies the I/O slot which is to be accessed.

Ø is the command for reading digital filter scale.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(data)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(data) represents filter secends from 8 µs~65000 µs. Be aware that (data) has 5 characters.

(cr) is the terminating character, carriage return (0Dh)

$aaSiØ

Example command: **$26S30(cr**)

response: **!2600765(cr)**

The command requests the ADAM-5080 in slot 3 of the ADAM-5000 system at address 26h to read the filter seconds. The module responds with 765µ seconds.

$aaSiCj5s

Name Set Counter Start/Stop

Description Request the addressed counter/frequency module to start or stop the counting.

Syntax **$aaSiCj5s(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i and the channel j of the module you want to set.

5 is the command for setting counter Start/Stop.

s represents start/stop command.
s=0 indicate stop counter.
s=1 indicate start counter.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiØ

Example command: **$26S3C251(cr**)

response: **!26(cr)**

The command requests channel 2 of ADAM-5080 in slot 3 in ADAM-5000 system at address 26h to start counter.

$aaSiCj5

Name Read counter Start/Stop

Description Requests the addressed counter/frequency module to indicate whether counters are active.

Syntax **$aaSiCj5(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot **i** and the channel **j** of the module you want to set.

5 is the command for reading counter Start/Stop.

(cr) is the terminating character, carriage return (0Dh)

Response **!aas(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

s represents start/stop command.
s=0 indicate stop counter.
s=1 indicate start counter.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCj5

Example command: **$26S3C25(cr**)

response: **!261(cr)**

The channel 2 of ADAM-5080 in slot 3 in ADAM-5000 system at address 26h is instructed to return its counter status. The counter status is in start status.

$aaSiCj6

Name Clear Counter

Description Clear the counters of the specified counter/frequency module

Syntax **$aaSiCj6(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i and the channel j for the module you want to return a prior value.

6 is the command for clearing counter.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCj6

Example command: **$26S3C26(cr**)

response: **!26(cr)**

The command requests the channel 2 of ADAM-5080 in slot 3 in ADAM-5000 system at address 26h to clear counter value.

$aaSiCj7

Name Read Overflow Flag

Description The command requests the addressed module to return the status of the overflow flag of counter.

Syntax **$aaSi7(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot i (i : 0 to 3).

7 is the command for the last value readback.

Response **!aaff ff ff ff(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

ffffffff is the number of overflow for a specified channel. (**ffffffff** represents 0~3 channels, each of which is represented by one **ff**).

(cr) is the terminating character, carriage return (0Dh)

Note: *When this command is issued, the overflow value is cleared and starts afresh.*

$aaSiCj6

Example command: **$26S37(cr**)

response: **!2600000001(cr)**

The command requests the ADAM-5080 of slot 3 in ADAM-5000 system at address 26h to return the overflow value. The overflow value in channel 3 is 01. The others are 00.

$aaSiCjP(data)

Name Set Initial Counter Value

Description Set initial counter value for counter of the specified counter module.

Syntax **@aaSiCjP(data)(cr)**

@ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i and the channel j for the module you want to return a prior value.

P represents Set Initial Counter Value command.

(data) is initial value from 0 to 4294967296. Be aware that (data) has 10 characters.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCj6

Example command: **@26S3C2P0000004369(cr**)

response: **!26(cr)**

The channel 2 of ADAM-5080 in slot 3 in ADAM-5000 system at address 26h is instructed to set initial counter value. The initial counter value is 4369.

$aaSiCjPG

Name Read Initial Counter

Description Read initial counter value of specified module.

Syntax **@aaSiCjG(cr)**

@ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i and the channel j for the module you want to return a prior value.

G is the last value readback command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(data)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(data) is initial value from 0 to 4294967295.Be aware that (data) has 10 characters.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCjG

Example command: **@26S3C2G(cr**)

response: **!260000004369(cr)**

The channel 2 of ADAM-5080 in slot 3 in ADAM-5000 system at address 26h is instructed to return counter initial value. The initial counter value is 4369.

$aaSiCjAhEs

Name Set Alarm Disable/Latch

Description The addressed counter module is instructed to set alarm disable or latch.

Syntax **$aaSiCjAhEs(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j. **AhEs** is the command for setting Alarm Disable/Latch Mode command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

s indicates alarm enable/disable and can have the value D = Disable, E=Enable

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjG

Example command: **$03S0C1ALED(cr**)

response: **!03(cr)**

Channel 1 of slot 0 of ADAM-5080 in ADAM-5000 system at address 03h is instructed to disable its Low alarm function.

The module confirms that its Low alarm function has been disable.

$aaSiCjAh

Name Read Alarm Disable/Latch

Description Return the alarm mode for the specified channel.

Syntax **$aaSiCjAh(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

A is the Read Alarm Mode command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aap(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

p indicates alarm mode.
p=D, if alarm is Disable.
P=L, if alarm is Latch.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjAh

Example command: **$03S0C1AL(cr**)

response: **!03L(cr)**

Channel 1 of slot 0 of ADAM-5080 in ADAM-5000 system at address 03h is instructed to return its Low alarm mode.

The system responds that it is latched.

$aaSiCjCh

Name Clear Alarm Status

Description Returns the alarm status to normal

Syntax **$aaSiCjCh(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

C is the clear Alarm Mode command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjCh

Example command: **$03S0C1CL(cr**)

response: **!03(cr)**

Channel 1 of slot 0 of ADAM-5080 in ADAM-5000 system at address 03h is instructed to set its Low alarm state to normal.

The system confirms it has done so accordingly.

$aaSiCAhCSkCh

Name Set Alarm Connection

Description Connect the High/Low alarm of the specified input channel to the specified digital output in the addressed ADAM-5000 system

Syntax **$aaSiCjAhCSkCn(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j . **AhC** is the command for setting Alarm Connection command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

SkCn identifies the desired slot k and the desired digital output point n (n : 0 to F). To disconnect the digital output, k and n should be set as '*'.

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCAhCSkCh

Example command: **$03S0C1ALCS1C0(cr**)

response: **!03(cr)**

Channel 1 of slot 0 of ADAM-5080 in ADAM-5000 system at address 03h is instructed to connect its Low alarm to the digital output of point 0 of slot 1 in the same ADAM-5000 system.

The system confirms it has dome so accordingly.

$aaSiCjRhC

Name Read Alarm Connection

Description Return the High/Low alarm limit output connection of a specified input channel in the addressed ADAM-5000 system

Syntax **$aaSiCjRhC(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

RhC is the command for reading Alarm Connection.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aaSkCn(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

SkCn identifies the desired slot k and the desired digital output point n (n : 0 to F) to which the input alarm is connected. If the values of k and n are ‘*’, the analog input has no connection with a digital output point.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCAhCSkCh

Example command: **$03S0C1RLC(cr**)

response: **!03SØC1(cr)**

Channel 1 of slot 0 of ADAM-5080 in ADAM-5000 system at address 03h is instructed to read its Low alarm output connection.

The system responds that the Low alarm output connects to the digital output at point 0 of slot 1 in the same ADAM-5000 system.

$aaSiCjAhU(data)

Name Set Alarm Limit

Description Set the High/Low alarm limit value for the specified input channel of a specified ADAM-5000 system.

Syntax **$aaSiCjAhU(data)(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

AhU is the Set Alarm Limit command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(data) represents the desired alarm limit setting. The value is from 0 to 4294967295. Be aware that (data) has 10 characters.

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjAhU(data)

Example command: **$03SØC1AHU0000000020(cr**)

response: **!03(cr)**

The channel 1 of slot 0 of ADAM-5080 in ADAM-5000 system at address 03h is configured to set High alarm limit value to 20.

$aaSiCjRhU

Name Read Alarm Limit

Description Return the High/Low alarm limit value for the specified input channel in the addressed ADAM-5000 system

Syntax **$aaSiCjRhU(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

RhU is the Read Alarm Limit command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(data)(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(data) represents the desired alarm limit setting. The format is always in engineering units. Be aware that (data) has 10 characters.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjRhU

Example command: **$03SØC1RHU(cr**)

response: **!030000000026(cr)**

The channel 1 of slot 0 of ADAM-5080 in the ADAM-5000 system at address 03h is configured to return the High alarm limit value.

The High alarm limit value is 26.

$aaSiCjS

Name Read Alarm Status

Description Read whether an alarm occurred for the specified input channel in the specified ADAM-5000 system

Syntax **$aaSiCjS(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

S is the Read Alarm Status command.

(cr) represents terminating character, carriage return (0Dh)

Response **!aahl(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

h represents the status of High alarm. ‘1’ means the High alarm occurred, ‘0’ means it did not occur.

l represents the status of Low alarm. ‘1’ means the Low alarm occurred, ‘0’ means it did not occur.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjS

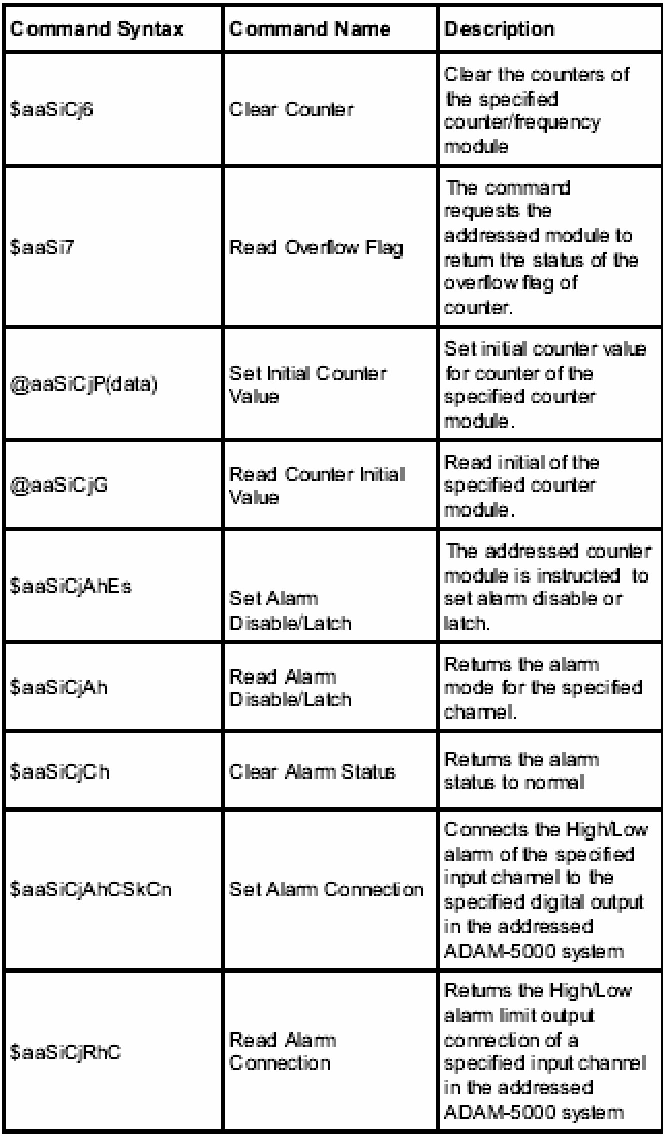
Example command: **$03SØC1S**

response: **!0311(cr)**

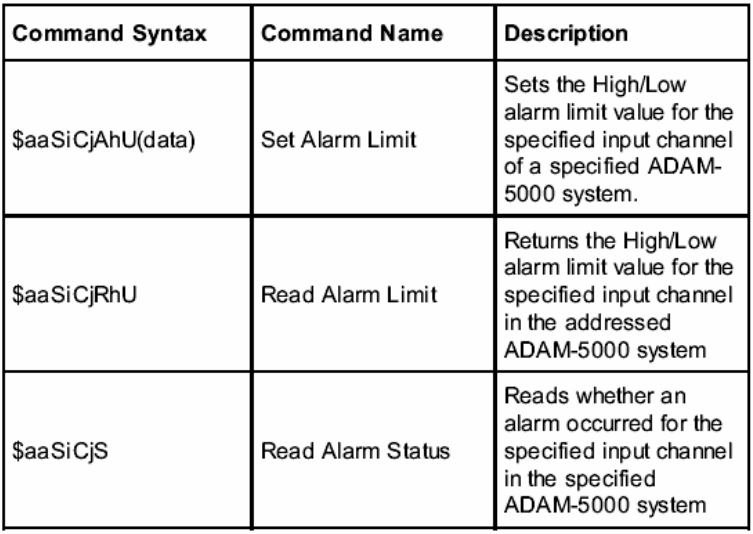
The channel 1 of slot 0 of ADAM-5080 in the ADAM-5000 system at address 03h is configured to read alarm status.

The High alarm has occured and low alarm has occured.

6.11 ADAM-5081 Counter/Frequency Command Set



Command Syntax	Command Name	Description
$aaSiCj6	Clear Counter	Clear the counters of the specified counter/frequency module
$aaSi7	Read Overflow Flag	The command requests the addressed module to return the status of the overflow flag of counter.
@aaSiCjP(data)	Set Initial Counter Value	Set initial counter value for counter of the specified counter module.
@aaSiCjG	Read Counter Initial Value	Read initial of the specified counter module.
$aaSiCjAhEs	Set Alarm Disable/Latch	The addressed counter module is instructed to set alarm disable or latch.
$aaSiCjAh	Read Alarm Disable/Latch	Returns the alarm mode for the specified channel.
$aaSiCjCh	Clear Alarm Status	Returns the alarm status to normal
$aaSiCjAhCSkCn	Set Alarm Connection	Connects the High/Low alarm of the specified input channel to the specified digital output in the addressed ADAM-5000 system
$aaSiCjRhC	Read Alarm Connection	Returns the High/Low alarm limit output connection of a specified input channel in the addressed ADAM-5000 system



Command Syntax	Command Name	Description
$aaSiCjAhU(data)	Set Alarm Limit	Sets the High/Low alarm limit value for the specified input channel of a specified ADAM-5000 system.
$aaSiCjRhU	Read Alarm Limit	Returns the High/Low alarm limit value for the specified input channel in the addressed ADAM-5000 system
$aaSiCjS	Read Alarm Status	Reads whether an alarm occurred for the specified input channel in the specified ADAM-5000 system

$aaT

Name Read Module Name

Description Returns the module name from a specified ADAM-5000 system.

Syntax **$aaT(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

T is the command for reading Module Name.

(cr) is the terminating character, carriage return (0Dh).

Response **!aaFFFFFFFF(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

FFFFFFFF indicates the I/O slot which ADAM-5081 module is in.

(cr) is the terminating character, carriage return (0Dh).

$aaT

Example command: $25T(cr)

Response !25FF80FFFF(cr)

ADAM-5081 is plug in slot 1 and the command requests the system at address 25h to send its module name.

$aaF

Name Read Firmware Version

Description Returns the firmware version code from a specified ADAM-5000 system.

Syntax **$aaF(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

F is the command for reading Firmware Version.

(cr) is the terminating character, carriage return (0Dh).

Response **!aa(version)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error, communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(version) represents the firmware version of the ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh).

$aaF

Eample command: **$18F(cr)**

response: **!18A2.3(cr)**

The command requsets the system at address 18h to send its firmware version.

The system responds with firmware version A2.3.

$aaSiArrff

Name Set Configuration

Description Set slot index and counter mode.

Syntax **$aaSiArrff(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to configure.

Si identifies the I/O slot **i** you want to configure.

A is command for setting I/O module configuration.

rr indicates which mode is.
rr=00 represents Bi-direction counter mode.
rr=01 represenrs UP/DOWN counter mode.
rr=02 represents Frequency mode.

ff indicates which format is
ff=**00** represents the engineer format.
ff=02 represents the hexdecimal format.

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiArrff

Example command: **$24S1A0002(cr**)

response: **!24(cr)**

The ADAM-5081 in Slot 1 of ADAM-5000 system at address 24h is in Bi-direction mode and configured for hexdecimal format.

$aaSiB

Name Read Configuration.

Description The command requests the Configuration of slot

Syntax **$aaSiB(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the desired slot i

B represents the configuration status command

(cr) is the terminating character, carriage return (0Dh).

Response. **!aarrff(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command is received.

? delimiter character indicating the command is invalid.
aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

rr=00 represents Bi-direction counter mode.
rr=01 represenrs UP/DOWN counter mode.
rr=02 represents Frequency mode.

ff indicates which format is
ff=**00** represents the engineer format.
ff=02 represents the hexdecimal format.

(cr) is the terminating character, carriage return (0Dh).

$aaSiB

Example command: **$35S3B(cr**)

response: **!350100(cr)**

The ADAM-5081 in Slot 3 of ADAM-5000 system at address 35h responds that it is configured in UP/DOWN counter mode and for engineering unit data format.

#aaSi

Name Read All Channel Counter (Frequency) Data

Description Return the input value of all channels for the specified input module for a specified system in engineering unit only.

Syntax **#aaSi(cr)**

# is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si is the I/O slot of ADAM-5000 system you want to read.

(cr) is the terminating character, carriage return (0Dh)

Response **>(data) (data) (data) (data)   (cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

> is a delimiter character.

? is a delimiter character indicating the command being invalid.

(data) is the input value in engineering units of the interrogated module of the specified system. If the numbers of (data) are ten ,counter/frequency mode is in decimal format.If the numbers of (data) are eight,counter/frequency mode is in hexdecimal format. If (data) = “ ”, it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh).

$aaSi

Example command: **#16S2(cr**)

response:

If the response you got is in Counter mode, you'll see one similar to the example below:

>1235458013267521306934521463051832106549(cr)

What you see here is actually the input values of all channels that is returned from slot 2 of the ADAM-5000 system at address 16h.

As all 4 values are concatenated into one numerical string such as above, we can still easily discern the values of 4 channels specifically as:

1235458013, 2675213069, 3452146305 and 1832106549

If the response is

>0e88fa63c33697b52a68d61fe2ca6915(cr)

The command requests the module in slot 2 of the ADAM-5000 system at address 16h to return the input values of all channels.

The module response that input values if all channels are hexadecimal:

0e88fa63,c33697b5,2a68d61f,e2ca6915

#aaSi

However, if the response is in frequency mode, you'll see one similar to the example below:

>0000098700000000649000000076200000011600(cr)

As all 4 values are concatenated into one numerical string such as above, we can still easily discern the values of 4 channels specifically as:

0000098700,0000064900,0000076200,0000011600

What you see here is actually the input values of all channels returned from slot 2 of the ADAM-5000 system at address 16h and in decimal format. However, it is not the actual frequency.

Each actual frequency can be obtained by dividing the response value by 100. Therefore, taking an example of the value above, the actual frequency should be:

actual frequency = 98700/100 = 987

If the response is:

>0000F10000020000000310000000DD400(cr)

The command requests the module in slot 2 of the ADAM-5000 system at address 16h to return the input values of all channels.

The module response that input values if all channels are hexadecimal:

0000F100,00020000,00031000,000DD400

The actual frequency can be obtained by transferring hexadecimal format to decimal format .Then divide the response value by 100. Therefore, taking an example of the value above, the actual frequency should be:

F100 (hexadecimal)=24100 (decimal)

actual frequency = 24100/100 = 241

$aaSiCj

Name Read One Channel Counter (Frequency) Data

Description The command will return the input value from one of the four channels of a specified module.

Syntax **#aaSiCj(cr)**

is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system you want to interrogate.

Si identifies the I/O slot you want to interrogate.

Cj identifies the channel you want to read.

(cr) is the terminating character, carriage return (0Dh)

Response **>(data)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

> is a delimiter character.

? delimiter character indicating the command was invalid.

(data) is the input value in engineering units of the interrogated module of the specified system. If the numbers of (data) are ten, counter/frequency mode is in decimal format. If the numbers of (data) are eight
, counter/frequency mode is in hexadecimal format. If (data) = “ ”, it means the channel is invalid.

(cr) is the terminating character, carriage return (0Dh)

#aaSiCj

Example command: **$35S3C2(cr**)

response: **>0000000451(cr)**

The command requests the ADAM-5081 module in slot 3 of the ADAM-5000 system at address 35h to return the input value of channel 2.

The counter module responds that the input value of channel 2 is 451.

$aaSiØ(data)

Name Set Digital filter Scale

Description Set the filter seconds to start to measure the input signal.

Syntax **$aaSiØ(data)(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system which is to be calibrate.

Si identifies the sepcified slot.

Ø is the command for setting digital filter scale.

(data) represents filter secends from 8µs~65000 µs. Be aware that (data) has 5 characters.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiØ(data)

Example command: **$26S3000765(cr**)

response: **!26(cr)**

The ADAM-5081 in slot 3 of the ADAM-5000 system at address 26h need 765µ seconds to start to measure the input.

$aaSiØ

Name Read Digital filter scale

Description Read the filter seconds to start to measure the input signal.

Syntax **$aaSiØ(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system which is to be calibrate.

Si identifies the I/O slot which is to be accessed.

Ø is the command for reading digital filter scale.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(data)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(data) represents filter secends from 8 µs~65000 µs. Be aware that (data) has 5 characters.

(cr) is the terminating character, carriage return (0Dh)

$aaSiØ

Example command: **$26S30(cr**)

response: **!2600765(cr)**

The command requests the ADAM-5081 in slot 3 of the ADAM-5000 system at address 26h to read the filter seconds. The module responds with 765µ seconds.

$aaSiCj5s

Name Set Counter Start/Stop

Description Request the addressed counter/frequency module to start or stop the counting.

Syntax **$aaSiCj5s(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i and the channel j of the module you want to set.

5 is the command for setting counter Start/Stop.

s represents start/stop command.
s=0 indicate stop counter.
s=1 indicate start counter.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiØ

Example command: **$26S3C251(cr**)

response: **!26(cr)**

The command requests channel 2 of ADAM-5081 in slot 3 in ADAM-5000 system at address 26h to start counter.

$aaSiCj5

Name Read counter Start/Stop

Description Requests the addressed counter/frequency module to indicate whether counters are active.

Syntax **$aaSiCj5(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot **i** and the channel **j** of the module you want to set.

5 is the command for reading counter Start/Stop.

(cr) is the terminating character, carriage return (0Dh)

Response **!aas(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exists.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

s represents start/stop command.
s=0 indicate stop counter.
s=1 indicate start counter.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCj5

Example command: **$26S3C25(cr**)

response: **!261(cr)**

The channel 2 of ADAM-5081 in slot 3 in ADAM-5000 system at address 26h is instructed to return its counter status. The counter status is in start status.

$aaSiCj6

Name Clear Counter

Description Clear the counters of the specified counter/frequency module

Syntax **$aaSiCj6(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i and the channel j for the module you want to return a prior value.

6 is the command for clearing counter.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCj6

Example command: **$26S3C26(cr**)

response: **!26(cr)**

The command requests the channel 2 of ADAM-5081 in slot 3 in ADAM-5000 system at address 26h to clear counter value.

$aaSiCj7

Name Read Overflow Flag

Description The command requests the addressed module to return the status of the overflow flag of counter.

Syntax **$aaSi7(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

Si identifies the I/O slot i (i : 0 to 3).

7 is the command for the last value readback.

Response **!aaff ff ff ff(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

ffffffff is the number of overflow for a specified channel. (**ffffffff** represents 0~3 channels, each of which is represented by one **ff**).

(cr) is the terminating character, carriage return (0Dh)

Note: *When this command is issued, the overflow value is cleared and starts afresh.*

$aaSiCj6

Example

command: **$26S37(cr**)

response: **!2600000001(cr)**

The command requests the ADAM-5081 of slot 3 in ADAM-5000 system at address 26h to return the overflow value. The overflow value in channel 3 is 01. The others are 00.

$aaSiCjP(data)

Name Set Initial Counter Value

Description Set initial counter value for counter of the specified counter module.

Syntax **@aaSiCjP(data)(cr)**

@ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i and the channel j for the module you want to return a prior value.

P represents Set Initial Counter Value command.

(data) is initial value from 0 to 4294967296. Be aware that (data) has 10 characters.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCj6

Example command: **@26S3C2P0000004369(cr**)

response: **!26(cr)**

The channel 2 of ADAM-5081 in slot 3 in ADAM-5000 system at address 26h is instructed to set initial counter value. The initial counter value is 4369.

$aaSiCjPG

Name Read Initial Counter

Description Read initial counter value of specified module.

Syntax **@aaSiCjG(cr)**

@ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000 system.

SiCj identifies the I/O slot i and the channel j for the module you want to return a prior value.

G is the last value readback command.

(cr) is the terminating character, carriage return (0Dh)

Response **!aa(data)(cr)** if the command is valid.

?aa(cr) if an invalid operation was entered.

There is no response if the module detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

(data) is initial value from 0 to 4294967295.Be aware that (data) has 10 characters.

(cr) is the terminating character, carriage return (0Dh)

$aaSiCjG

Example command: **@26S3C2G(cr**)

response: **!260000004369(cr)**

The channel 2 of ADAM-5081 in slot 3 in ADAM-5000 system at address 26h is instructed to return counter initial value. The initial counter value is 4369.

$aaSiCjAhEs

Name Set Alarm Disable/Latch

Description The addressed counter module is instructed to set alarm disable or latch.

Syntax **$aaSiCjAhEs(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j. **AhEs** is the command for setting Alarm Disable/Latch Mode command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

s indicates alarm enable/disable and can have the value D = Disable, E=Enable

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjG

Example command: **$03S0C1ALED(cr**)

response: **!03(cr)**

Channel 1 of slot 0 of ADAM-5081 in ADAM-5000 system at address 03h is instructed to disable its Low alarm function.

The module confirms that its Low alarm function has been disable.

$aaSiCjAh

Name Read Alarm Disable/Latch

Description Return the alarm mode for the specified channel.

Syntax **$aaSiCjAh(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

A is the Read Alarm Mode command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aap(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

p indicates alarm mode.
p=D, if alarm is Disable.
P=L, if alarm is Latch.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjAh

Example command: **$03S0C1AL(cr**)

response: **!03L(cr)**

Channel 1 of slot 0 of ADAM-5081 in ADAM-5000 system at address 03h is instructed to return its Low alarm mode.

The system responds that it is latched.

$aaSiCjCh

Name Clear Alarm Status

Description Returns the alarm status to normal

Syntax **$aaSiCjCh(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

C is the clear Alarm Mode command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjCh

Example command: **$03S0C1CL(cr**)

response: **!03(cr)**

Channel 1 of slot 0 of ADAM-5081 in ADAM-5000 system at address 03h is instructed to set its Low alarm state to normal.

The system confirms it has done so accordingly.

$aaSiCAhCSkCh

Name Set Alarm Connection

Description Connect the High/Low alarm of the specified input channel to the specified digital output in the addressed ADAM-5000 system

Syntax **$aaSiCjAhCSkCn(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j . **AhC** is the command for setting Alarm Connection command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

SkCn identifies the desired slot k and the desired digital output point n (n : 0 to F). To disconnect the digital output, k and n should be set as ‘*’.

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCAhCSkCh

Example command: **$03S0C1ALCS1C0(cr**)

response: **!03(cr)**

Channel 1 of slot 0 of ADAM-5081 in ADAM-5000 system at address 03h is instructed to connect its Low alarm to the digital output of point 0 of slot 1 in the same ADAM-5000 system.

The system confirms it has dome so accordingly.

$aaSiCjRhC

Name	Read Alarm Connection
Description	Return the High/Low alarm limit output connection of a specified input channel in the addressed ADAM-5000 system
Syntax	**$aaSiCjRhC(cr)** **$** is a delimiter character. **aa** (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system. **SiCj** identifies the desired slot i and the desired channel j. **RhC** is the command for reading Alarm Connection. **h** indicates alarm type and can have the value H = High alarm, L = Low alarm **(cr)** represents terminating character, carriage return (0Dh)
Response	**!aaSkCn(cr)** if the command was valid There is no response if the system detects a syntax error or communication error or if the specified address does not exist. **!** delimiter character indicating a valid command was received. **aa** represents the 2-character hexadecimal address of the corresponding ADAM-5000 system. **SkCn** identifies the desired slot k and the desired digital output point n (n : 0 to F) to which the input alarm is connected. If the values of k and n are '*', the analog input has no connection with a digital output point. **(cr)** represents terminating character, carriage return (0Dh)

$aaSiCAhCSkCh

Example command: **$03S0C1RLC(cr**)

response: **!03SØC1(cr)**

Channel 1 of slot 0 of ADAM-5081 in ADAM-5000 system at address 03h is instructed to read its Low alarm output connection.

The system responds that the Low alarm output connects to the digital output at point 0 of slot 1 in the same ADAM-5000 system.

$aaSiCjAhU(data)

Name Set Alarm Limit

Description Set the High/Low alarm limit value for the specified input channel of a specified ADAM-5000 system.

Syntax **$aaSiCjAhU(data)(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

AhU is the Set Alarm Limit command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(data) represents the desired alarm limit setting. The value is from 0 to 4294967295. Be aware that (data) has 10 characters.

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjAhU(data)

Example command: **$03SØC1AHU0000000020(cr**)

response: **!03(cr)**

The channel 1 of slot 0 of ADAM-5081 in ADAM-5000 system at address 03h is configured to set High alarm limit value to 20.

$aaSiCjRhU

Name Read Alarm Limit

Description Return the High/Low alarm limit value for the specified input channel in the addressed ADAM-5000 system

Syntax **$aaSiCjRhU(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.

SiCj identifies the desired slot i and the desired channel j.

RhU is the Read Alarm Limit command.

h indicates alarm type and can have the value H = High alarm, L = Low alarm

(cr) represents terminating character, carriage return (0Dh)

Response **!aa(data)(cr)** if the command was valid

There is no response if the system detects a syntax error or communication error or if the specified address does not exist.

! delimiter character indicating a valid command was received.

aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.

(data) represents the desired alarm limit setting. The format is always in engineering units. Be aware that (data) has 10 characters.

(cr) represents terminating character, carriage return (0Dh)

$aaSiCjRhU

Example command: **$03SØC1RHU(cr**)

response: **!030000000026(cr)**

The channel 1 of slot 0 of ADAM-5081 in the ADAM-5000 system at address 03h is configured to return the High alarm limit value.

The High alarm limit value is 26.

$aaSiCjS

Name	Read Alarm Status
Description	Read whether an alarm occurred for the specified input channel in the specified ADAM-5000 system
Syntax	**$aaSiCjS(cr)**
	$ is a delimiter character.
	aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000 system.
	SiCj identifies the desired slot i and the desired channel j.
	S is the Read Alarm Status command.
	(cr) represents terminating character, carriage return (0Dh)
Response	**!aahl(cr)** if the command was valid
	There is no response if the system detects a syntax error or communication error or if the specified address does not exist.
	! delimiter character indicating a valid command was received.
	aa represents the 2-character hexadecimal address of the corresponding ADAM-5000 system.
	h represents the status of High alarm. ‘1’ means the High alarm occurred, ‘0’ means it did not occur.
	l represents the status of Low alarm. ‘1’ means the Low alarm occurred, ‘0’ means it did not occur.
	(cr) represents terminating character, carriage return (0Dh)

$aaSiCjS

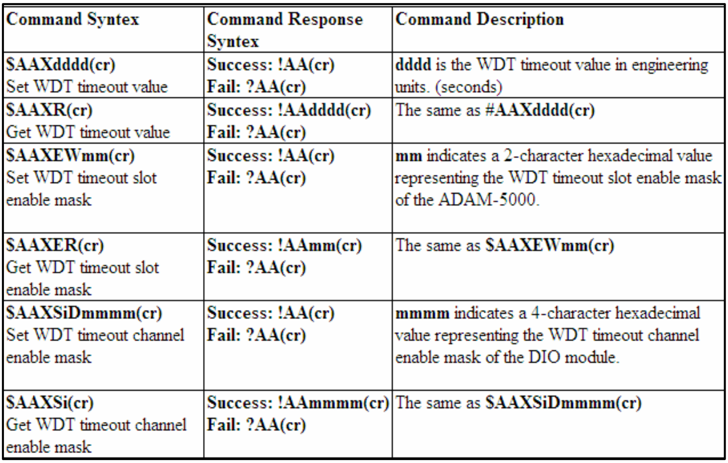
Example command: **$03SØC1S**

response: **!0311(cr)**

The channel 1 of slot 0 of ADAM-5081 in the ADAM-5000 system at address 03h is configured to read alarm status.

The High alarm has occured and low alarm has occured.

6.12 WatchDog Timer Command Set



Command Syntex	Command Response Syntex	Command Description
$AAXdddd(cr) Set WDT timeout value	**Success: !AA(cr)** **Fail: ?AA(cr)**	**dddd** is the WDT timeout value in engineering units. (seconds)
$AAXR(cr) Get WDT timeout value	**Success: !AAdddd(cr)** **Fail: ?AA(cr)**	The same as **#AAXdddd(cr)**
$AAXEWmm(cr) Set WDT timeout slot enable mask	**Success: !AA(cr)** **Fail: ?AA(cr)**	**mm** indicates a 2-character hexadecimal value representing the WDT timeout slot enable mask of the ADAM-5000.
$AAXER(cr) Get WDT timeout slot enable mask	**Success: !AAmm(cr)** **Fail: ?AA(cr)**	The same as **$AAXEWmm(cr)**
$AAXSiDmmmm(cr) Set WDT timeout channel enable mask	**Success: !AA(cr)** **Fail: ?AA(cr)**	**mmmm** indicates a 4-character hexadecimal value representing the WDT timeout channel enable mask of the DIO module.
$AAXSi(cr) Get WDT timeout channel enable mask	**Success: !AAmmmm(cr)** **Fail: ?AA(cr)**	The same as **$AAXSiDmmmm(cr)**

$AAXdddd

Description Set WDT timeout value

Syntax $AAXdddd(cr)

dddd is the WDT timeout value in engineering units. (seconds)

Response Success: !AA(cr)

Fail: ?AA(cr)

Example Command: $01X1234

Response:!01

$AAXR

Description Get WDT timeout value

Syntax $AAXR(cr)

dddd is the WDT timeout value in engineering units. (seconds)

Response Success: !AA(cr)

Fail: ?AA(cr)

Example Command: $01XR

Response: !011234

$AAXEWmm

Description Set WDT timeout slot enable mask

Syntax $AAXEWmm(cr)

mm indicates a 2-character hexadecimal value representing the WDT timeout slot enable mask of the ADAM-5000.

Response Success: !AA(cr)

Fail: ?AA(cr)

Example Command: $01XEWFF

Response: !01

$AAXER

Description Get WDT timeout slot enable mask

Syntax $AAXER(cr)

mm indicates a 2-character hexadecimal value representing the WDT timeout slot enable mask of the ADAM-5000.

Response Success: !AAmm(cr)

Fail: ?AA(cr)

Example Command: $01XER

Response: !01FF

$AAXSiDmmmm

Description Set WDT timeout channel enable mask

Syntax $AAXSiDmmmm(cr)

mmmm indicates a 4-character hexadecimal value representing the WDT timeout channel enable mask of the DIO module.

Response Success: !AA(cr)

Fail: ?AA(cr)

Example Command: $01XS0DFFFF

Response: !01

$AAXSi

Description Get WDT timeout channel enable mask

Syntax $AAXSi(cr)

mmmm indicates a 4-character hexadecimal value representing the WDT timeout channel enable mask of the DIO module.

Response Success: !AAmmmm(cr)

Fail: ?AA(cr)

Example Command: $01XS0

Response: !01FFFF

7

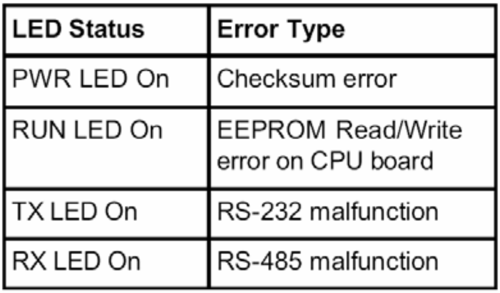
Troubleshooting

The ADAM-5000 system provides two kinds of diagnosis: hardware diagnosis and software diagnosis to help the user detect and identify various types of system and I/O module failures.

7.1 Hardware Diagnosis

When the ADAM-5000 is first powered on, the system does a self-diagnosis. The diagnosis information will be indicated on the LEDs of the system module in the following sequence:

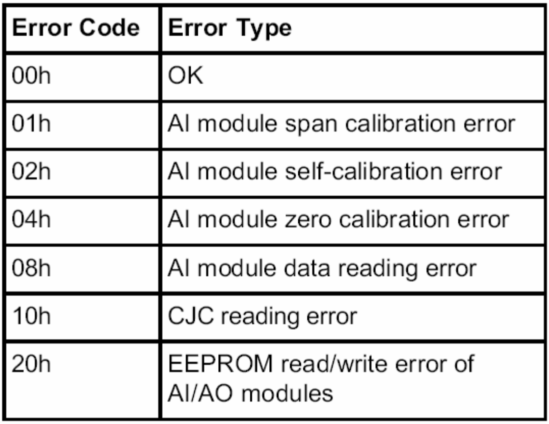
1. The LEDs will come on according to the following sequence: PWR -> RUN -> TX -> RX,
2. Then all LEDs will go off
3. If the system self test is OK, the LEDs will follow the sequence outlined in Steps 1 and 2. If the system has some problems, the LEDs indicate errors as shown in the following table.



LED Status	Error Type
PWR LED On	Checksum error
RUN LED On	EEPROM Read/Write error on CPU board
TX LED On	RS-232 malfunction
RX LED On	RS-485 malfunction

7.2 Software Diagnosis

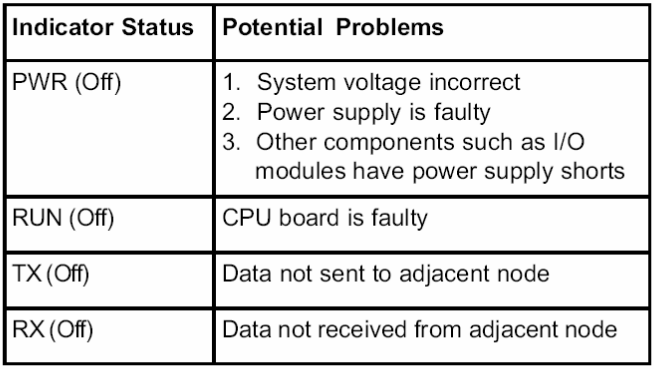
The ADAM-5000 provides an ASCII command $AAE (Refer to the command set in Chapter 6) to read the status of I/O modules through the RS-232 port for field diagnosis or RS-485 port for remote diagnosis. The response of such a command is !AAFFFFFFFF (FF represents the error message of a slot from 0 to 3). The error messages are detailed in the following table:



Error Code	Error Type
00h	OK
01h	AI module span calibration error
02h	AI module self-calibration error
04h	AI module zero calibration error
08h	AI module data reading error
10h	CJC reading error
20h	EEPROM read/write error of AI/AO modules

7.3 System Indicators

While the ADAM-5000 system is in operation the indicators on the front can help you diagnose problems with the system. The table below gives a quick reference of potential problems associated with each status indicator.



Indicator Status	Potential Problems
PWR (Off)	1. System voltage incorrect 2. Power supply is faulty 3. Other components such as I/O modules have power supply shorts
RUN (Off)	CPU board is faulty
TX (Off)	Data not sent to adjacent node
RX (Off)	Data not received from adjacent node

POW Indicator

In general there are 3 reasons for the system power status LED (PWR) to be OFF.

1. External power to the system is incorrect or is not applied.
2. Power supply is faulty.
3. Other component(s) have the power supply shut down.

Incorrect External Power

If the voltage to the power supply is not correct, the system may not operate properly or may not operate at all. Use the following guidelines to correct the problem.

1. First, turn off the system power and check all incoming wiring for loose connections.
2. If the connections are acceptable, reconnect the system power and measure the voltage at the power terminal strip to insure it is within specification. If the voltage is not correct, shut down the system and correct the problem.
3. If all wiring is connected correctly and the incoming power is within the specifications required, the system module should be returned for repair.

Faulty Power Supply

Substitute a power supply known to be good to see if this corrects the problem. If you have experienced major power surges, it is possible that the system and power supply have been damaged. If you suspect this is the cause of the power supply damage, a line conditioner which removes damaging voltage spikes should be used in the future.

Device or Module causing Power Supply to Shutdown

It is possible a faulty module or external device using the system 5V can shut down the power supply. This 5V can be coming from the system module.

To test for a device causing this problem:

1. Turn off power to the system module
2. Disconnect all external devices (such as communication cables) from the system module.
3. Reapply power to the system.

If the power supply operates normally, you probably have either a shorted device or a shorted cable. If the power supply does not operate normally, then test for a module causing the problem by using the following procedure.

To isolate which module is causing the problem, disconnect the external power and remove one module at one time till the PWR LED operates normally. Follow the procedure below:

1. Turn off power to the base.
2. Remove a module from the base.
3. Reapply power to the base.

RUN Indicator

If the self test for the system has passed, the RUN LED is still on except no power.

7.4 Communication Problems

If you cannot establish communications with the system module, check these items:

- The cable is disconnected
- The cable has a broken wire or has been wired incorrectly
- The cable is improperly terminated or grounded
- The device connected is not operating at the correct baud rate
- The device connected to the port is sending data incorrectly
- A grounding difference exists between the two devices
- Electrical noise is causing intermittent errors
- The system module has a bad communication port and it should be replaced

7.5 I/O Module Troubleshooting

There is a LED to indicate the connection between the base and an I/O module in any ADAM-5000 system. The LED is on when the connection is good. If you suspect an I/O error, there are several things that could be causing the problem.

- A loose terminal block
- The power supply has failed
- The module has failed

Some Quick Steps

When troubleshooting the ADAM-5000 series digital I/O modules, there are a few facts you should be aware of. These facts may assist you in quickly correcting an I/O problem.

- The digital output modules can not detect shorted or open output points. If you suspect one or more points on an output module to be faulty, you should measure the voltage drop from the common to the suspect point.
- Leakage current can be a problem when connecting field devices to I/O modules. False input signals can be generated when the leakage current of an output device is great enough to turn on the connected input device. To correct this, install a resistor in parallel with the input or output of the circuit. The value of this resistor will depend on the amount of leakage current and the voltage applied but usually a 10K to 20 KΩ resistor will work. Insure the wattage rating of the resistor is correct for your applications.
- The easiest method to determine if a module has failed is to replace it if you have a spare. However, if you suspect another device to have caused the failure in the replacement module as well. As a point of caution, you may want to check devices or power supplies connected to the failed module before replacing it with a spare module.

A

Quick Start Example

Quick Start Example

This chapter provides guidelines to what is needed to set up and install a distributed ADAM-5000 network system. A quick hookup scheme is provided that lets you configure a single system before you install a network system.

Be sure to carefully plan the layout and configuration of your network before you start. Guidelines regarding layout are given in Appendix B: RS-485 Network.

A.1 System Requirement to setup an ADAM-5000 System

Before you setup an ADAM-5000 system, you can follow the steps to install I/O modules into the ADAM-5000 base.

1. Align the module with the grooves on the top and bottom of the base.
2. Push the unit straight into the base until it is firmly seated in the backplane connector.
3. Push-in the retaining clips at the top and bottom of the unit to secure the module to the base.

The following list gives an overview of what is needed to setup, install and configure an ADAM-5000 environment.

- ADAM-5000/485 and I/O Modules
- A host computer, such as an IBM PC/AT compatible, that can output ASCII characters with an RS-232C or RS-485 port
- Power supply for the ADAM-5000 system (+10 to +30 V_{DC})
- ADAM Series Utility Software
- ADAM Isolated RS-232/RS-485 Converter (optional)
- ADAM Repeater (optional)

Host Computer

Any computer or terminal that can output in ASCII format over either RS-232 or RS-485 can be connected as the host computer. When only RS-232 is available, an ADAM RS-232/RS-485 Converter is required to transform the host signals to the correct RS-485 protocol. The converter also provides opto-isolation and transformer -based isolation to protect your equipment.

Power Supply

For the ease of use in industrial environments the ADAM-5000 systems designed to accept industry standard +24 V_{DC} unregulated power. Operation is guaranteed when using any power supply between +10 and +30 V_{DC}. Power ripples must be limited to 100 mV peak to peak while the voltage in all cases must be maintained between +10 and +30 V_{DC}. When the systems are powered remotely the effects of line voltage drops must be considered.

All systems onboard switching regulators to sustain good efficiency over the 10-30 V input range, therefore, we can assume that the actual current draw is inversely proportional to the line voltage. The follow- ing example shows how to calculate the required current that a power supply should be able to provide.

Assume that a +24 V_{DC} will be used to power an ADAM-5000/485 and four ADAM-5017 input modules. The distance from Power supply to modules is not so big that significant line voltage drop will occur. One ADAM-5000/485 system and one ADAM-5017 module consume a maximum of 1.0 W (ADAM-5000) and 1.2 W (ADAM-5017). The total required power will equal 1 W + 4 x 1.2 W = 5.8 Watts. A power supply of +24 V_{DC} should, therefore, be able to supply a minimal current of 5.8
/ 24 = 0.25 Amps.

Small systems may be powered by using wall-mounted modular power supplies. Also when systems operate on long communication lines (>500 feet) it is often more reliable to power the systems locally with modular power supplies. These inexpensive units can easily be obtained from any electronics retail store.

The power cables should be selected according to the number of systems connected and the length of the power lines. When using a network with long cables, we advise the use of thicker wire to limit the line voltage drop. In addition to serious voltage drops, long voltage lines can also cause interference with communication wires.

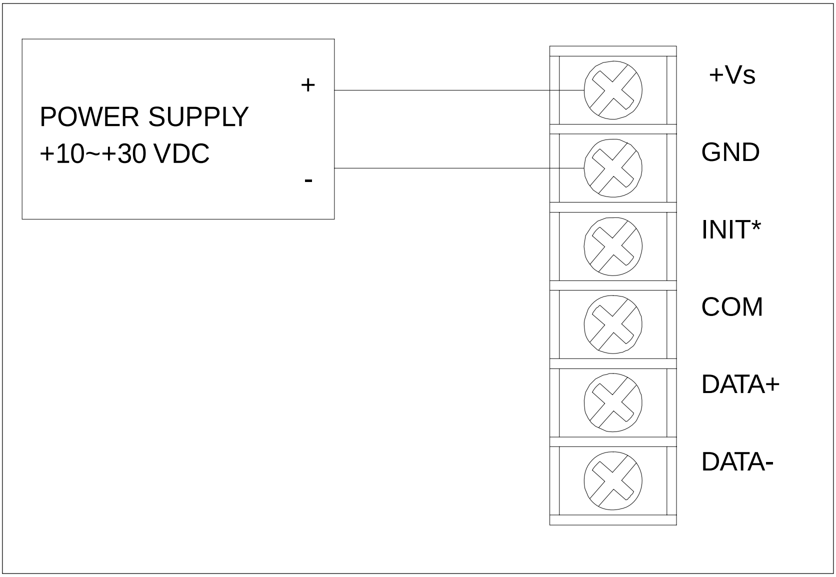


Figure A-1 *Power supply connections*

It is advisable to use standard coding colors for the power lines:

+Vs	(R)	Red
GND	(B)	Black

Communication Wiring

It is recommended that shielded, twisted-pair cables that comply with the EIA RS-485 standard be used with the ADAM-5000 network to reduce interference. Only one set of twisted-pair cables is required to transmit both DATA and RTS signals. It is advisable that the follow- ing standard colors be used for the communication lines:

DATA+	(Y)	Yellow
DATA-	(G)	Green

ADAM Utility Software

A menu-driven utility program is provided for ADAM-5000 system configuration, monitoring and calibration. It also includes a terminal emulation program that lets you easily communicate through the ADAM command set. (See Chapter 5, Utility Software)

ADAM Isolated RS-232/RS-485 Converter (optional)

When the host computer or terminal has only a RS-232 port, an ADAM Isolated RS-232/RS-485 Converter, connected to the host's RS-232 port, is required. Since this module is not addressable by the host, the baud rate must be set using a switch inside the module. The factory default setting is 9600 baud.

ADAM Repeater (optional)

When communication lines exceed 4000ft (1200 meters) or the number of ADAM-5000 systems connected is more than 32, a repeater should be connected to expand the first segment. Up to 32 repeater modules can be connected allowing connection of up to 256 ADAM-5000 systems. As with the converter module, the repeater module is not addressable by the host and the baud rate must be set by changing the switch inside the module. The factory default setting is 9600 baud.

A.2 Basic Configuration Hook-up

Before placing the system in an existing network, the system should be configured properly. Though the system is initially configured at the factory, it is recommended you check that the baud rate is set correctly.

Quick Start Example

Default Factory Settings

Baud rate: 9600

Bits/sec. Address: 01

(hexadecimal)

The basic hook-up for system configuration is show below:

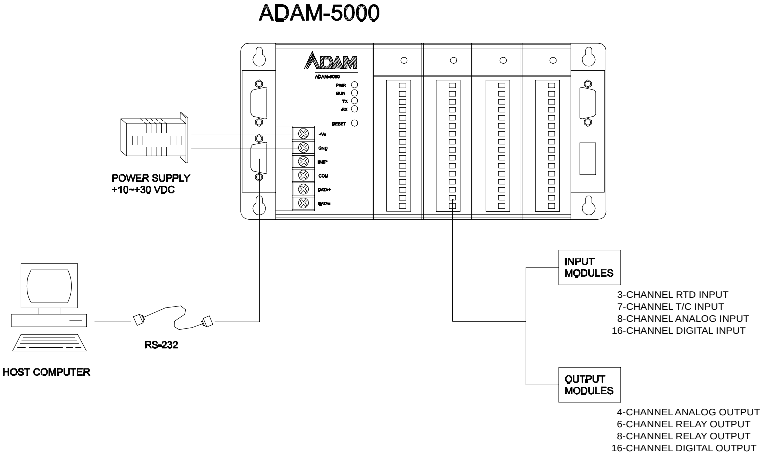


Figure A-2 *ADAM-5000 system hook-up and configuration*

The following items are required to configure an ADAM-5000 system: a personal computer with RS-232 port (baud rate sent to 9600) and the ADAM utility software.

Configuration with the ADAM Utility software

The easiest way to configure the ADAM system is by using the ADAM utility software. The easy-to-use, menu driven software will guide you through every step of the configuration process. (See Chapter 5, Utility Software).

ADAM systems can also be configured by issuing direct command from within the terminal emulation program that is included with the ADAM utility software.

The following example guides you through the setup of an analog input module. Assume that an ADAM-5018 Thermocouple Input module in slot 1 on an ADAM-5000/485 system still has its default settings (baud rate 9600 and address 01h). The system is first request- ed to send its default settings and then reconfigured.

Note: *An analog input module requires a maximum of 7 seconds to perform auto calibration and ranging after it is rebooted or powered on. During this time span, the module cannot be addressed to perform any other actions.*

Example:

Make sure that the module is properly connected as shown in Chapter 4, Figure 4-3. Power up all the connected devices, start the terminal emulation program, and issue the following command:

```
$01S1B(cr)
```

This command requests the module in slot 1 of the ADAM-5000 system at address 01h to send its configuration status

```
!010500
```

The module of the system at address 01h responds that it is configured for an input range of ±2.5 V, integration time of 50 ms (60 Hz), format = engineering units and no checksum checking or generation. To change the configuration setting of the analog input module, the following command is issued:

```
$01S1A0F00(cr)
```

$ = change configuration

01 = target module at address 01

0F = set input range to type K thermocouple

00 = set data format to engineering units, 50ms (60Hz)

(See Chapter 6, Command Set for a full description of the syntax of the configuration command for an analog input module)

When the module received the configuration command it will respond with its new address:

```
!01(cr)
```

Wait 7 seconds to let the new configuration settings take effect before issuing a new command to the module.

Note: All reconfiguration except changing of baud rate and checksum values can be done dynamically, i.e. the modules need not be reset. When changing baud rate or checksum, these changes should be made for all connected devices. After reconfiguration, all modules should be powered down and then powered up to force a reboot and let the changes take effect. See the next section for a strategy for changing baud rate and or checksum for an entire network.

A.3 Baud Rate and Checksum

ADAM-5000 systems contain an EEPROM to store configuration information and calibration constants. The ROM replaces the usual array of switches and pots required to specify baud rate, input/output range, etc. The ADAM-5000 system can be configured remotely through their communication ports, without having to physically alter pot or switch settings.

Since there is no visual indication of a system's configuration status, it is not possible to visually determine baud rate and other system settings. It might not be possible to establish communications with a system whose baud rate and address are unknown. To overcome this problem, every system has an input terminal labeled INIT*. By booting the system while connecting the INIT* terminal with the system's GND terminal, the system configuration is forced into a known state. This state is called the INIT* state.

INIT* state defaults:

Baud rate:9600
Address:00h
Checksum:disabled

Forcing the system into the INIT* state does not change any parameters in the system's EEPROM. When the system is in the INIT* state with its INIT* and GND terminal shorted, all configuration settings can be changed and the system will respond to all other commands normally.

Changing Baud Rate and Checksum

Baud rate and checksum setting have several things in common:

- They should be the same for all systems and host computer.
- Their setting can only be changed by putting a system in the INIT* state.
- A changed setting can only take effect after a system is rebooted.

To alter baud rate or checksum settings you must perform the following steps:

1. Power on all components except the ADAM-5000 system
2. Power the ADAM-5000 system ON while shorting the INIT* and GND terminals as shown in Figure A-3.

Quick Start Example

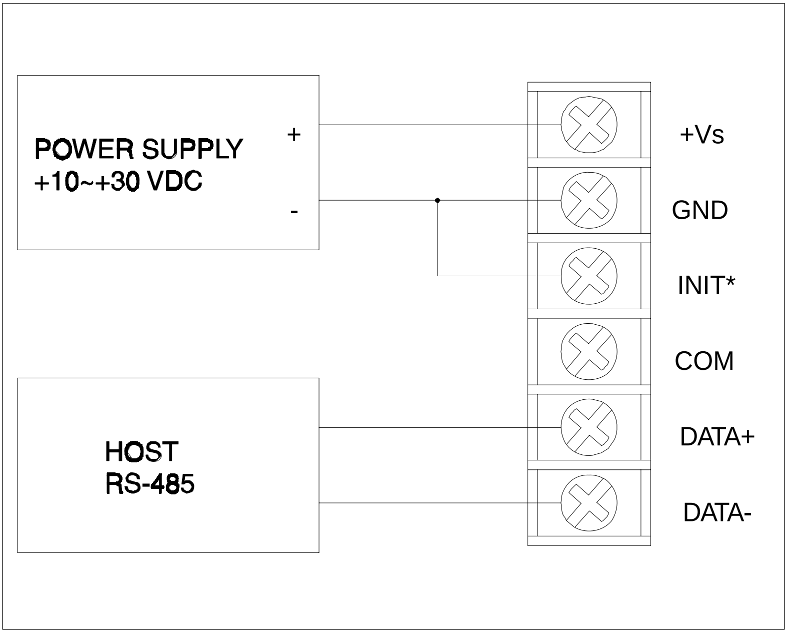


***Figure A-3** Grounding the INIT* terminal*

3. Wait at least 7 seconds to let self-calibration and ranging takeef- fect.

4. Configure the baud rate and/or checksum status.

5. Switch the power to the ADAM-5000 system OFF.

6. Remove the grounding on the INIT* terminal and power the ADAM-5000 system ON.

7. Wait at least 7 seconds to let self-calibration and ranging takeef- fect.

8. Check the settings. (If the baud rate has changed, the settings on the host computer should be changed accordingly)

A.3 A Distributed ADAM-5000 Network System Hook-up

The figure below gives an example of how multiple ADAM systems should be connected:

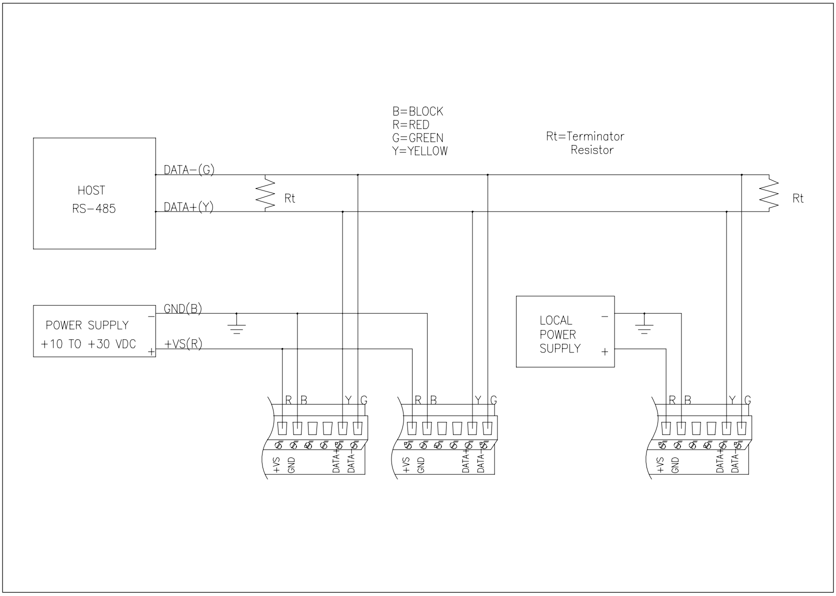


***Figure A-4** ADAM-5000 network system hook-up*

This page intentionally left blank

B

Data Formats And I/O Ranges

Data Formats and I/O Ranges

B.1 Analog Input Formats

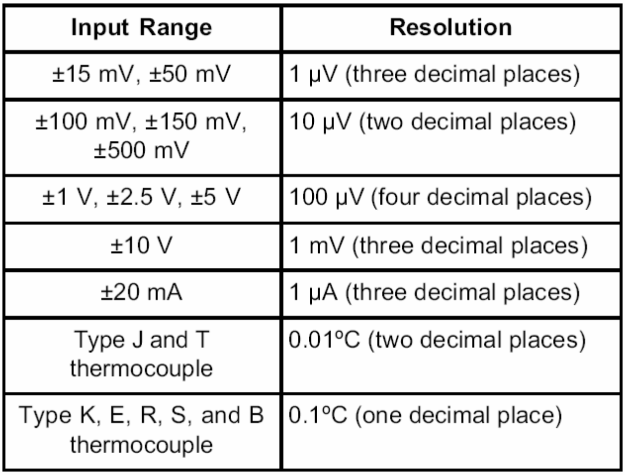
The ADAM analog input modules can be configured to transmit data to the host in Engineering Units.

Engineering Units

Data can be represented in Engineering Units by setting bits 0 and 1 of the data format/checksum/integration time parameter to 0.

This format presents data in natural units, such as degrees, volts, millivolts, and milliamps. The Engineering Units format is readily parsed by the majority of computer languages because the total data string length, including sign, digits and decimal point, does not exceed seven characters.

The data format is a plus (+) or minus (-) sign, followed by five decimal digits and a decimal point. The input range which is employed determines the resolution, or the number of decimal places used, as illustrated in the following table:



Input Range	Resolution
±15 mV, ±50 mV	1 μV (three decimal places)
±100 mV, ±150 mV, ±500 mV	10 μV (two decimal places)
±1 V, ±2.5 V, ±5 V	100 μV (four decimal places)
±10 V	1 mV (three decimal places)
±20 mA	1 μA (three decimal places)
Type J and T thermocouple	0.01°C (two decimal places)
Type K, E, R, S, and B thermocouple	0.1°C (one decimal place)

Example 1

The input value is -2.65 and the corresponding analog input module is configured for a range of ±5 V. The response to the Analog Data In command is:

```
-2.6500(cr)
```

Example 2

The input value is 305.5ºC. The analog input module is configured for a Type J thermocouple whose range is 0ºC to 760ºC. The response to the Analog Data In command is:

```
+305.50(cr)
```

Example 3

The input value is +5.653 V. The analog input module is configured for a range of ±5 V range. When the engineering units format is used, the ADAM Series analog input modules are configured so that they automatically provide an over range capability. The response to the Analog Data In command in this case is:

```
+5.6530(cr)
```

B.2 Analog Input Ranges - ADAM-5018

Module	Range Code	Input Range Description	Data Formats	+F.S.	Zero	-F.S.	Displayed Resolution	Actual Value
ADAM-5018	00h	±15 mV	Engineering Units	+15.000	±00.000	-15.000	1 µV	Reading/ 1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	01h	±50 mV	Engineering Units	+50.000	±00.000	-50.000	1 µV	Reading/ 100
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	02h	±100 mV	Engineering Units	+100.00	±000.00	-100.00	10 µV	Reading/ 100
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	03h	±500 mV	Engineering Units	+500.00	±000.00	-500.00	10 µV	Reading/ 10
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	04h	±1 V	Engineering Units	+1.0000	±0.0000	-1.0000	100 µV	Reading/ 10000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	05h	±2.5 V	Engineering Units	+2.5000	±0.0000	-2.5000	100 µV	Reading/ 10000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	06h	±20 mA	Engineering Units	+20.000	±00.000	-20.000	1 µA	Reading/ 1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	07h	Not Used						

Module	Range Code	Input Range Description	Data Formats	Maximum Specified Signal	Minimum Specified Signal	Displayed Resolution	Actual Value
ADAM-5018	0Eh	Type J Thermocouple 0°C to 760°C	Engineering Units	+760.00	+000.00	0.1°C	Reading/ 10
			% of FSR	+100.00	+000.00	0.01%	
			Two's Complement	7FFF	0000	1 LSB	
	0Fh	Type K Thermocouple 0°C to 1370°C	Engineering Units	+1370.0	+0000.0	0.1°C	Reading/ 10
			% of FSR	+100.00	+000.00	0.01%	
			Two's Complement	7FFF	0000	1 LSB	
	10h	Type T Thermocouple -100°C to 400°C	Engineering Units	+400.00	-100.00	0.1°C	Reading/ 10
			% of FSR	+100.00	-025.00	0.01%	
			Two's Complement	7FFF	E000	1 LSB	
	11h	Type E Thermocouple 0°C to 1000°C	Engineering Units	+1000.00	+0000.0	0.1°C	Reading/ 10
			% of FSR	+100.00	±000.00	0.01%	
			Two's Complement	7FFF	0000	1 LSB	
	12h	Type R Thermocouple 500°C to 1750°C	Engineering Units	+1750.0	+0500.0	0.1°C	Reading/ 10
			% of FSR	+100.00	+028.57	0.01%	
			Two's Complement	7FFF	2492	1 LSB	
	13h	Type S Thermocouple 500°C to 1750°C	Engineering Units	+1750.0	+0500.00	0.1°C	Reading/ 10
			% of FSR	+100.00	+028.57	0.01%	
			Two's Complement	7FFF	2492	1 LSB	
	14h	Type B Thermocouple 500°C to 1800°C	Engineering Units	+1800.0	+0500.0	0.1°C	Reading/ 10
			% of FSR	+100.00	+027.77	0.01%	
			Two's Complement	7FFF	2381	1 LSB	

Module	Range Code	Input Range Description	Data Formats	+F.S.	Zero	-F.S.	Displayed Resolution	Actual Value
ADAM-5018P	00h	±15 mV	Engineering Units	+15.000	±00.000	-15.000	1 µV	Reading/ 1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	01h	±50 mV	Engineering Units	+50.000	±00.000	-50.000	1 µV	Reading/ 100
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	02h	±100 mV	Engineering Units	+100.00	±000.00	-100.00	10 µV	Reading/ 100
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	03h	±500 mV	Engineering Units	+500.00	±000.00	-500.00	10 µV	Reading/ 10
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	04h	±1 V	Engineering Units	+1.0000	±0.0000	-1.0000	100 µV	Reading/ 10000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	05h	±2.5 V	Engineering Units	+2.5000	±0.0000	-2.5000	100 µV	Reading/ 10000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	06h	±20 mA	Engineering Units	+20.000	±00.000	-20.000	1 µA	Reading/ 1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	07h	4~20mA	Engineering Units	+20.000	±04.000	-	-	Reading/ 1000
			% of FSR	+100.00	±000.00	-	-	
			Two's Complement	7999	1999	-	-	

Module	Range Code	Input Range Description	Data Formats	Maximum Specified Signal	Minimum Specified Signal	Displayed Resolution	Actual Value
ADAM-5018P	0Eh	Type J Thermocouple 0°C to 760°C	Engineering Units	+760.00	+000.00	0.1°C	Reading/ 10
			% of FSR	+100.00	+000.00	0.01%	
			Two's Complement	7FFF	0000	1 LSB	
	0Fh	Type K Thermocouple 0°C to 1370°C	Engineering Units	+1370.0	+0000.0	0.1°C	Reading/ 10
			% of FSR	+100.00	+000.00	0.01%	
			Two's Complement	7FFF	0000	1 LSB	
	10h	Type T Thermocouple -100°C to 400°C	Engineering Units	+400.00	-100.00	0.1°C	Reading/ 10
			% of FSR	+100.00	-025.00	0.01%	
			Two's Complement	7FFF	E000	1 LSB	
	11h	Type E Thermocouple 0°C to 1000°C	Engineering Units	+1000.00	+0000.0	0.1°C	Reading/ 10
			% of FSR	+100.00	±000.00	0.01%	
			Two's Complement	7FFF	0000	1 LSB	
	12h	Type R Thermocouple 500°C to 1750°C	Engineering Units	+1750.0	+0500.0	0.1°C	Reading/ 10
			% of FSR	+100.00	+028.57	0.01%	
			Two's Complement	7FFF	2492	1 LSB	
	13h	Type S Thermocouple 500°C to 1750°C	Engineering Units	+1750.0	+0500.00	0.1°C	Reading/ 10
			% of FSR	+100.00	+028.57	0.01%	
			Two's Complement	7FFF	2492	1 LSB	
	14h	Type B Thermocouple 500°C to 1800°C	Engineering Units	+1800.0	+0500.0	0.1°C	Reading/ 10
			% of FSR	+100.00	+027.77	0.01%	
			Two's Complement	7FFF	2381	1 LSB	

B.3 Analog Input Ranges - ADAM-5017H

Range Code	Input Range	Data Formats	+Full Scale	Zero	-Full Scale	Displayed Resolution
00h	±10 V	Engineering	11	0	-11	2.7 mV
		Two's Comp	0FFF	0	EFFF	1
01h	0 ~ 10 V	Engineering	11	0	Don't care	2.7 mV
		Two's Comp	0FFF	0	Don't care	1
02h	±5 V	Engineering	5.5	0	-5.5	1.3 mV
		Two's Comp	0FFF	0	EFFF	1
03h	0 ~ 5 V	Engineering	5.5	0	Don't care	1.3 mV
		Two's Comp	0FFF	0	Don't care	1
04h	±2.5 V	Engineering	2.75	0	-2.75	0.67 mV
		Two's Comp	0FFF	0	EFFF	1
05h	0 ~ 2.5 V	Engineering	2.75	0	Don't care	0.67 mV
		Two's Comp	0FFF	0	Don't care	1
06h	±1 V	Engineering	1.375	0	-1.375	0.34 mV
		Two's Comp	0FFF	0	EFFF	1
07h	0 ~ 1 V	Engineering	1.375	0	Don't care	0.34 mV
		Two's Comp	0FFF	0	Don't care	1
08h	±500 mV	Engineering	687.5	0	-687.5	0.16 mV
		Two's Comp	0FFF	0	EFFF	1
09h	0 ~ 500 mV	Engineering	687.5	0	Don't care	0.16 mV
		Two's Comp	0FFF	0	Don't care	1
0ah	4 ~ 20 mA	Engineering	22	4.0	Don't care	5.3 µA
		Two's Comp	0FFF	02E9	Don't care	1
0bh	0 ~ 20 mA	Engineering	22	0	Don't care	5.3 µA
		Two's Comp	0FFF	0	Don't care	1

Note: The full scale values in this table are theoretical values for your reference; actual values will vary.

B.3.1 Analog Input Ranges - ADAM-5017UH

Range Code	Input Range	Data Formats	+Full Scale	Zero	-Full Scale	Displayed Resolution
08h	±10 V	Engineering	+10.000	+00.000	-10.000	1 mV
		Two's Comp	0FFF	0	7FFF	1
48h	0 ~ 10 V	Engineering	+10.000	+00.000	-	1 mV
		Two's Comp	0FFF	0	Don't care	1
46h	0~20mA	Engineering	+20.000	+00.000	-	1 μV
		Two's Comp	0FFF	0	Don't care	1
07h	4~20mA	Engineering	+20.000	+00.000	-	1 μV
		Two's Comp	0FFF	0	Don't care	1

Note: The full scale values in this table are theoretical values for your reference; actual values will vary.

B.4 Analog Ontput Formats

You can configure ADAM analog output modules to receive data from the host in Engineering Units.

Engineering Units

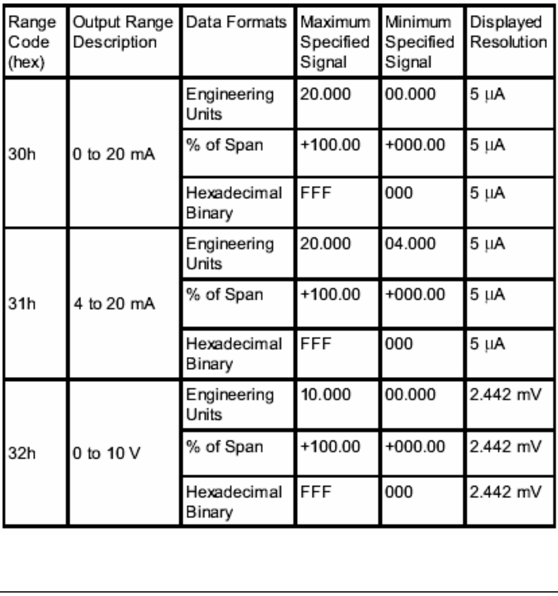
Data can be represented in engineering units by setting bits 0 and 1 of the data format/checksum/integration time parameter to 0.

This format presents data in natural units, such as milliamps. The Engineering Units format is readily parsed by the majority of computer languages as the total data string length is fixed at six characters: two decimal digits, a decimal point and three decimal digits. The resolution is 5 µA.

Example:

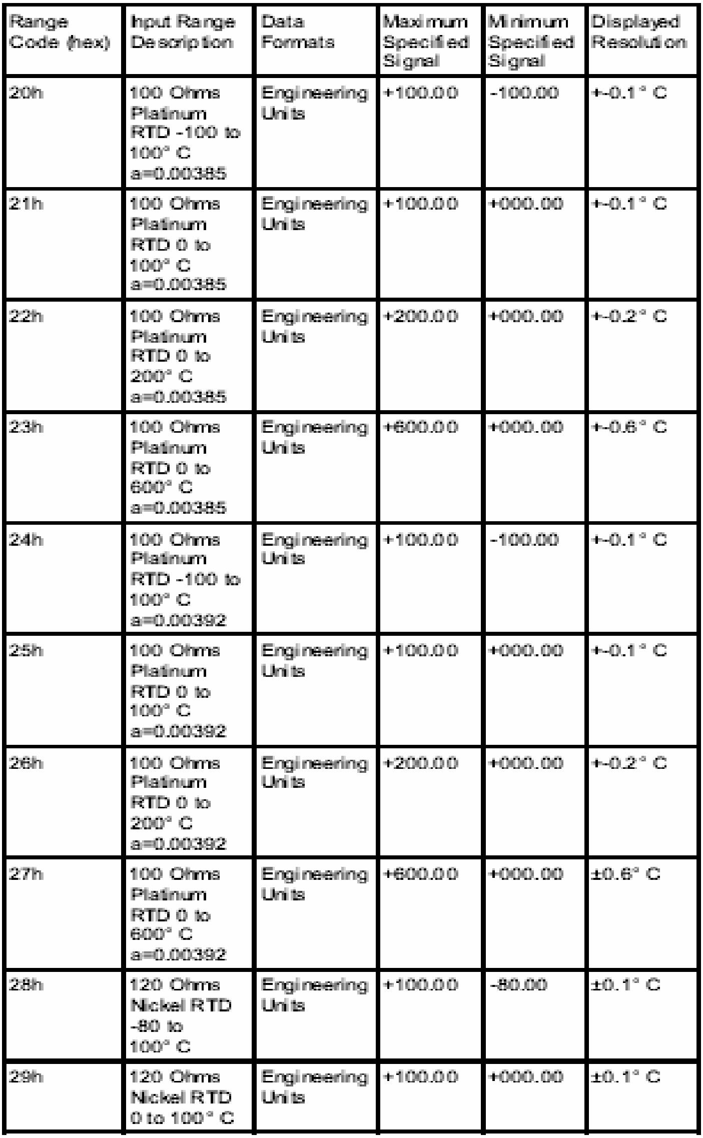
Channel 1 of the analog output module in slot 0 of an ADAM-5000 system at address 01h is configured for a 0 to 20 mA range. If the output value +4.762 mA is desired, the format of the Analog Data Out command would be #01S0C14.762<cr>

B.5 Analog Ontput Ranges



Range Code (hex)	Output Range Description	Data Formats	Maximum Specified Signal	Minimum Specified Signal	Displayed Resolution
30h	0 to 20 mA	Engineering Units	20.000	00.000	5 µA
		% of Span	+100.00	+000.00	5 µA
		Hexadecimal Binary	FFF	000	5 µA
31h	4 to 20 mA	Engineering Units	20.000	04.000	5 µA
		% of Span	+100.00	+000.00	5 µA
		Hexadecimal Binary	FFF	000	5 µA
32h	0 to 10 V	Engineering Units	10.000	00.000	2.442 mV
		% of Span	+100.00	+000.00	2.442 mV
		Hexadecimal Binary	FFF	000	2.442 mV

B.6 ADAM-5013 RTD Input Format and Ranges



Range Code (hex)	Input Range Description	Data Formats	Maximum Specified Signal	Minimum Specified Signal	Displayed Resolution
20h	100 Ohms Platinum RTD -100 to 100° C a=0.00385	Engineering Units	+100.00	-100.00	+-0.1° C
21h	100 Ohms Platinum RTD 0 to 100° C a=0.00385	Engineering Units	+100.00	+000.00	+-0.1° C
22h	100 Ohms Platinum RTD 0 to 200° C a=0.00385	Engineering Units	+200.00	+000.00	+-0.2° C
23h	100 Ohms Platinum RTD 0 to 600° C a=0.00385	Engineering Units	+600.00	+000.00	+-0.6° C
24h	100 Ohms Platinum RTD -100 to 100° C a=0.00392	Engineering Units	+100.00	-100.00	+-0.1° C
25h	100 Ohms Platinum RTD 0 to 100° C a=0.00392	Engineering Units	+100.00	+000.00	+-0.1° C
26h	100 Ohms Platinum RTD 0 to 200° C a=0.00392	Engineering Units	+200.00	+000.00	+-0.2° C
27h	100 Ohms Platinum RTD 0 to 600° C a=0.00392	Engineering Units	+600.00	+000.00	±0.6° C
28h	120 Ohms Nickel RTD -80 to 100° C	Engineering Units	+100.00	-80.00	±0.1° C
29h	120 Ohms Nickel RTD 0 to 100° C	Engineering Units	+100.00	+000.00	±0.1° C

C

RS-485 Network

RS-485 Network

EIA RS-485 is the industry's most widely used bidirectional, balanced transmission line standard. It is specifically developed for industrial multi-drop systems that should be able to transmit and receive data at high rates or over long distances.

The specifications of the EIA RS-485 protocol are as follows:

- Maximum line length per segment: 1200 meters (4000 feet)
- Throughput of 10 Mbaud and beyond -Differential transmission (balanced lines) with high resistance against noise
- Maximum 32 nodes per segment
- Bi-directional master-slave communication over a single set of twisted-pair cables
- Parallel connected nodes, true multi-drop

ADAM-5000 systems are fully isolated and use just a single set of twisted pair wires to send and receive! Since the nodes are connected in parallel they can be freely disconnected from the host without affecting the functioning of the remaining nodes. An industry stan- dard, shielded twisted pair is preferable due to the high noise ratio of the environment.

When nodes communicate through the network, no sending conflicts can occur since a simple command/response sequence is used. There is always one initiator (with no address) and many slaves (with addresses). In this case, the master is a personal computer that is connected with its serial, RS-232, port to an ADAM RS-232/RS-485 converter. The slaves are the ADAM-5000 systems. When systems are not transmitting data, they are in listen mode. The host computer initiates a command/response sequence with one of the systems. Commands normally contain the address of the module the host wants to communicate with. The system with the matching address carries out the command and sends its response to the host.

Appendix C

C.1 Basic Network Layout

Multi-drop RS-485 implies that there are two main wires in a segment. The connected systems tap from these two lines with so called drop cables. Thus all connections are parallel and connecting or disconnecting of a node doesn’t affect the network as a whole. Since ADAM-5000 systems use the RS-485 standard and an ASCII-based commands set, they can connect and communicate with all ASCII-based computers and terminals. The basic layouts that can be used for an RS-485 network are:

Daisychain

The last module of a segment is a repeater. It is directly connected to the main-wires thereby ending the first segment and starting the next segment. Up to 32 addressable systems can be daisychained . This limitation is a physical one. When using more systems per segment the IC driver current rapidly decreases, causing communication errors.   In total, the network can hold up to 256 addressable systems. The limitation on this number is the two-character hexadecimal address code that can address 256 combinations. The ADAM converter, ADAM repeaters and the host computer are non addressable units and therefore are not included in these numbers.

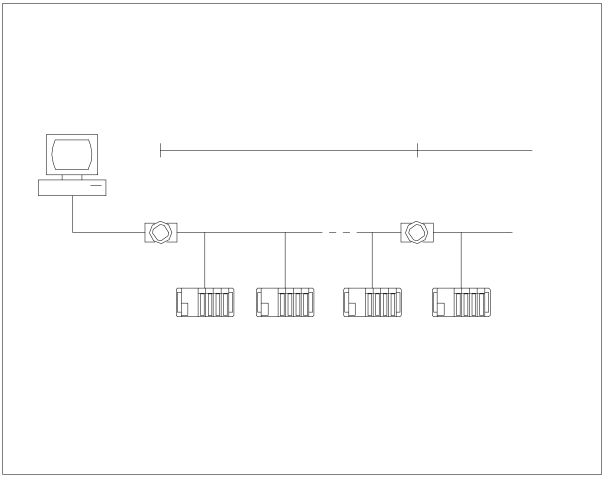


Figure C-1 *Daisychaining*

Star Layout

In this scheme the repeaters are connected to drop-down cables from the main wires of the first segment. A tree structure is the result. This scheme is not recommended when using long lines since it will cause a serious amount of signal distortion due to signal reflections in several line-endings.

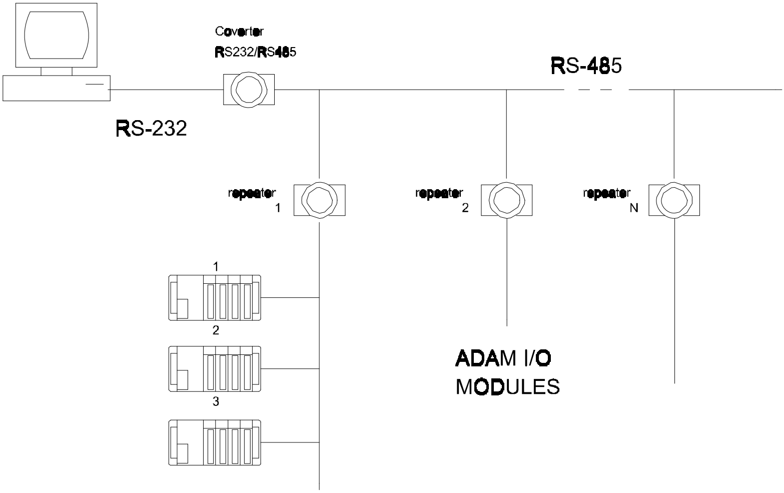


Figure C-2 *Star structure*

Random

This is a combination of daisychain and hierarchical structure.

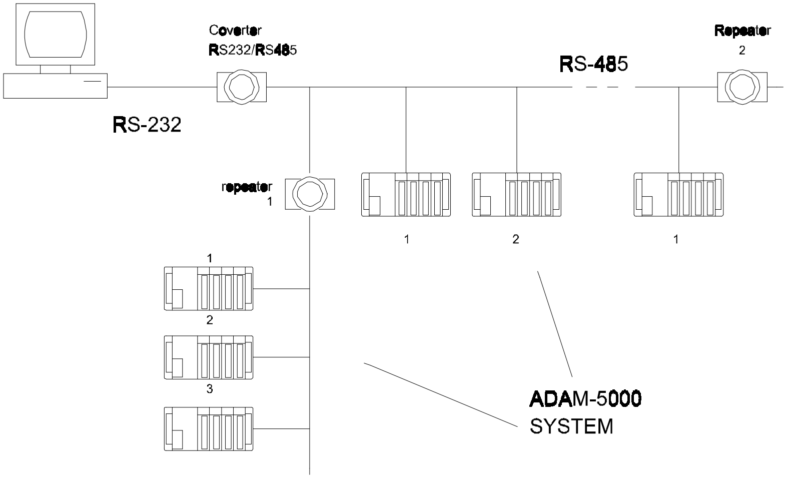


Figure C-3 Random Sstructure

Combination of an ADAM-4000 and an ADAM-5000 in a RS-498 Network

The following figure shows how to integrate ADAM-4000 and ADAM-5000 systems in a network.

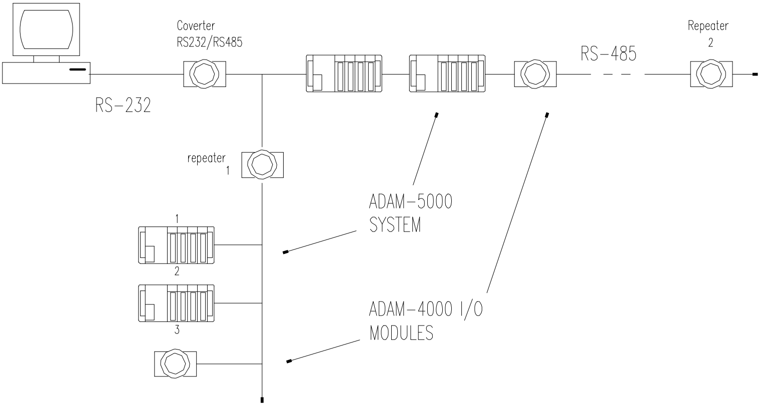


***Figure C-4** ADAM-4000 and ADAM-5000 in a network*

Note: *The speed of ADAM-4000 and ADAM-5000 in a RS-485 network should be the same.*

C.2 Line Termination

Each discontinuity in impedance causes reflections and distortion. When a impedance discontinuity occurs in the transmission line the immediate effect is signal reflection. This will lead to signal distortion. Specially at line ends this mismatch causes problems. To eliminate this discontinuity, terminate the line with a resistor.

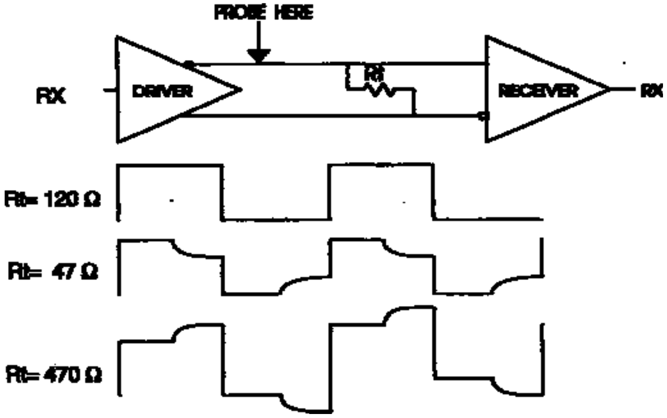


Figure C-5 *Signal distortion*

The value of the resistor should be a close as possible to the characteristic impedance of the line. Although receiver devices add some resistance to the whole of the transmission line, normally it is sufficient to the resistor impedance should equal the characteristic impedance of the line.

Example:

Each input of the receivers has a nominal input impedance of 18 kΩ feeding into a diode transistor- resistor biasing network that is equivalent to an 18 kΩ input resistor tied to a common mode voltage of
2.4 V. It is this configuration which provides the large common range of the receiver required for RS-485 systems! (See Figure C-6 below).

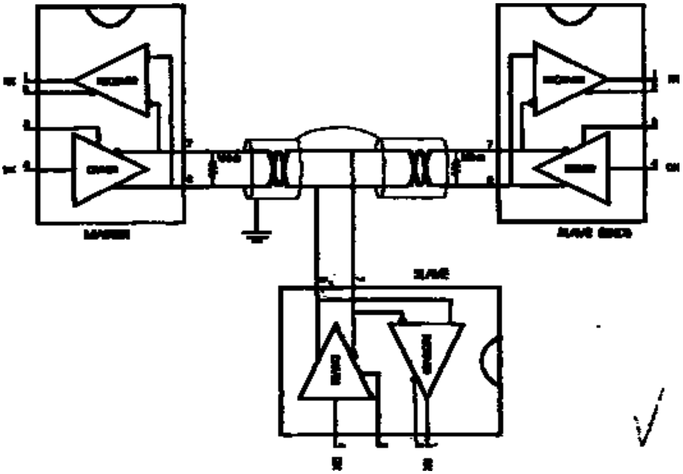


Figure C-6 *Termination resister locations*

Because each input is biased to 2.4 V, the nominal common mode voltage of balanced RS-485 systems, the 18 kΩ on the input can be taken as being in series across the input of each individual receiver. If thirty of these receivers are put closely together at the end of the transmission line, they will tend to react as thirty 36kΩ resistors in parallel with the termination resistor. The overall effective resistance will need to be close to the characteristics of the line. The effective parallel receiver resistance R_P will therefore be equal to:

$R_P = 36 \times 10^3/30 = 1200\ \Omega$

While the termination receptor R_T will equal:

$R_T = R_O / [1 - R_O/R_P]$

Thus for a line with a characteristic impedance of 100 Ω resistor

$R_T = 100/[1 - 100/1200] = 110\ \Omega$

Since this value lies within 10% of the line characteristic impedance. Thus as already stated above the line termination resistor R_T will normally equal the characteristic impedance Z_o.

The star connection causes a multitude of these discontinuities since there are several transmission lines and is therefore not recommend.

Note: *The recommend method wiring method, that causes a minimum amount of reflection, is daisy chaining where all receivers tapped from one transmission line needs only to be terminated twice.*

C.3 RS-485 Data Flow Control

The RS-485 standard uses a single pair of wires to send and receive data. This line sharing requires some method to control the direction of the data flow. RTS (Request To Send) and CTS (Clear To Send) are the most commonly used methods.

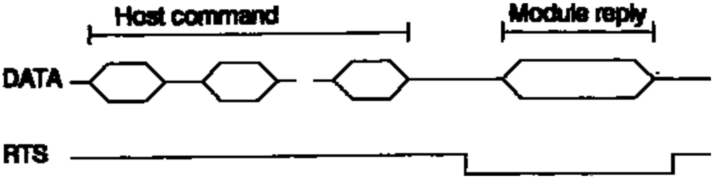


Figure C-7 *RS-485 data flow control with RTS*

Intelligent RS-485 Control

ADAM-4510 and ADAM-4520 are both equipped with an I/O circuit which can automatically sense the direction of the data flow. No handshaking with the host (like RTS, Request to Send) is necessary to receive data and forward it in the correct direction. You can use any software written for half-duplex RS-232 with an ADAM network without modification. The RS-485 control is completely transparent to the user.

This page intentionally left blank

D

How to Use the Checksum Feature

How to Use the Checksum Feature

A checksum helps you to detect errors in commands from the host to the modules, and in responses from the modules to the host. The feature adds two extra checksum characters to the command or response string, which does reduce the throughput.

D.1 Checksum Enable/Disable

To enable configuration of a module's checksum feature, its INIT* terminal should be shorted to its GND terminal, after which the module should be rebooted. The checksum feature is enabled by setting bit 6 of the data format/checksum parameter to 1. To disable the checksum, set the parameter to 0. Remember that when using the checksum feature, it should always be enabled for all connected devices including the host computer.

The checksum is represented by a 2-character ASCII hexadecimal format and is transmitted just prior to the carriage return. The checksum equals the modulo-256 (100h) sum of all the ASCII values in the command preceding the checksum. If the checksum in a command is missing or incorrect the module will not respond.

Example 1

The following is an example of an Analog Data In command and response when the checksum is enabled:

Command: #05S10C(cr)

Response: +3.56719D(cr)

The input value at the module in slot 1 of the ADAM-5000 systems at address 05h is +3.5671 V. (The date format is engineering units.) The command checksum (0Ch) is the sum of the ASCII values of the following characters: #, 0, 5, S and 1. The response checksum (9Dh) is the sum of the ASCII values of the following characters: ">" "+" "3" "4" "5" "6" "7" and "1"

Example 2

This example explains how to calculate the checksum value of a Read High alarm limit command string:

Case 1. (If the Checksum feature is **disabled**)

Command: $07S1RH(cr)

Response: !07+2.0500(cr) when the command is valid.

Case 2. (If the Checksum feature is enabled)

Command: $07S1RHA9(cr)

Response: !07+2.0500D8(cr)

where:

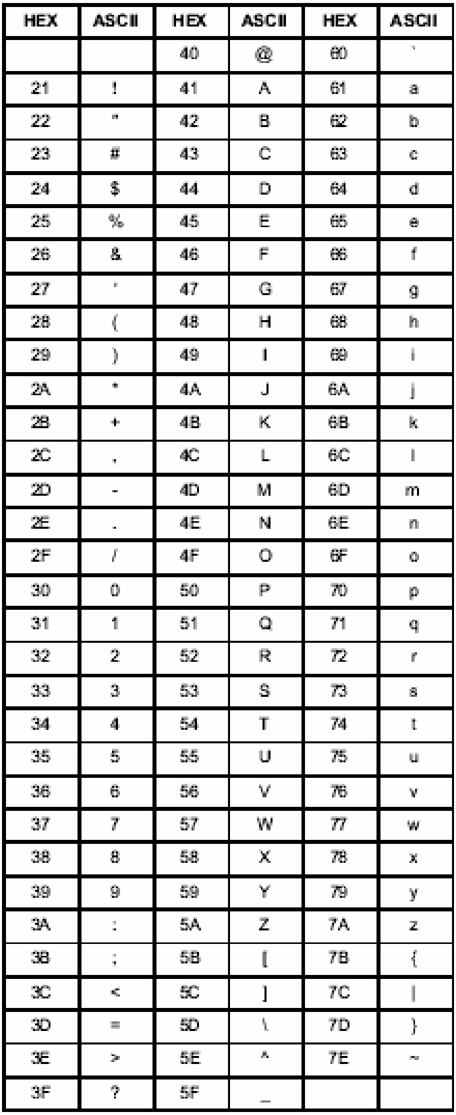
A9 represents the checksum of this command, and<R>D8 represents the checksum of the response.

The checksum of the command string is derived as shown below:

A9h = (24h+ 30h + 37h + 53h + 31h + 52h + 48h) MOD 100h

The hexadecimal ASCII codes for $, 0, 7, S, 1, R and H are 24h, 30h, 37h, 53h, 31h, 52h and 48h respectively. The sum of these ASCII codes is 1A9h. The modulo-256(100h) - of 1A9h is A9h.

Printable ASCII Characters



HEX	ASCII	HEX	ASCII	HEX	ASCII
		40	@	60	`
21	!	41	A	61	a
22	"	42	B	62	b
23	#	43	C	63	c
24	$	44	D	64	d
25	%	45	E	65	e
26	&	46	F	66	f
27	'	47	G	67	g
28	(	48	H	68	h
29	)	49	I	69	i
2A	*	4A	J	6A	j
2B	+	4B	K	6B	k
2C	,	4C	L	6C	l
2D	-	4D	M	6D	m
2E	.	4E	N	6E	n
2F	/	4F	O	6F	o
30	0	50	P	70	p
31	1	51	Q	71	q
32	2	52	R	72	r
33	3	53	S	73	s
34	4	54	T	74	t
35	5	55	U	75	u
36	6	56	V	76	v
37	7	57	W	77	w
38	8	58	X	78	x
39	9	59	Y	79	y
3A	:	5A	Z	7A	z
3B	;	5B	[	7B	{
3C	<	5C	]	7C	|
3D	=	5D	\	7D	}
3E	>	5E	^	7E	~
3F	?	5F	_		

E

ADAM-4000/5000 System Grounding Installation

ADAM-4000/5000 System Grounding Installation

E.1 Power Supplies For relevant wiring issues, please refer to the following scheme:

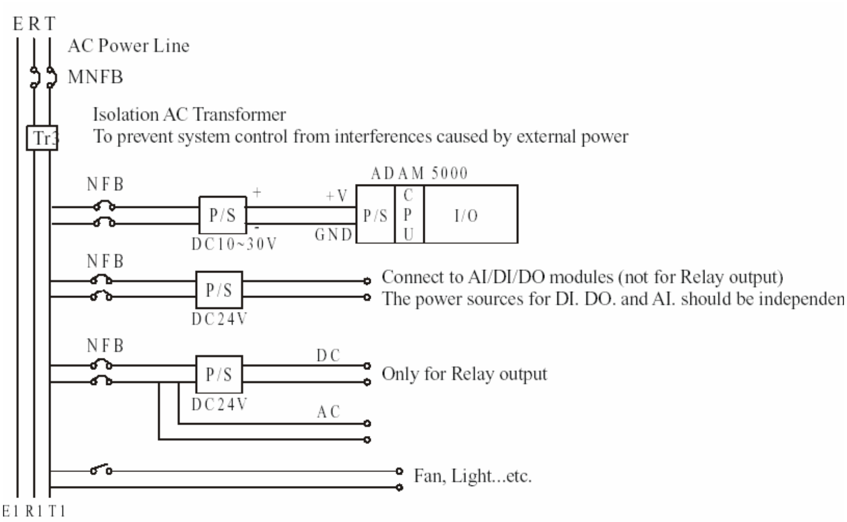

E R T
AC Power Line
MNFB
Isolation AC Transformer
To prevent system control from interferences caused by external power
Tr
NFB
P/S
+
-
DC10~30V
+V
GND
ADAM 5000
P/S
C P U
I/O
NFB
P/S
DC24V
Connect to AI/DI/DO modules (not for Relay output)
The power sources for DI. DO. and AI. should be independen
NFB
P/S
DC24V
DC
Only for Relay output
AC
Fan, Light...etc.
E1 R1 T1

Figure E-1 : *Grounding Scheme*

E.2 Grounding Scheme

The outer case for the module is made of iron and fitted with a fan and convection holes with filters.

If possible, wiring should be connected to the module through an external terminal block (T/B) to avoid external wires directly getting into the inside of the module. Its advantages are (1) a clear demarcation for external/internal wiring responsibility, (2) wire numbering can be arranged in an explicit and concise manner, and (3) an easy diagnostics for the wiring problems and a more aesthetical layout plan.

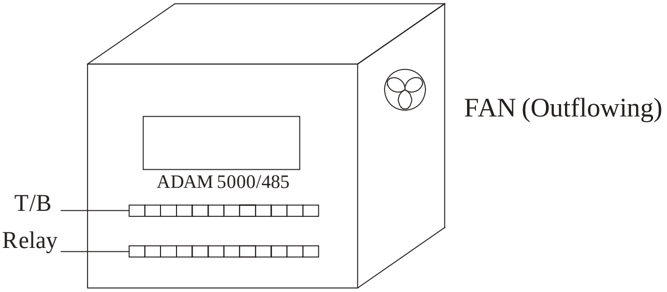


***Figure E-2 :** External Terminal Block Reference*

E.3 External DI,DO,AI,AO Wiring Reference

The common end of some D.I. and D.O. modules is connected with the GND of ADAM-5000/4000 system. Therefore, the common end of external DI and DO signal wiring should not be grounded with those on-site machineries.

Within an environment that is subject to multiple interferences, it is advised that a higher voltage level, e.g. a voltage above 12 V_{DC}, should be used to ward off possible interferences.

The signal wire for AI and AO must be of a shielded type, i.e. with surrounding copper mesh and aluminum foil for proper shielding. For the specification of the signal wires, please refer to User's Manual.

E.4 Requirements for RS-485 signal wires

Use RS-485 twisted-pair as signal wire. The quality of signal transmission can be improved in proportion with the number of twists per foot of the wire. If the wire has more twists per foot, the signal quality could be better.

Twisted -pair wire compliant with EIA-422 or EIA-485 standards, which contains 24AWG thin copper conductor with copper mesh and aluminum foil for shielding.

The shielding material of the wires should only be grounded on one end as illustrated in the following diagram. This is to avoid ground loop.

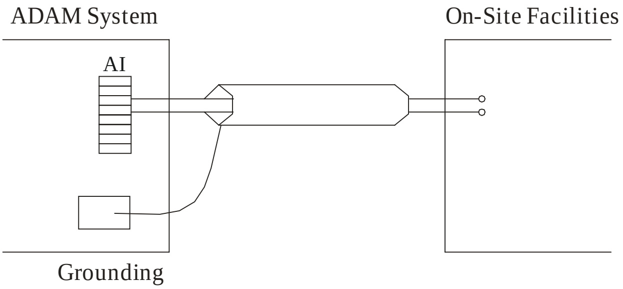


Figure E-3 : *Grounding for on-site facilities and ADAM-5000/4000 Systems*

Since shielded twisted-pair has been adopted for signal wires, only DATA+ and DATA- of ADAM-5000 system should be connected. And the shielding materials should be treated in the same manner as with AI and AO signal wires, i.e. it should be connected to Ground on only one end through the COM port on computer or on ADAM-5000 system such as illustrated in the following diagram:

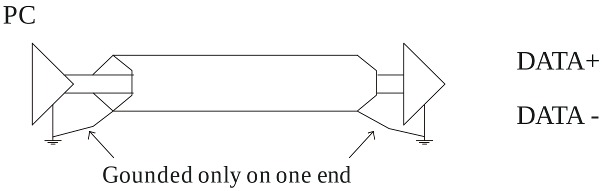


Figure E-4 : *Grounding for signal wires*

E.5 Grounding reference (Grounding bar for the factory environment should have a standard resistance below 5 Ω)

Since ADAM-4000 / 5000 system comes with a plastic outer case with DC power supply, its grounding procedure should be done according to the following points:

Power supply : The E terminal of the external power supply should be connected with the panel.

The outer case of panel should be fixed with two grounding bus. Connect the ground of power (E-terminal) to the grounding bus with shortest path. Use single contact for connection.

Another grounding bus is for connection with AI and AO shielded signal wires. While AI and AO signal wires enter inside the panel, the shielding materials is stripped off and its copper mesh should be entangled together (There is no effect leaving alone any single wire strand). Connect to grounding bus in the shortest path, and then connect the two grounding buses in a way such as the following illustration: (Please note that wire length should not be too long, otherwise it will compromise the quality of the twisted- pair wires)

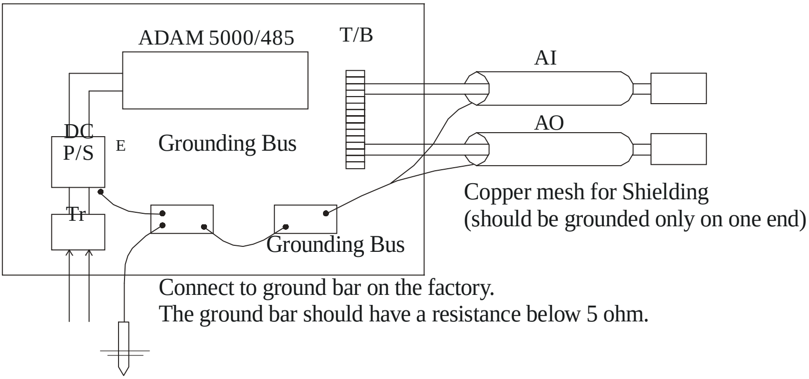

ADAM 5000/485
T/B
AI
AO
DC
P/S
E
Grounding Bus
Tr
Grounding Bus
Copper mesh for Shielding
(should be grounded only on one end)
Connect to ground bar on the factory.
The ground bar should have a resistance below 5 ohm.

Figure E-5 : *Grounding Reference*

E.6 Some Suggestions on Wiring Layout

Since communication is carried through high-frequency signals, it is advisable that the wiring layout should be paid due attention to. Any wire should best remain as a single integral wire. Nevertheless, if you should need another wire for extended connection, it is suggested that you use soldering iron to connect the disparate wires together. The parts of copper mesh should be soldered together too.

Generally, factories will layout the power lines, control lines, signal lines and communication lines within separate conduits. Since communication lines and signal lines are most susceptible to interference, you should consider avoid laying them parallel with any power line in close distance. Nevertheless, if they should remain parallel with the power line, just keep a proper distance between them. Basically, an AC current up to 2A should require a distance of 50 cm. The bigger the current or voltage, the longer the distance is required.

For communication lines and AI/O signal lines, it is suggested that they should be carried within Zinc-gilded tube for crush resistance. Meanwhile, one end of the zinc-gilded tube should be connected to factory facilities and grounded together.

While planning your wire layouts, you should consider layouts that can save wire length.

F

Grounding Reference

,

Field Grounding and Shielding Application

Overview

Unfortunately, it’s impossible to finish the system integration task at a time. We always meet some troubles in field. Such as communication network or system isn’t stable, noise influence, and equipment is damaged or hungs up by thunders. However, the most possible issue is just the improper wiring; ie, grounding and shieldinF. As you know the 80/20 rule in our life: we spend 20% time for 80% works, but 80% time for left 20% works. So to system integration, we paid 20% cost for Wire / Cable and 80% cost for Equipment. However, 80% reliability depends on Grounding and ShieldinF. In a word, we just need to pay 20% investment and work on those two issues to get a high reliable system.

This application note will bring you some concepts about field grounding and shieldinF. Below topics will be illustrated in following pages.

1. Grounding

   1.1 The ‘Earth’ for reference

   1.2 The ‘Frame Ground’ and ‘Grounding Bar’

   1.3 Normal Mode and Common Mode

   1.4 Wire impedance

   1.5 Single Point Grounding

2. Shielding

   2.1 Cable Shield

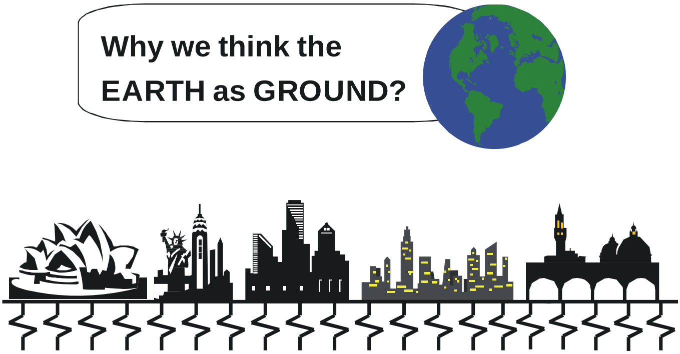
   2.2 System Shielding

3. Noise Reduction Techniques

4. Check Point List

F.1 Grounding

1.1 The Earth for reference



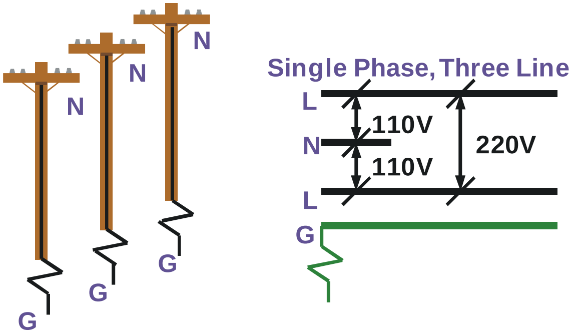
As you know that the EARTH can t be conductive indeed. But those parallel resistors make the EARTH as a single point and just for reference.

Figure F-1 : *Think the EARTH as GROUND*

- Why we think the EARTH as GROUND?

As you know that the EARTH can not be conductive indeed. But all buildings base on EARTH. Steels, concretion and relational cables such as Lighting Arrester and power system were connected to EARTH. Think them as resistors, then those infinite parallel resistors make the EARTH as a single point and just for reference.

1.2 The Frame Ground and Grounding Bar



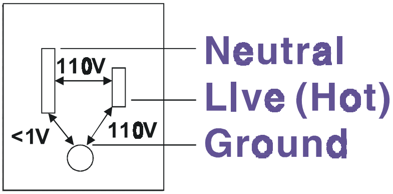
~~Neutral~~ is the physical cable from Generator.
~~Ground~~ is the local physical cable that connected to ~~Ground Bar~~.

Figure F-2 : *Grounding Bar*

According to previous description, the grounding is the most important issue for our system. Just like ‘Frame Ground’ of the computer, this signal offers a reference point of the electronic circuit inside the computer. When we want to communicate with this computer, not only ‘signal ground’, but also ‘frame ground’ should be connected to make a reference point of each other’s electronic circuit. Generally speaking, it’s necessary to build a individual grounding bar for each system, such as computer networks, telecommunication networks, power system, . . . , etc. Those individual grounding bars not only provide the individual reference point, but also make the earth as a real ground!

1.3The Frame Ground and Grounding Bar

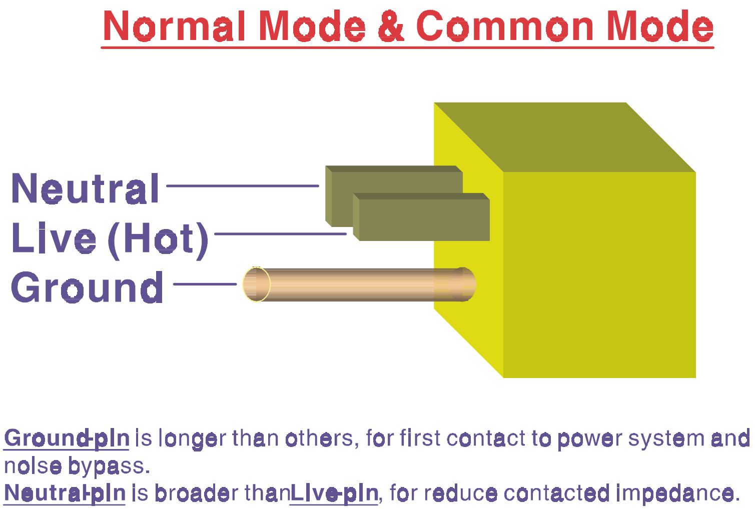
Normal Mode & Common Mode



Normal Mode: refers to defects occurring between the live and neutral conductors. Normal mode is sometimes abbreviated as NM, or L-N for live-to-neutral.
Common Mode: refers to defects occurring between either conductor and ground. It is sometimes abbreviated as CM, or N-G for neutral-to-ground.

Figure F-3 : *Normal mode and Common mode*

Have you ever tried to measure the voltage between 'Hot' and concrete floor, or measure the voltage between 'Neutral' and concrete floor? You will get nonsense value with above testinF. 'Hot' and 'Neutral' were just a relational signal, so you will get the AC110V or AC220V by measure those two signal. Normal mode and common mode just show you that the 'frame ground' is the most important reference signal for all the systems and equipments.



Normal Mode & Common Mode

Ground-pin is longer than others, for first contact to power system and noise bypass.
Neutral-pin is broader than**Live-pin**, for reduce contacted impedance.

Figure F-4 : Normal mode and Common mode

- Ground-pin is longer than others, for first contact to power system and noise bypass.
- Neutral-pin is broader than Live-pin, for reduce contact impedance.

1.4 Wire impedance

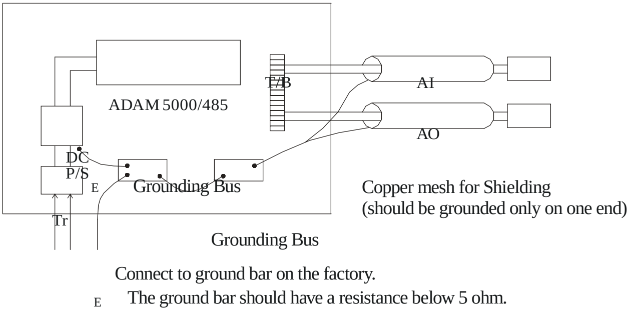
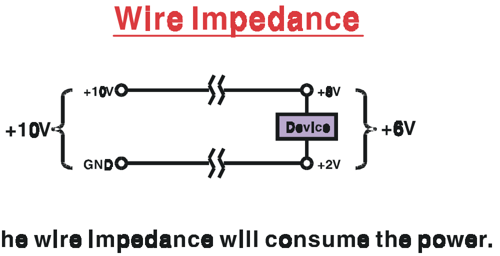

ADAM 5000/485
T/B
AI
AO
DC
P/S
E
Grounding Bus
Tr
Copper mesh for Shielding
(should be grounded only on one end)
Grounding Bus
Connect to ground bar on the factory.
E
The ground bar should have a resistance below 5 ohm.

***Figure F-5 :** The purpose of high voltage transmission*

- What’s the purpose of high voltage transmission?

We can see the high voltage tower stand at suburban. The power plant raises the voltage while generating the power, then downs the voltage when transmits the power to power station. What’s the purpose of high voltage transmission do you think? According to the energy formula, P = V * I, so the current will be reduced while raising the voltage. Besides, as you know that each cable has the wire impedance. So, referring to Ohm rule (V = I * R), this decreased current makes the low power consumption. So the high voltage transmission just for reducing the power consumption.


Wire Impedance

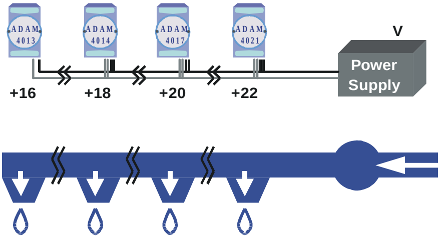
The wire impedance will consume the power.

***Figure F-6 :** Wire impedance*

Above diagram just shows you that the wire impedance will consume the power.

1.5 Single Point Grounding

Single Point Grounding



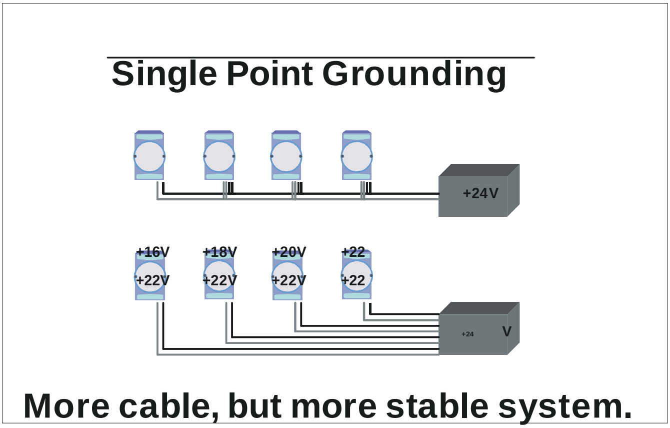
Those devices will influence each other with swiftly load change.

Figure F-7 : *Single point groundF.(1)*

- What's Single Point Grounding?

Maybe you had some displease experiences just like take hot water shower in Winter. When someone turns on another hot water hydrant near the Heater, you'll be impressed with the cold water!

The bottom diagram of above figure just shows that those devices will influence each other with swiftly load change. For example, normally we turn on all the four hydrants for testinF. When you close the hydrant 3 and hydrant 4, the other two hydrants will get a more flow. In other words, the hydrant can not keep a constant flow rate.



Single Point Grounding

More cable, but more stable system.

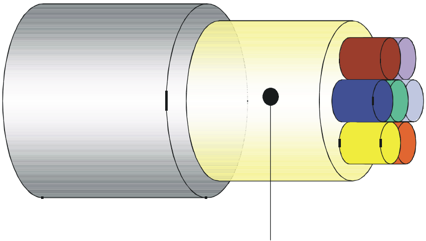
Figure F-8 : *Single point groundinF.(2)*

Above diagram shows you that single point grounding system will be a more stable system. Actually, when you use the thin cable powering those devices, the end device will get lower power. The thin cable will consume the energy.

F.2 Shielding

2.1 Cable Shield

Single Isolated Cable



Use ***Aluminum foil*** to cover those wires, for isolating the external noise.

Figure F-9 : *Single isolated cable*

- Single isolated cable

Above diagram shows you the structure of the isolated cable. You can see the isolated layer spiraling Aluminum foil to cover those wires. This spiraled structure makes an isolated layer for isolating the cables from the external noise.

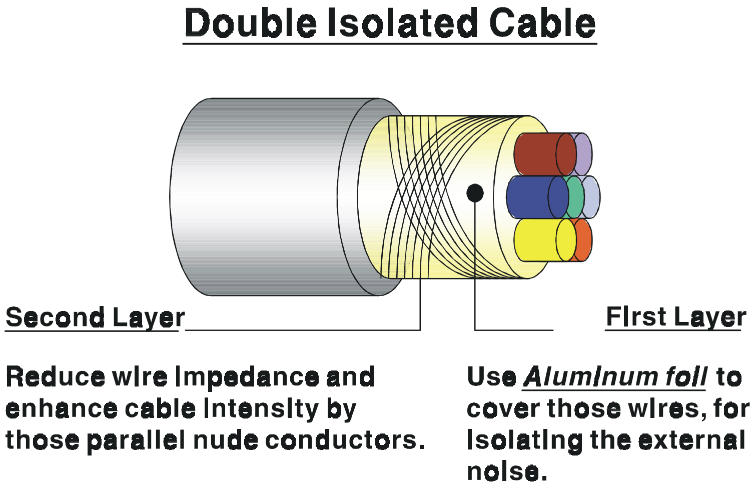

Double Isolated Cable
Second Layer
First Layer
Reduce wire Impedance and enhance cable Intensity by those parallel nude conductors.
Use *Aluminum foil* to cover those wires, for Isolating the external noise.

Figure F-10 : *Double isolated cable*

- Double isolated cable

You can see the double isolated cable structure as figure 10. The first isolated layer spiraling Aluminum foil covers those wires. The second isolated layer spiraling and crossing several nude conductors cover the first layer shielding and those wires. This spiraled structure makes an isolated layer for isolating those external noise.

Besides, following tips just for your reference.

- The shield of cable can’t be used for signal ground.

The shield is just designed for adhering noise, so the environment noise will couple and interfere your system when you use the shield as signal ground.

- The density of shield is the higher the better, especially for commu- nication network.
- Use double isolated cable for communication network / AI / AO.
- Both sides of shields should be connected to their frame while inside the device. (for EMI consideration)
- Don’t strip off too long of plastic cover for solderinF.

2.2 System Shielding

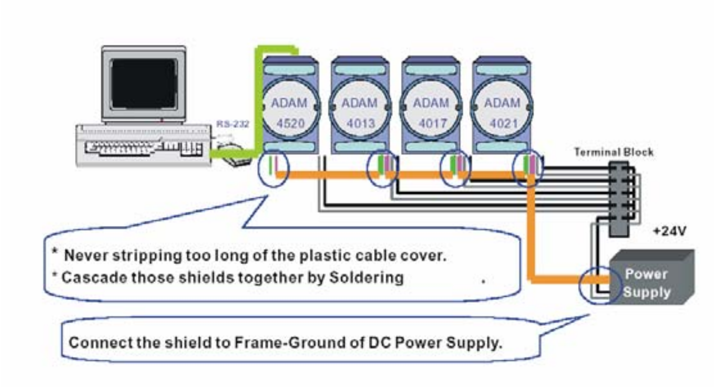

RS-232
ADAM 4520
ADAM 4013
ADAM 4017
ADAM 4021
Terminal Block
+24V
Power Supply
* Never stripping too long of the plastic cable cover.
* Cascade those shields together by Soldering
Connect the shield to Frame-Ground of DC Power Supply.

Figure F-11 :*System Shielding*

- Never stripping too long of the plastic cable cover. Otherwise, this improper status will destroy the characteristic of the Shielded-Twisted-Pair cable. Besides, those nude wires are easy to adhere the noise.

- Cascade those shields together by “Soldering”. Please refer to following page for further detail explanation.

- Connect the shield to Frame-Ground of DC power supply to force those adhered noise flow to the ‘frame ground’ of the DC power supply. (The ‘frame ground’ of the DC power supply should be connected to the system ground)

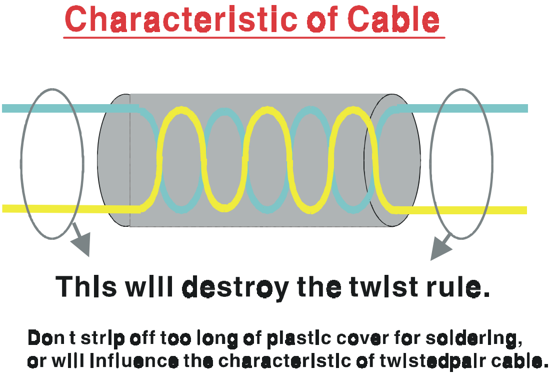
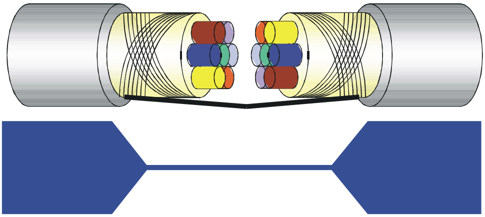

Characteristic of Cable
This will destroy the twist rule.
Don t strip off too long of plastic cover for soldering,
or will influence the characteristic of twistedpair cable.

Figure F-12 :*The characteristic of the cable*

- The characteristic of the cable

Don’t strip off too long of plastic cover for solderinF. Otherwise will influence the characteristic of the Shielded-Twisted-Pair cable, and will make an easy way to adhere noise.

System Shielding



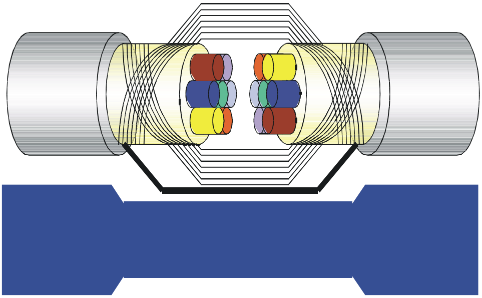
A difficult way for signal.

Figure F-13 : *System Shielding(1)*

- Shield connection (1)

When you want to visit somewhere, you must like to find out an easiest way to achieve your goal, aren’t you? So as electronic circuit, all signals use the easiest way. If we connected those two cables just with few wires, it is a difficult way for signal. So the noise will try to find out another path for easier way for flow.

System Shielding



A more easy way for signal.

Figure F-14 : *System Shielding(2)*

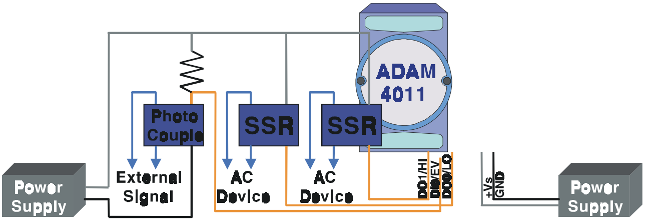
- Shield connection (2)

Above diagram shows you that the fill soldering just makes a easier way for the signal.

F.3 Noise Reduction Techniques

- Enclose noise sources in shield enclosures.
- Place sensitive equipment in shielded enclosure and away from computer equipment.
- Use separate grounds between noise sources and signals.
- Keep ground/signal leads as short as possible.
- Use Twisted and Shielded signal leads.
- Ground shields on one end ONLY while the reference grounds are not the same.
- It's almost communication problem while system unstable.
- Add another Grounding Bar if necessary.
- The diameter of power apply cable must be over 2.0 mm2.
- Independent grounding is needed for A/I, A/O, and communication network while using the jumper box.
- Use noise reduction filters if necessary. (TVS, etc)
- You can also refer to FIPS 94 Standard. FIPS 94 recommends that the computer system should be placed closer to its power source to eliminate load-induced common mode noise.

Noise Reduction Techniques



eparate Load and Devlce power.

ascade ampllfy/lsolatlon clrcult before I/O channel.

Figure F-15 : *Noise Reduction Techniques*

F.4 Check Point List

- Follow the single point grounding rule?
- Normal mode and common mode voltage?
- Separate the DC and AC ground?
- Reject the noise factor?
- The shield is connected suitable?
- The diameter of wire thick enough?
- How about the soldering of connection?
- The terminal screw tightly?

ADAM-5000 I/O Modbus Mapping Table

G

The model list of ADAM-5000 I/O series support Modbus protocol

G.1 The ADAM-5000 series main system's DIP Switch Setting:

- 5000/485:
  - #8 of DIP switch ON-> not fixed vaila address
  - #8 of DIP switch OFF-> fixed vaila address

- 5000/E: Same as 5000/485

- 5510(simu5000):

  - #5 of DIP swith ON->Modbus protocol
  - #5 of DIP swith OFF->Advantech protocol
  - Only fixed Modbus address vailable

(Note:

- The fixed Modbus address rule follows the Adam-5000/TCP Modbus address rule
- The start address must be a number of series number(1, 9, 17, …, n*8+1)

)

G.2 Modbus Commands for 5000 Series

Set COM port config.	#aaOUrff(cr) >aa(cr) OU: cmd. name r: reserved(any value) ff: Bit 1, 0 (Stop Bits) 00: 1 Stop Bits 01: 1.5 Stop Bits 10: 2 Stop Bits Bit 4, 3, 2 (Data Bits) 000: 5 001: 6 010: 7 011: 8 Bit 7, 6, 5 (Parity) 000: None 001: Even 010: Odd	
Read COM port config.	$aaOU(cr) >aarff(cr)	
Read Modbus slot address setting	$aaPNssFF(cr) >aaxxxxy (cr) PN: command name ss: slot xxxx: modbus start address (0x0001~0x2710=>1~10000) or (0x2711~0x4E20=>10001~20000) or (0x7531~0x9C40=>30001~40000) or (0x9C41~0xC350=>40001~50000) y: Length Index Idx Len 0 0 1 1 2 2 3 4 4 8 5 16 6 32 7 64	

Write Modbus slot address setting	#aaPNssFFxxxxy (cr) >aa(cr)	

Finish Modbus address setting	#aaPD(cr)
	>aa(cr)	

Read Modbus addressing mode	$aaPD(cr) >aaS(cr)	S: 1=>adjustable modbus address 0=> fixed modbus address

G.3 Modbus Address Mapping

	ADDR 4X	Item	Attribute	
	410001	Module Name 1	R	0x5485:5000/485 0x500E:5000/E
	410002	Module Name 2	R	
	410003	Versoin 1	R	0xv2 0x72
	410004	Versoin 2	R	0x00 0x00
	410005	Slot0, 1 Module Name	R	0x18 0x10(5024, 5050)
	410006	Slot2, 3 Module Name	R	0x01 (Ext.)
	410007	Slot4, 5 Module Name	R	
	410008	Slot6, 7 Module Name	R	
	410009	Slot0, 1 Ext. Module Name	R	
	410010	Slot2, 3 Ext. Module Name	R	
	410011	Slot4, 5 Ext. Module Name	R	
	410012	Slot6, 7 Ext. Module Name	R	
	410013	Modbus Addr. Mode	R	0x0001:Fixed modbus address 0x0000: Flexible modbus address
	410014~ 410020	Reserved		
	410021	Slot0 Start Addr.	R	40001
	410022	Slot0 End Addr.	R	40008
	410023	Slot1 Start Addr.	R	
	410024	Slot1 End Addr.	R	
	410025	Slot2 Start Addr.	R	
	410026	Slot2 End Addr.	R	
	410027	Slot3 Start Addr.	R	
	410028	Slot3 End Addr.	R	
	410029	Slot4 Start Addr.	R	

	410030	Slot4 End Addr.	R	
	410031	Slot5 Start Addr.	R	
	410032	Slot5 End Addr.	R	
	410033	Slot6 Start Addr.	R	
	410034	Slot6 End Addr.	R	
	410035	Slot7 Start Addr.	R	
	410036	Slot7 End Addr.	R	
	410037	Slot8 Start Addr.	R	
	410038	Slot8 End Addr.	R	
	410039~410056	Reserved		
	410101	Slot0 Lo Alarm Flag	R	
	410102	Slot0 Hi Alarm Flag	R	
	410103	Slot1 Lo Alarm Flag	R	
	410104	Slot1 Hi Alarm Flag	R	
	410105	Slot2 Lo Alarm Flag	R	
	410106	Slot2 Hi Alarm Flag	R	
	410107	Slot3 Lo Alarm Flag	R	
	410108	Slot3 Hi Alarm Flag	R	
	410109	Slot4 Lo Alarm Flag	R	
	410110	Slot4 Hi Alarm Flag	R	
	410111	Slot5 Lo Alarm Flag	R	
	410112	Slot5 Hi Alarm Flag	R	
	410113	Slot6 Lo Alarm Flag	R	
	410114	Slot6 Hi Alarm Flag	R	
	410115	Slot7 Lo Alarm Flag	R	
	410116	Slot7 Hi Alarm Flag	R	

	410117~ 410150	Reserved		
Control & Status Flag	410151~ 410152	Slot0 Control & Status Flag	R/W	
	410153~ 410154	Slot1 Control & Status Flag	R/W	
	410155~ 410156	Slot2 Control & Status Flag	R/W	
	410157~ 410158	Slot3 Control & Status Flag	R/W	
	410159~ 410160	Slot4 Control & Status Flag	R/W	
	410161~ 410162	Slot5 Control & Status Flag	R/W	
	410163~ 410164	Slot6 Control & Status Flag	R/W	
	410165~ 410166	Slot7 Control & Status Flag	R/W	
	410201~ 410208	Slot0 Ch0~Ch15 Type Code	R	0x01 0x02 (0~10V, +/-5V)
	410209~ 410216	Slot1 Ch0~Ch15 Type Code	R	
	410217~ 410224	Slot2 Ch0~Ch15 Type Code	R	
	410233~ 410240	Slot4 Ch0~Ch15 Type Code	R	
	410241~ 410248	Slot5 Ch0~Ch15 Type Code	R	
	410249~ 410256	Slot6 Ch0~Ch15 Type Code	R	
	410257~ 410264	Slot7 Ch0~Ch15 Type Code	R	

G.5 Address mapping of ADAM-5080 For ADAM-5000 ModBus

- Slot Start Address :S(word)

	OFFSET	bit	
		HIGH BYTE	LOW BYTE
CH0	S+0	CH0 counter value (Long Word)	
	S+1		
CH1	S+2	CH1 counter value(Long Word)	
	S+3		
CH2	S+4	CH2 counter value(Long Word)	
	S+5		
CH3	S+6	CH3 counter value(Long Word)	
	S+7		
CH0 Status flg	S+8	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
		reserve	Bit 0 :Counter start/stop Bit 1:Counter Reset Bit 2 :Counter Over Flow Flag
CH1 Status flg	S+9	As Above	As Above
CH2 Status flg	S+10	As Above	As Above
CH3 Status flg	S+11	As Above	As Above
	S+12~ S+15	RESERVE	

Nots:
1st bit: Default ON “1”, available to set ON/OFF to start/stop counting.
2nd bit: Normal OFF “0”, only accept a pulse ON signal to clear the counter and only available when 1st bit is OFF.
3rd bit: Normal OFF “0”, only turning ON “1” when counter overflow. Users can write “0”to clear the overflow flag. tuig

priority level
1st bit > 2nd bit > 3rd bit