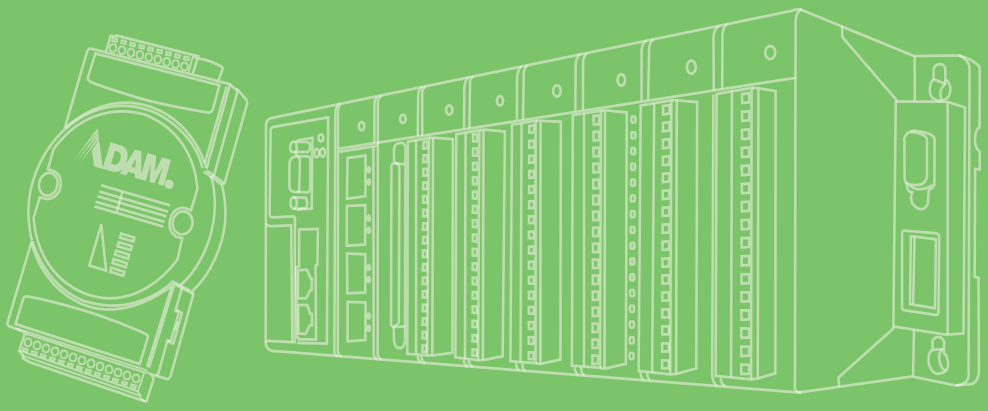
User Manual



ADAM-5000/TCP Series

Distributed DA&C System Based on Ethernet

ADVANTECH

Enabling an Intelligent Planet

Copyright

Acknowledgements

IBM and PC are trademarks of International Business Machines Corporation.

All other product names or trademarks are properties of their respective owners.

Product Warranty (2 years)

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

This warranty does not apply to any products which have been repaired or altered by persons other than repair personnel authorized by Advantech, or which have been subject to misuse, abuse, accident or improper installation. Advantech assumes no liability under the terms of this warranty as a consequence of such events.

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 is 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. (For example, CPU speed, Advantech products used, other hardware and software used, etc.) Note anything abnormal and list any onscreen 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, obtain an RMA (return merchandize authorization) number from your dealer. This allows us to process your return more quickly.
4. Carefully pack the defective product, a fully-completed Repair and Replacement Order Card and a photocopy proof of purchase date (such as your sales receipt) in a shippable container. A product returned without proof of the purchase date 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.

Part No. 2003M50020 Edition 1

December 2020

Declaration of Conformity

CE

This product has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Please contact your local supplier for ordering information.

FCC Class A

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

FM

This equipment has passed the FM certification. According to the National Fire Protection Association, work sites are classified into different classes, divisions and groups, based on hazard considerations. This equipment is compliant with the specifications of Class I, Division 2, Groups A, B, C and D indoor hazards.

Technical Support and Assistance

1. Visit the Advantech web site at www.advantech.com/support where you can find the latest information about the product.
2. Contact your distributor, sales representative, or Advantech's customer service center for technical support if you need additional assistance. Please have the following information ready before you call:
   - Product name and serial number
   - Description of your peripheral attachments
   - Description of your software (OS, version, application software, etc.)
   - A complete description of the problem
   - The exact wording of any error messages

Safety Precaution - Static Electricity

Follow these simple precautions to protect yourself from harm and the products from damage.

- To avoid electrical shock, always disconnect the power from your PC chassis before you work on it. Don't touch any components on the CPU card or other cards while the PC is on.

Disconnect power before making any configuration changes. The sudden rush of power as you connect a jumper or install a card may damage sensitive electronic components.

Safety Instructions

1. Read these safety instructions carefully.
2. Keep this User Manual for later reference.
3. Disconnect this equipment from any AC outlet before cleaning. Use a damp cloth. Do not use liquid or spray detergents for cleaning.
4. For plug-in equipment, the power outlet socket must be located near the equipment and must be easily accessible.
5. Keep this equipment away from humidity.
6. Put this equipment on a reliable surface during installation. Dropping it or letting it fall may cause damage.
7. The openings on the enclosure are for air convection. Protect the equipment from overheating. DO NOT COVER THE OPENINGS.
8. Make sure the voltage of the power source is correct before connecting the equipment to the power outlet.
9. Position the power cord so that people cannot step on it. Do not place anything over the power cord.
10. All cautions and warnings on the equipment should be noted.
11. If the equipment is not used for a long time, disconnect it from the power source to avoid damage by transient overvoltage.
12. Never pour any liquid into an opening. This may cause fire or electrical shock.
13. Never open the equipment. For safety reasons, the equipment should be opened only by qualified service personnel.
14. If one of the following situations arises, get the equipment checked by service personnel:
15. The power cord or plug is damaged.
16. Liquid has penetrated into the equipment.
17. The equipment has been exposed to moisture.
18. The equipment does not work well, or you cannot get it to work according to the user's manual.
19. The equipment has been dropped and damaged.
20. The equipment has obvious signs of breakage.
21. DO NOT LEAVE THIS EQUIPMENT IN AN ENVIRONMENT WHERE THE STORAGE TEMPERATURE MAY GO BELOW -20° C (-4° F) OR ABOVE 60° C (140° F). THIS COULD DAMAGE THE EQUIPMENT. THE EQUIPMENT SHOULD BE IN A CONTROLLED ENVIRONMENT.
22. CAUTION: DANGER OF EXPLOSION IF BATTERY IS INCORRECTLY REPLACED. REPLACE ONLY WITH THE SAME OR EQUIVALENT TYPE RECOMMENDED BY THE MANUFACTURER, DISCARD USED BATTERIES ACCORDING TO THE MANUFACTURER'S INSTRUCTIONS.
23. The sound pressure level at the operator's position according to IEC 704-1:1982 is no more than 70 dB (A).

DISCLAIMER: This set of instructions is given according to IEC 704-1. Advantech disclaims all responsibility for the accuracy of any statements contained herein.

Contents

Chapter 1 Understanding Your System 1

1.1 Introduction 2
1.2 Major Features 2
1.2.1 Communication Network 2
1.2.2 Modbus/TCP Protocol 2
1.2.3 Hardware Capacity & Diagnostic 2
1.2.4 Communicating Isolation 3
1.2.5 Completed set of I/O modules for total solutions 3
1.2.6 Built-in real-time OS and watchdog timer 3
1.2.7 Software Support 3
1.2.8 Security Setting 3
1.2.9 UDP Data Stream 3
1.2.10 Modbus Ethernet Data Gateway 3
1.3 Technical Specification of ADAM-5000/TCP 4
1.3.1 System 4
1.3.2 Ethernet Communication 4
1.3.3 Serial Communication 4
1.3.4 Power 4
1.3.5 Isolation 4
1.3.6 Mechanical 4
1.3.7 Environment 5
1.3.8 Dimensions 5
Figure 1.1 ADAM-5000/TCP system & I/O module dimensions .. 5
1.3.9 Basic Function Block Diagram 5
Figure 1.2 Function block diagram 5
1.4 LED Status of ADAM-5000/TCP Series main unit 6
Figure 1.3 ADAM-5000/TCP LED Indicators 6

Chapter 2 Selecting Your Hardware Components 7

2.1 Selecting I/O Module 8
Table 2.1: I/O Selection Guidelines 8
Figure 2.1 ADAM-5000 I/O Module Selection Chart 9
Table 2.2: I/O Modules Selection Guide 10
2.2 Selecting Power Supply 12
Table 2.3: Power Consumption of ADAM-5000 series 12
Table 2.4: Power Supply Specification Table 14
2.3 Selecting Link Terminal and Cable 14
Figure 2.2 Ethernet Terminal and Cable Connection 14
Table 2.5: Ethernet RJ-45 port Pin Assignment 15
Figure 2.3 RS-485 Terminal and Cable Connection 15
2.4 Selecting Operator Interface 16

Chapter 3 Hardware Installation Guide 17

3.1 Determining the proper environment 18
3.1.1 Check the content of shipping box 18
3.1.2 System Requirement 18
3.1.3 I/O modules 18
3.2 Installing your main unit and module 18
Figure 3.1 Module alignment and installation 19

Figure 3.2 Secure the module to the system 19
3.3 Mounting 19
3.3.1 Panel mounting 19
Figure 3.3 ADAM-5000/TCP panel mounting screw placement 20
3.3.2 DIN rail mounting 20
Figure 3.4 ADAM-5000/TCP DIN rail mounting 20
Figure 3.5 Secure ADAM-5000/TCP System to a DIN rail 21
3.4 Wiring and Connections 22
3.4.1 Power supply wiring 22
Figure 3.6 ADAM-5000/TCP power wiring 22
3.4.2 I/O modules wiring 23
Figure 3.7 ADAM-5000 I/O Module Terminal Block wiring 23
3.4.3 System Network Connections 24
Figure 3.8 System network connection 24
Figure 3.9 Serial Network Connection 25
3.5 Assigning address for I/O Modules 26
Figure 3.10 I/O Modules Address Mapping 26

Chapter 4 I/O modules 27

4.1 I/O Modules 28
Table 4.1: I/O Module Support List 28

Chapter 5 System Hardware Configuration 29

5.1 System Hardware Configuration 30
Figure 5.1 Hardware Configuration 30
5.2 Install Utility Software on Host PC 30
5.3 AdamApax .NET Utility Overview 30
5.3.1 Main Menu 31
Figure 5.2 Operation Screen 31
Figure 5.3 Tool Bar 33
5.3.2 Ethernet Network Setting 33
Figure 5.4 Network Setting 33
Figure 5.5 Define Device Name and Description 34
Figure 5.6 TCP/IP Network setting 34
5.3.3 I/O Module Configuration 35
Figure 5.7 Digital I/O Module Configuration 35
Figure 5.8 Operation and Indication Icons 36
Figure 5.9 Current Analog Input Status 36
Figure 5.10 Setting range and integration time 37
Figure 5.11 Analog Module Configuration Screen 37
Figure 5.12 Counter/Frequency Module Configuration 38
Figure 5.13 Location of Counter/Frequency Module 39
5.3.4 Alarm Setting To satisfy the needs of various applications, ... 39
Figure 5.14 Alarm Setting for Analog Input and Counter Modules .. 40
5.3.5 Firmware Update 40
Figure 5.15 Firmware Upgrade 41
5.3.6 Security Setting 41
Figure 5.16 Password Setting 41
5.3.7 Terminal Emulation 42
Figure 5.17 Command Emulation 42
5.3.8 Data Stream 42
Figure 5.18 Data Stream Configuration 43
Figure 5.19 Data Stream Monitoring 43
5.3.9 Data Gateway Setting 43
Figure 5.20 RS-485 Modbus Network Setting 44

Chapter 6 Programming 45

6.1 Introduction 46
6.2 DLL (Dynamic Link Library) Driver 46
6.2.1 Index 46
6.2.2 Programming Flows 47
6.2.3 Function Descriptions 52
6.2.4 Return Codes 60
6.2.5 Data Structure 60
6.3 ADAM-5000/TCP Command 61
6.3.1 Command Structure 61
Figure 6.1 Request Comment Structure 61
Figure 6.2 Response Comment Structure 61
6.3.2 Modbus Function Code Introduction 62
Table 6.1: Response Comment Structure 62
6.4 Apply with ASCII Command for ADAM-5000/TCP System 65
Figure 6.3 ASCII Command Structure in ADAM-5000/TCP 65
6.4.1 Sytax of ASCII 65
6.4.2 System Command Set 66
Table 6.2: CPU Command Set 66
Table 6.3: Baud rate codes 68
6.4.3 Analog Input Command Set 72
Table 6.4: ADAM-5013 RTD Input Command Set 72
Table 6.5: ADAM-5017/5018 Analog Input Command Set 83
Figure 6.4 Data format for 8-bit parameters 84
Table 6.6: ADAM-5017H /5017UH Analog Input command Set 95
Table 6.7: Analog Input Alarm Command Set 102
Table 6.8: Analog Input Alarm Command Set 112
6.4.4 Analog Output Command Set 122
Table 6.9: Analog Output command Set 122
Figure 6.5 The other bits are not used and are set to 0 123
6.4.5 Digital Input/Output Command Set 132
Table 6.10: Digital Input/Output Command Set 132
Table 6.11: ADAM-5080 Counter/Frequency Command Set 138
6.4.6 WatchDog Timer Command Set 162
Table 6.12: WatchDog Timer Command Set 162
Table 6.13: ADAM-5081 Counter/Frequency Command Set 165

Appendix A Design Worksheets 189

Table A.1: I/O Data Base 190
Table A.2: Summary Required Modules 191
Table A.3: Table for Programming 192

Appendix B Data Formats and I/O Ranges 195

B.1 Analog Input Formats 196
B.2 Analog Input Ranges - ADAM-5017 197
B.3 Analog Input Ranges - ADAM-5018 198

Appendix C Grounding Reference 203

C.1 Grounding 204
C.1.1 The 'Earth' for reference 204
Figure C.1 Think the EARTH as GROUND. 204
C.1.2 The 'Frame Ground' and 'Grounding Bar' 205
Figure C.2 Grounding Bar 205

Figure C.3 Normal mode and Common modC 205
C.1.3 Normal Mode and Common Mode 206
Figure C.4 Normal mode and Common modC 206
C.1.4 Wire impedance 207
Figure C.5 The purpose of high voltage transmission 207
Figure C.6 Wire impedancC 207
C.1.5 ADAM-5000/TCP User's Manual 208
Figure C.7 Single point grounding (1) 208
Figure C.8 Single point grounding (2) 208
C.2 Shielding 209
C.2.1 Cable Shield 209
Figure C.9 Single isolated cable 209
Figure C.10Double isolated cable 210
C.2.2 System Shielding 211
Figure C.11System Shielding 211
Figure C.12The characteristic of the cable 211
Figure C.13System Shielding (1) 212
Figure C.14System Shielding (2) 212
C.3 Noise Reduction Techniques 213
Figure C.15Noise Reduction Techniques 213
C.4 Check Point List 213

Chapter 1

Understanding Your System

1.1 Introduction

Ethernet connectivity is trending in industrial applications. Longer communication distances, faster communication speeds, and greater advantages attract people into developing their system based upon this network. But there used to be a threshold in connecting information layers and field control layers. People usually had to prepare a data exchange server between information systems and control systems as a communication bridge. Obviously, it takes a lot of time and money. To meet user's requirements, Advantech announces the new DA&C system, the ADAM-5000/TCP Series, the Ethernet I/O solution for people developing their eAutomation architecture. It can be applied to various applications, such as traffic, building, telecom, water treatment, and others.

ADAM-5000/TCP Series include the following 2 products:

ADAM-5000/TCP: 8-slot Distributed DA&C System for Ethernet

ADAM-5000L/TCP: 4-slot Distributed DA&C System for Ethernet

1.2 Major Features

1.2.1 Communication Network

By adopting a 32-bit RISC CPU, the ADAM-5000/TCP Series has greatly advanced data processing abilities for the user, especially for network communications (response time < 5ms). There is a standard RJ-45 modular jack Ethernet port on the ADAM-5000/TCP'S CPU board, and I/O modules field signals would be able to link with the Ethernet directly without assistance from other hardware devices such as converters or data gateways. The communication speeds can be autoswitched between 10 M and 100 Mbps data transfer rate depending upon the network environment. Through an Ethernet network, your DA&C systems, computer workstations, and higher-level enterprise MIS servers can access plant- floor data. Such data can be used in system supervising, product scheduling, statistical quality control, and more.

1.2.2 Modbus/TCP Protocol

Modbus/TCP is one of the most popular standards for industrial Ethernet networks. Following this communication protocol, the ADAM-5000/TCP Series is easy to integrate with any HMI software packages or user-developed applications that support Modbus. Users do not have to prepare a specific driver for the ADAM-5000/TCP Series when they install the DA&C system with their own operating application. Moreover, the ADAM-5000/TCP Series works as a Modbus data server. It allows eight PCs or tasks to access its current data simultaneously from anywhere: LAN, Intranet, or Internet.

1.2.3 Hardware Capacity & Diagnostic

Advantech's ADAM-5000/TCP Series is designed with a high I/O capacity and supports all types of ADAM-5000 I/O modules. Providing eight slots for any mixed modules, this DA&C system handles up to 128 I/O points (four ADAM-5024s allowed). Different from other main units, the ADAM-5000/TCP Series not only has a higher I/O capacity, but it also has a smarter diagnostic ability. There are eight indicators on the front case of the CPU module. Users can read the system status clearly, including power, CPU, Ethernet link, communication active, communication rate, and more. In addition, there are also Tx and Rx LEDs on the Ethernet port, indicating data transfer and reception.

1.2.4 Communicating Isolation

High-speed transient suppressors isolate ADAM-5000/TCP Series Ethernet port from dangerous voltage (up to 1500VDC) power spikes and avoid surge damage to the whole system.

1.2.5 Completed set of I/O modules for total solutions

The ADAM-5000/TCP Series uses a convenient backplane system common to the ADAM-5000 series. Advantech's complete line of ADAM-5000 modules integrates with the ADAM-5000/TCP Series to support your applications (not including ADAM-5090). Full ranges of digital module supports 10 to 30 VDC input and outputs. A set of analog modules provide 16-bit resolution and programmable input and output (including bipolar) signal ranges. For details, refer to Chapter 4 I/O Modules.

1.2.6 Built-in real-time OS and watchdog timer

The microprocessor also includes a real-time OS and watchdog timer. A real-time OS is available to handle several tasks at the same time. The watchdog timer is designed to automatically reset the microprocessor if the system fails. This feature greatly reduces the level of maintenance required and makes the ADAM-5000/TCP Series ideal for use in applications which require a high level of system performance and stability.

1.2.7 Software Support

The ADAM-5000/TCP Series firmware has a built-in Modbus/TCP server based on the Modbus standard. Therefore, Advantech provides all the necessary DLL drivers, OPC Server, and Windows Utility for users for client data for the ADAM-5000/TCP Series. Users can configure this DA&C system via a Windows Utility; and integrate them with HMI software packages via Modbus/TCP driver or Modbus/TCP OPC Server. Even more, you can use the DLL driver or standard Modbus/TCP library to develop your own applications.

1.2.8 Security Setting

Though Ethernet technology comes with great benefits in speed and integration, there also exist risks about network invasion from outside. For this reason, a security protection design was built into the ADAM-5000/TCP Series. Once the user has set the password into the ADAM-5000/ TCP firmware, important system configurations (network, firmware, password) can only be changed through password verification.

1.2.9 UDP Data Stream

Most of time, each host PC in a DA&C system needs to regularly request the I/O devices via TCP/IP packets to update current data. This may cause data problems and lower performance on the network, especially when there are frequent communications between multi-servers and I/O devices. To reduce communication issues in the host computer on your Ethernet network, the ADAM-5000/TCP Series also supports UDP (User Datagram Protocol) protocol to broadcast data packets to specific IPs without requesting commands. Users can apply this feature to implement data streaming, event triggers, and other advanced functions.

1.2.10 Modbus Ethernet Data Gateway

Much more than an I/O system, ADAM-5000/TCP Series provides an RS-485 network interface for other Modbus devices integration. It works as an Ethernet data gateway, upgrading Modbus serial network devices up to Ethernet layer. ADAM-5511

supports a maximum of 16 nodes or 3rd party products with Modbus protocol support which are allowed to integrate with ADAM-5000/TCP Series. This feature enlarges your system scope, as opposed to other general I/O systems.

1.3 Technical Specification of ADAM-5000/TCP

1.3.1 System

- **CPU:** ARM 32-bit RISC CPU
- **Memory:** 4 MB Flash RAM
- **Operating System:** Real-time O/S
- **Timer BIOS:** Yes
- **I/O Capacity:** 8 slots (ADAM-5000/TCP)
  4 slots (ADAM-5000L/TCP)
- **Status Indicator:** Power (3.3V, 5V), CPU, Communication (Link, Collide, 10/100 Mbps)
- **CPU Power Consumption:** 5.0W
- **Reset Push Bottom:** Yes

1.3.2 Ethernet Communication

- **Ethernet:** 10 BASE-T IEEE 802.3
  100 BASE-TX IEEE 802.3u
- **Wiring:** UTP, category 5 or greater
- **Bus Connection:** 2 RJ45 LAN port share one MAC address
- **Comm. Protocol:** Modbus/TCP
- **Data Transfer Rate:** Up to 100 Mbps
- **Max Communication Distance:** 100 meters
- **Even Response Time:** < 5 ms
- **Data Stream Rate:** 50 ms to 7 days

1.3.3 Serial Communication

- **RS-485 signals:** DATA +, DATA-
- **Mode:** Half duplex, multi-drop
- **Connector:** Screw terminal
- **Transmission Speed:** Up to 115.2 Kbps
- **Max. Transmission Distance:** 4000 feet (1220 m)

1.3.4 Power

- **Unregulated 10 to 30V_{DC}**
- **Protection:** Over-voltage and power reversal

1.3.5 Isolation

- **Ethernet Communication:** 1500 V DC
- **I/O Module:** 3000 V DC

1.3.6 Mechanical

- **Case:** KJW with captive mounting hardware
- **Plug-in Screw Terminal Block:**
  Accepts 0.5 mm2 to 2.5 mm2, 1 - #12 or 2 - #14 to #22 AWG

1.3.7 Environment

- **Operating Temperature:** - 10 to 70ºC (14 to 158ºF)
- **Storage Temperature:** - 25 to 85ºC (-13 to 185ºF)
- **Humidity:** 5 to 95%, non-condensing
- **Atmosphere:** No corrosive gases

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

1.3.8 Dimensions

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

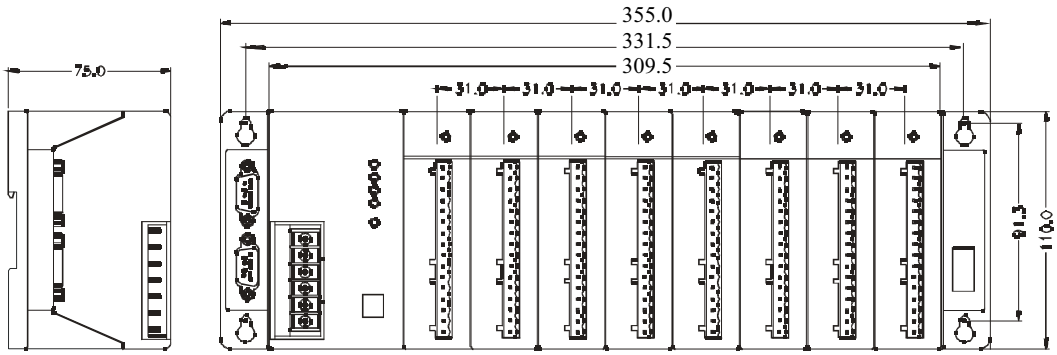


Figure 1.1 ADAM-5000/TCP system & I/O module dimensions

1.3.9 Basic Function Block Diagram

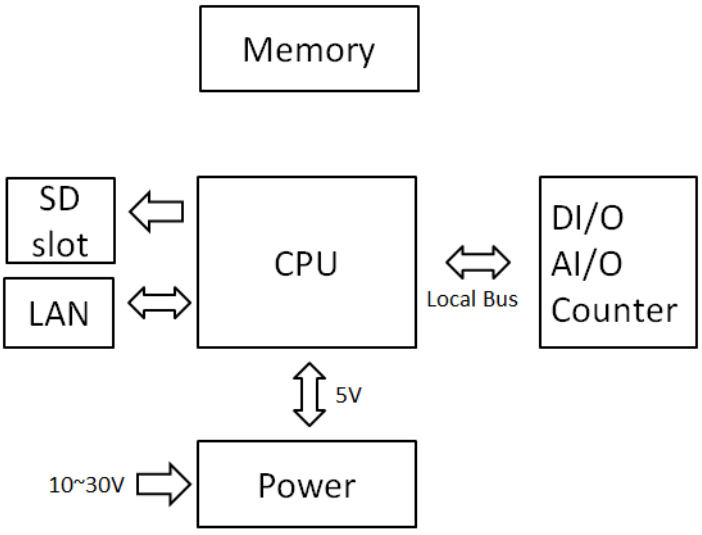


Figure 1.2 Function block diagram

1.4 LED Status of ADAM-5000/TCP Series main unit

There are eight LEDs on the ADAM-5000/TCP Series front panel. The LEDs indicate ADAM-5000/TCP's system status, as explained below:

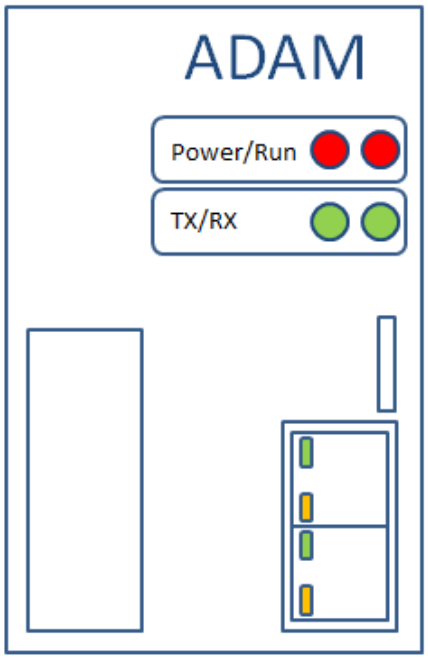


Figure 1.3 ADAM-5000/TCP LED Indicators

Power: Red indicator. This LED is normal when ADAM-5000/TCP Series system is powered on.

Chapter 2

Selecting Your Hardware Components

2.1 Selecting I/O Module

To organize an ADAM-5000/TCP Series data acquisition & control system, you need to select I/O modules to interface the main unit with field devices or processes that you have previously determined. There are several things you should consider when you select I/O modules.

- What type of I/O signal is applied in your system?
- How much I/O is required by your system?
- How will you place the main unit to concentrate the I/O points of an entire process.
- How many ADAM-5000/TCP Series main units are required for distributed I/O points arrangement?
- What is the required voltage range for each I/O module?
- What isolation environment is required for each I/O module?
- What are the noise and distance limitations for each I/O module?

Refer to table 2-1 I/O module selection guidelines.

Table 2.1: I/O Selection Guidelines

Choose this type of I/O module:	For these types of field devices or operations (examples):	Explanation:
Discrete input module and block I/O module	Selector switches, pushbuttons, photoelectric eyes, limit switches, circuit breakers, proximity switches, level switches, motor starter contacts, relay contacts, thumbwheel switches	Input modules sense ON/OFF or OPENED/CLOSED signals. Discrete signals can be either AC or DC.
Discrete output module and block I/O module	Alarms, control relays, fans, lights, horns, valves, motor starters, solenoids	Output module signals interface with ON/OFF or OPENED/CLOSED devices. Discrete signals can be either AC or DC.
Analog input module	Thermocouple signals, RTD signals, temperature transducers, pressure transducers, load cell transducers, humidity transducers, flow transducers, potentiometers.	Convert continuous analog signals into input values for ADAM-5000/TCP
Analog output module	Analog valves, actuators, chart recorders, electric motor drives, analog meters	Interpret ADAM-5000/TCP series output to analog signals (generally through transducers) for field devices.

Advantech provides more than 15 types of ADAM-5000 I/O modules for various applications. Figure 2-1 and Table 2-2 will help you to select the ADAM-5000 I/O modules quickly and easily.

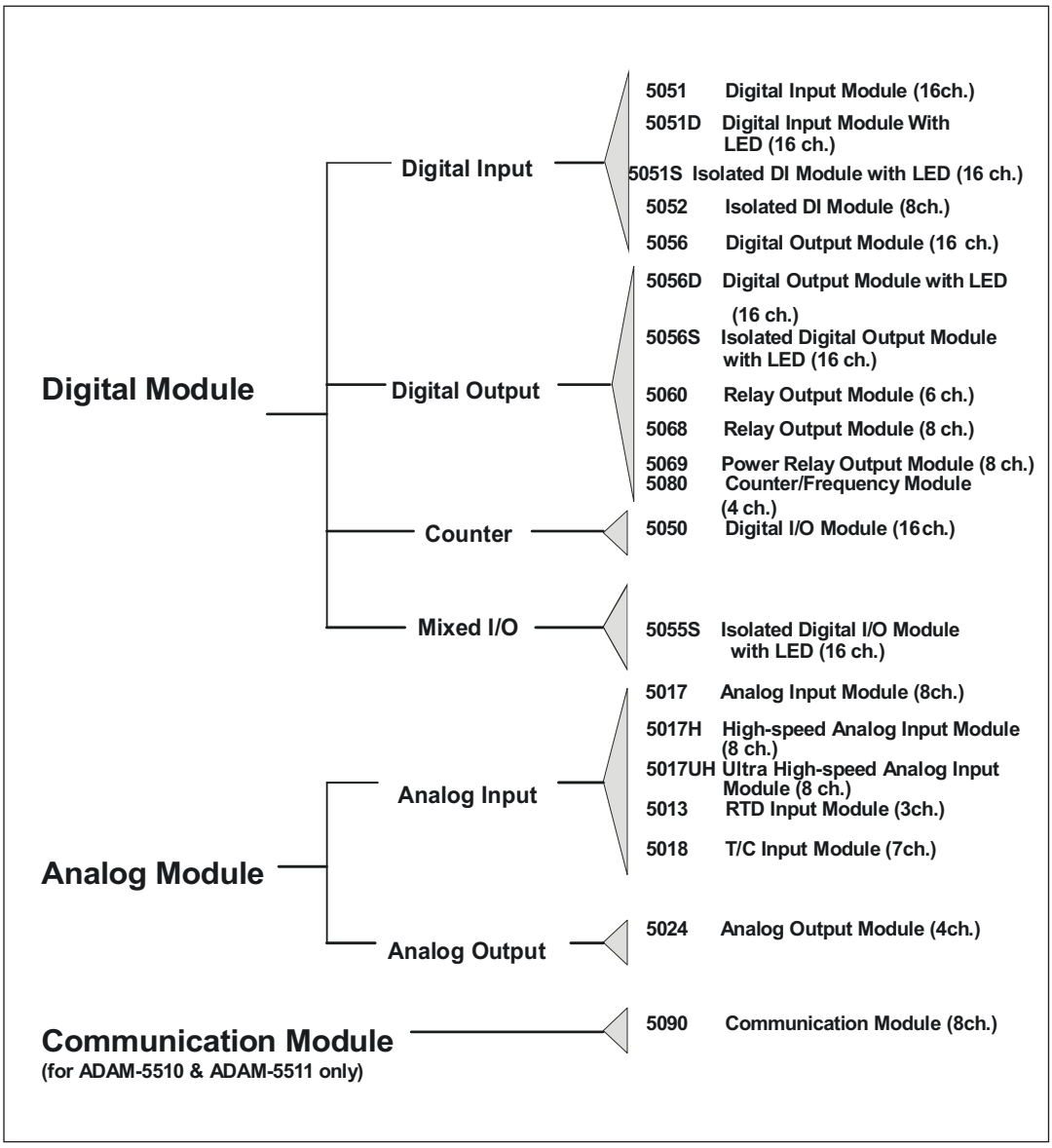


Figure 2.1 ADAM-5000 I/O Module Selection Chart

Table 2.2: I/O Modules Selection Guide

Module		ADAM-5013	ADAM-5017	ADAM-5017H	ADAM-5017UH	ADAM-5018	ADAM-5018P	ADAM-5024
	Resolution	16 bit	16 bit	12 bit	12 bit	16 bit	16 bit	-
Analog Input	Input Channel	3	8	8	8	7	7	-
	Sampling Rate	10	10	8K	200K	10	10	-
	Voltage Input	-	-±150 mV ±500 mV ±1 V ±5 V ±10 V	±250 mV ±500 mV ±1 V ±5 V ±10 V	V +10V V ±10 V	±15 mV ±50 mV ±100 mV ±500 mV ±1 V ±2.5 V	±15 mV ±50 mV ±100 mV ±500 mV ±1 V ±2.5 V	-
	Current Input	-	±20 mA*	±20 mA*	4~20mA* ±20 mA*	±20 mA*	±20 mA*	-
	Direct Sensor Input	Pt or Ni RTD	-		-	J, K, T, E, R, S, B	J, K, T, E, R, S, B	-
Analog Output	Resolution	-	-		-		-	12 bit
	Voltage Output	-	-		-		-	0~10 V
	Current Output	-	-		-		-	0~20 mA 4~20 mA
Digital Input and Digital Output	Digital Input Channels	-	-		-		-	-
	Digital Output Channels	-	-		-		-	-
Counter (32-bit)	Channels	-	-		-		-	-
	Input Frequency	-	-		-		-	-
	Mode	-	-		-		-	-
COMM	Channels	-	-		-		-	-
	Type	-	-		-		-	-
Isolation		3000 VDC	3000 VDC	3000 VDC	3000 VDC	3000 VDC	3000 VDC	3000 VDC

Module		ADAM-5050	ADAM-5051	ADAM-5051D	ADAM-5051S
	Resolution	-	-	-	-
Analog Input	Input Channel	-	-	-	-
	Sampling Rate	-	-	-	-
	Voltage Input	-	-	-	-
	Current Input	-	-	-	-
	Direct Sensor Input	-	-	-	-

Analog Output	Resolution	-	-	-	-
	Voltage Output	-	-	-	-
	Current Output	-	-	-	-
Digital Input and Digital Output	Digital Input Channels	16 DIO (bit-wise selectable)	16	16 W/LED	16 W/LED
	Digital Output Channels		-	-	-
Counter (32-bit)	Channels	-	-	-	-
	Input Frequency	-	-	-	-
	Mode	-	-	-	-
COMM	Channels	-	-	-	-
	Type	-	-	-	-
Isolation		-	-	-	2500 VDC

Module		**ADAM-5055S**	**ADAM-5055S**	**ADAM-5056**	**ADAM-5056D**	**ADAM-5056S /5056SO**
	Resolution	-	-	-	-	-
Analog Input	Input Channel	-	-	-	-	-
	Sampling Rate	-	-	-	-	-
	Voltage Input	-	-	-	-	-
	Current Input	-	-	-	-	-
	Direct Sensor Input	-	-	-	-	-
Analog Output	Resolution	-	-	-	-	-
	Voltage Output	-	-	-	-	-
	Current Output	-	-	-	-	-
Digital Input and Digital Output	Digital Input Channels	8	8 W/LED	-	-	-
	Digital Output Channels	-	8 W/LED	16	16 W/LED	16 W/LED
Counter (32-bit)	Channels	-	-	-	-	-
	Input Frequency	-	-	-	-	-
	Mode	-	-	-	-	-
COMM	Channels	-	-	-	-	-
	Type	-	-	-	-	-
Isolation		5000 VRMS	2500 VDC	-	-	2500 VDC

Module		ADAM-5060	ADAM-5068	ADAM-5080	ADAM-5090
	Resolution	-	-	-	-
Analog Input	Input Channel	-	-	-	-
	Sampling Rate	-	-	-	-
	Voltage Input	-	-	-	-
	Current Input	-	-	-	-
	Direct Sensor Input	-	-	-	-
Analog Output	Resolution	-	-	-	-
	Voltage Output	-	-	-	-
	Current Output	-	-	-	-
Digital Input and Digital Output	Digital Input Channels	-	-	-	-
	Digital Output Channels	6 relay (2 form A/ 4 form C)	8 relay (8 form A)	-	-
Counter (32-bit)	Channels	-	-	4	-
	Input Frequency	-	-	5000 Hz (max)	-
	Mode	-	-	Frequency, Up/Down Counter, Bi-direction Counter	-
COMM	Channels	-	-	-	4
	Type	-	-	-	RS-232
Isolation		-	-	1000 VRMS	-

2.2 Selecting Power Supply

ADAM-5000/TCP series system works under unregulated power source between +10 and +30 VDC. When you arrange different I/O modules on ADAM-5000/TCP's back plant, it may require comparable power supply. Use the following steps as guidelines for selecting a power supply for your ADAM-5000/TCP system.

- Refer to Table 2.3 to check the power consumption of ADAM-5000/TCP Series main unit and each I/O module.

Table 2.3: Power Consumption of ADAM-5000 series

Main Units	Description	Power Consumption
ADAM-5000/485	Distributed Data Acquisition and Control System based on RS-485	1.0 W
ADAM-5000E	Distributed Data Acquisition and Control System based on RS-485	4.0 W
ADAM-5000/TCP	Distributed Data Acquisition and Control System based on Ethernet	5.0 W

Table 2.3: Power Consumption of ADAM-5000 series		
ADAM-5510	PC-Based Programmable Controller (With Battery Backup)	1.0 W
ADAM-5510M	Enhanced PC-Based Programmable Controller (With Battery Backup)	1.2 W
ADAM-5511	PC-Based Programmable Controller with Modbus	1.0 W
ADAM-5510E	8-clot PC-Based Programmable Controller	1.2W
ADAM-5510/TCP	Ethernet-enabled PC-Based Programmable Controller	2.0W
ADAM-5510E/TCP	8-clot Ethernet-enabled PC-Based Programmable Controller	2.0W
Main Units	**Description**	**Power Consumption**
ADAM-5013	3-Channel RTD Input Module	1.1 W
ADAM-5017	8-Channel Analog Input Module (mV, mA or High Voltage)	1.25 W
ADAM-5017H	8-Channel High speed Analog Input Module (mV, mA or High Voltage)	2.2 W
ADAM-5017UH	8-Channel Ultra High speed Analog Input Module (mV, mA or High Voltage)	2.2 W
ADAM-5018	7-Channel Thermocouple Input Module (mV, V, mA, Thermocopule)	0.63 W
ADAM-5024	4-Channel Analog Output Module (V, mA)	2.9 W
ADAM-5050	16-Channel Universal DIO	1.2 W
ADAM-5051	16-Channel Digital Input Module	0.53 W
ADAM-5051D	16-Channel Digital Input w/LED Module	0.84 W
ADAM-5056S	16-Channel Isolated Digital Input w/LED Module	0.8 W
ADAM-5056SO	16-Channel Digital Input w/LED Module	0.84 W
ADAM-5052	8-Channel Isolated DI	0.27W
ADAM-5055S	16-Channel Isolated DIO w/LED Module	0.68 W
ADAM-5056	16-Channel Digital Output Module	0.53 W
ADAM-5056D	16-Channel Digital Output w/LED Module	0.84 W
ADAM-5056S	16-Channel Isolated Digital Output w/LED Module	0.6 W
ADAM-5060	6-Channel Relay Output Module (2 of Form A, 4 of Form C)	1.8 W
ADAM-5068	8-Channel Relay Output Module (8 of Form A)	1.8 W
ADAM-5080	4-Channel Counter/ Frequency Input Module	1.5 W
ADAM-5090	4-Port RS232 Module	0.6 W

Calculate the Summary of the whole system's power consumption.

For example, if there are these following items in your system.

ADAM-5000/TCP * 3 & ADAM-5024 * 4 & ADAM-5017 * 6 &

ADAM-5068 * 5 & ADAM-5050 * 5 & ADAM-5080 * 4

The power consumption is:

5W * 3 + 2.9W * 4 + 1.25 * 6 + 1.8W * 5 + 1.2W * 5 + 1.5W * 4 = 55.1W

ƒ Select a suitable power supply from Table 2.4 or other comparable power resources for system operation.

Table 2.4: Power Supply Specification Table			
Specification	**PWR-242**	**PWR-243**	**PWR-244**
Input			
Input Voltage	90~264 V_{AC}	85~132 V_{AC} 170~264 V_{AC}	100~240 V_{AC}
Input Frequency	47~63 Hz	47~63 Hz	47~63 Hz
Input Current	1.2 A max.	1.4 A max.	25 A/110 V_{AC} 50 A/220 V_{AC} (Inrush current)
Short Protection	Yes	Yes	Yes
Output			
Output Voltage	+24 V_{DC}	+24 V_{DC}	+24 V_{DC}
Output Current	2.1 A	3 A	4.2 A
Overload Protection	Yes	Yes	Yes
General			
Dimension	181 mm x 113 mm x 60 mm (L x W x H)	181 mm x 113 mm x 60 mm (L x W x H)	181 mm x 113 mm x 60 mm (L x W x H)
Operating Temperature	0~50º C (32~122º F)	0~50º C (32~122º F)	0~50º C (32~122º F)
DIN-rail Mountable	Yes	No	No

2.3 Selecting Link Terminal and Cable

Ethernet Network

Use the RJ-45 connector to connect the Ethernet port of the ADAM-5000/TCP Series to the Hub. The cable for connection should be Category 3 (for 10Mbps data rates) or Category 5 (for 100Mbps data rates). UTP/STP cables are compliant with EIA/TIA 586 specifications. The maximum length between the Hub and any ADAM-5000/TCP Series is up to 100 meters (approx. 300ft).

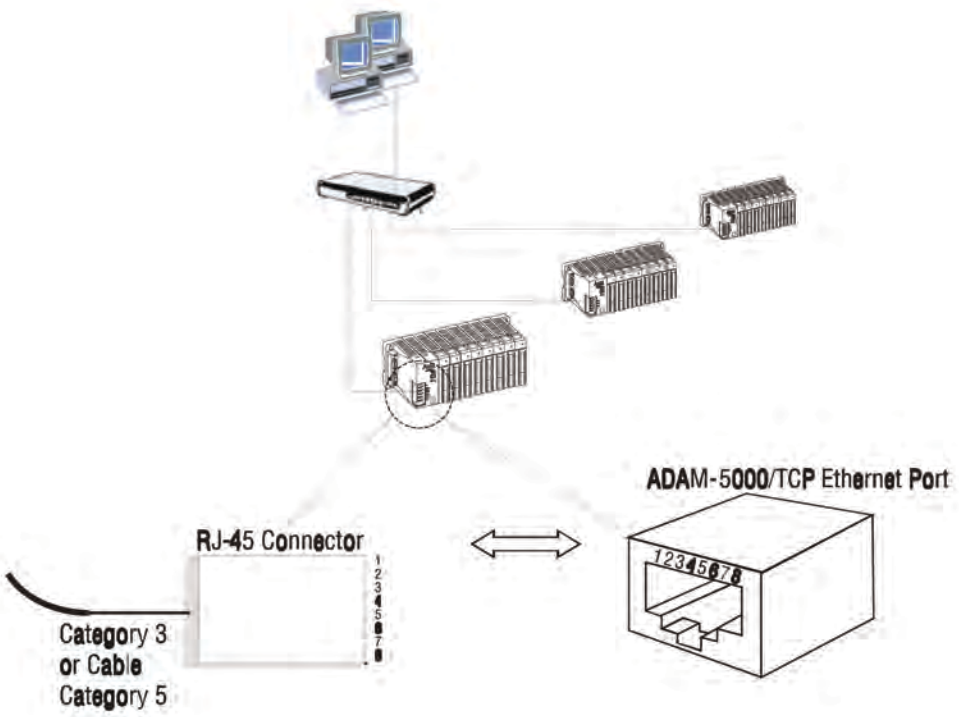


Figure 2.2 Ethernet Terminal and Cable Connection

Table 2.5: Ethernet RJ-45 port Pin Assignment		
Pin number	**Signal**	**Function**
1	RD+	Receive (+)
2	RD-	Receive (-)
3	TD+	Transmit (+)
4	(Not Used)	-
5	(Not Used)	-
6	TD-	Transmit (-)
7	(Not Used)	-
8	(Not Used)	-

Serial Network

The system uses screw terminal for RS-485 twisted pair connection as a data gateway between Ethernet Sever and serial Modbus devices. See Figure 2-3. The following information must be considered.

1. Twisted-pair wire compliant with EIA-422 or EIA-485 standards, which contains 24 AWG thin copper conductor with copper mesh and aluminum foil for shielding.
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 wire trays for routing where possible.
5. Avoid running wires near high energy wiring.
6. To reduce electrical noise, it should be twisted as tightly as possible.

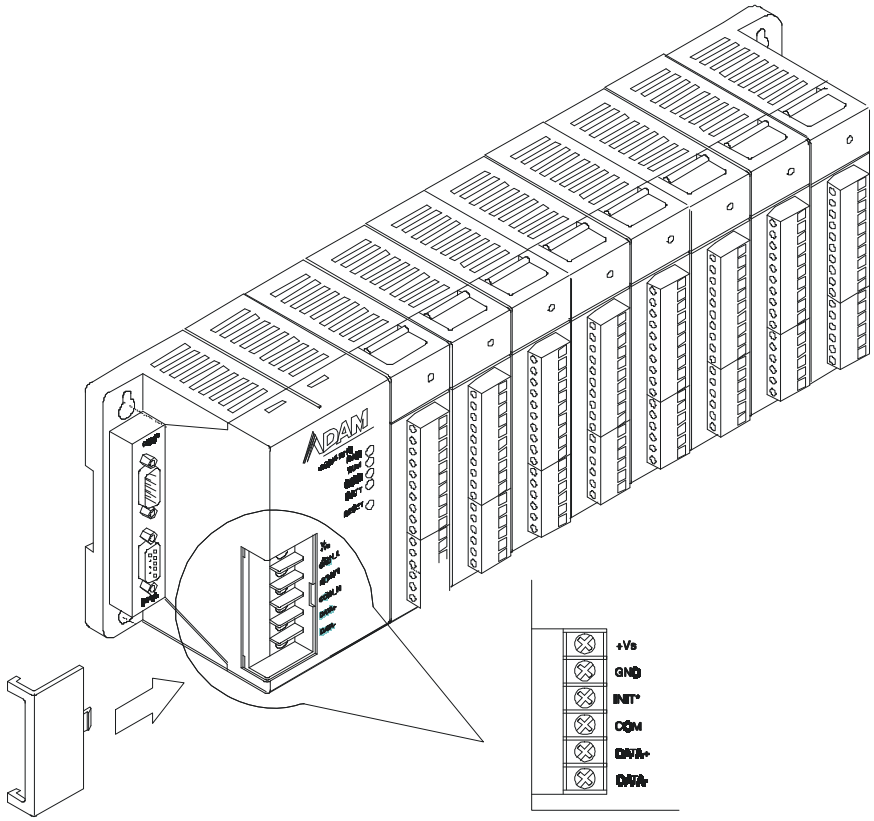


Figure 2.3 RS-485 Terminal and Cable Connection

2.4 Selecting Operator Interface

To complete your data acquisition and control system, selecting the operator interface is necessary. The Modbus/TCP protocol in ADAM-5000/TCP Series exhibits high flexibility for system integration for various applications. If you want to configure your ADAM-5000/TCP Series system, or monitor current status, Advantech offers free charge software:

- AdamApax .NET Utility

If you want to integrate ADAM-5000/TCP Series with HMI (Human Machine Interface) software in a SCADA (Supervisory Control and Data Acquisition) system. There are a lot of HMI software packages, which support Modbus/TCP drivers.

- Advantech WebAccess
- Wonderware InTouch
- Intellution Fix of i-Fix
- Advantech also provides an OPC Server - the most easy-to-use data exchange tool worldwide. Any HMI software designed with an OPC Client would be able to access ADAM-5000/TCP series systems.
- AdamApax .NET Class Library

With these ready-to-go application software packages, tasks such as remote data acquisition, process control, historical trending, and data analysis require only a few keystrokes to use.

Chapter 3

Hardware Installation Guide

3.1 Determining the proper environment

Before you start to install the ADAM-5000/TCP Series system, there are somethings that need to be checked.

3.1.1 Check the content of shipping box

Unpack the shipping boxes and make sure that the contents include:

- ADAM-5000/TCP Series main unit with two blank slot covers

3.1.2 System Requirement

- Host computer
  - IBM PC compatible computer with 486 CPU (Pentium is recommended)
  - Microsoft 95/98/2000/NT 4.0 (SP3 or SP4) or higher versions
  - At least 32 MB RAM
  - 20 MB of hard disk space available
  - VGA color monitor
  - 2x or higher speed CD-ROM
  - Mouse or other pointing devices
  - 10 or 100 Mbps Ethernet card
- 10 or 100 Mbps Ethernet Hub (at least 2 ports)
- Two Ethernet cables with RJ-45 connectors
- Power supply for ADAM-5000/TCP series (+10 to +30 V unregulated)

3.1.3 I/O modules

At least one I/O module is needed to use the system. Prepare the required I/O modules as the interface for a variety of field singles.

3.2 Installing your main unit and module

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 back plane connector (see figure 3-1). 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 (see figure 3-2).

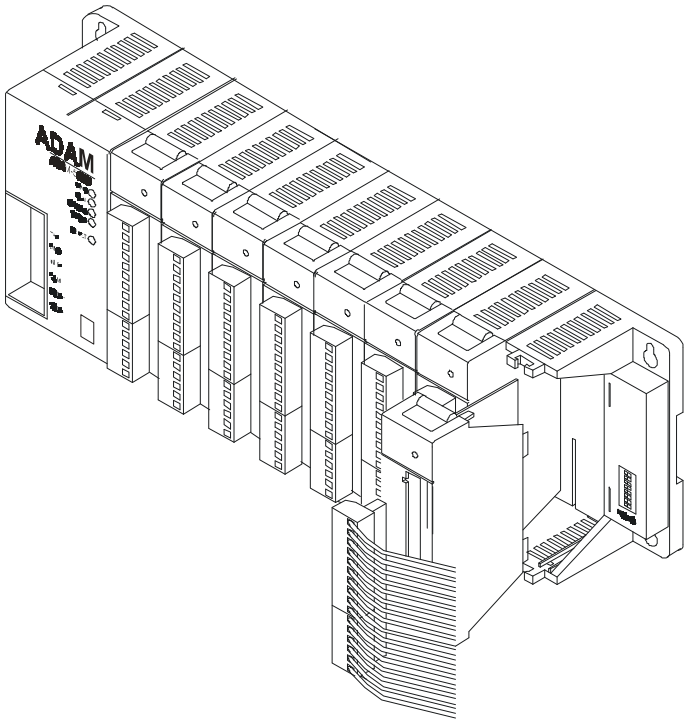


Figure 3.1 Module alignment and installation

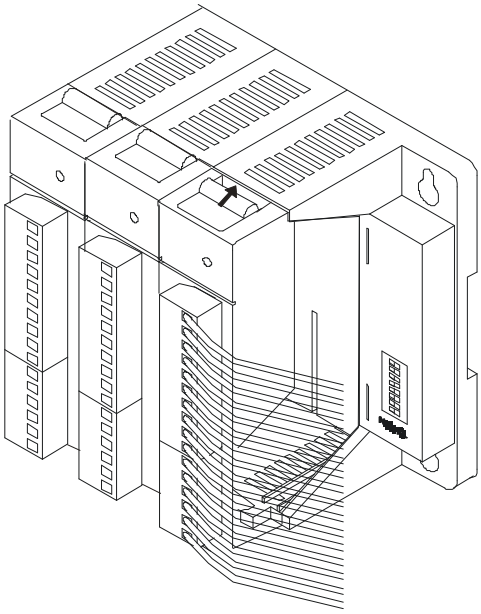


Figure 3.2 Secure the module to the system

3.3 Mounting

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

3.3.1 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 tapping screw (4 mm diameter) should be used.

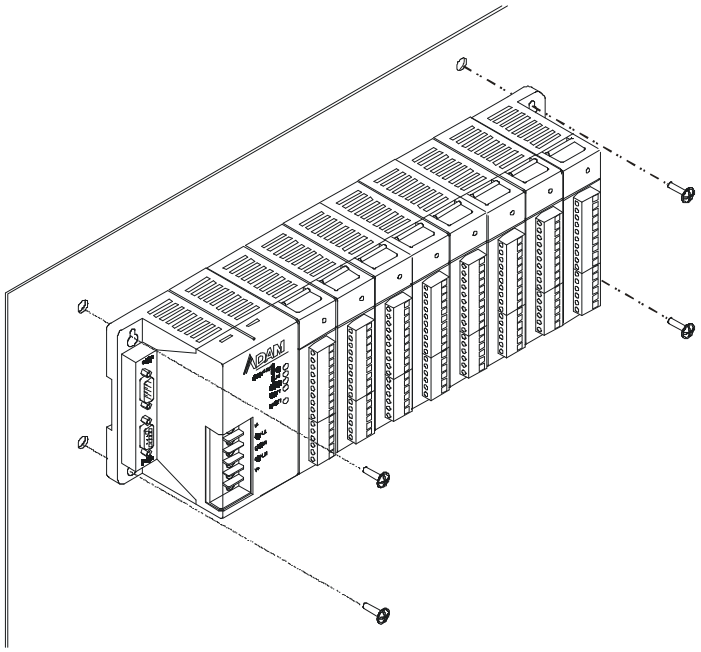


Figure 3.3 ADAM-5000/TCP panel mounting screw placement

3.3.2 DIN rail mounting

The system can also be secured to the cabinet by using mounting rails (see figure 3-4). If you mount the system on a rail, you should also consider using end brackets at the end of each rail. The end brackets help keep the system from sliding horizontally along the rail. This minimizes the possibility of accidentally pulling the wiring loose. If you examine the bottom of the system, you will notice two small retaining clips. To secure the system to a DIN rail, place the system on to the rail and gently push up on the retaining clips (see figure 3-5). 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.

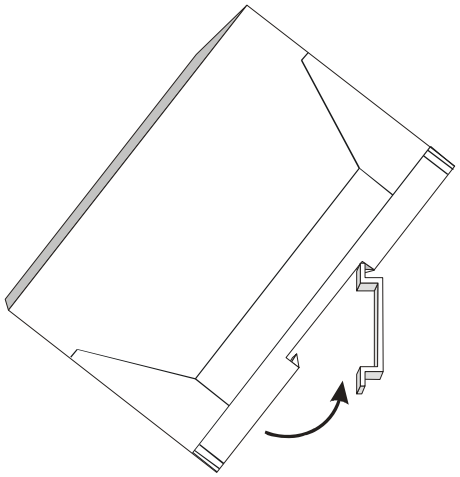


Figure 3.4 ADAM-5000/TCP DIN rail mounting

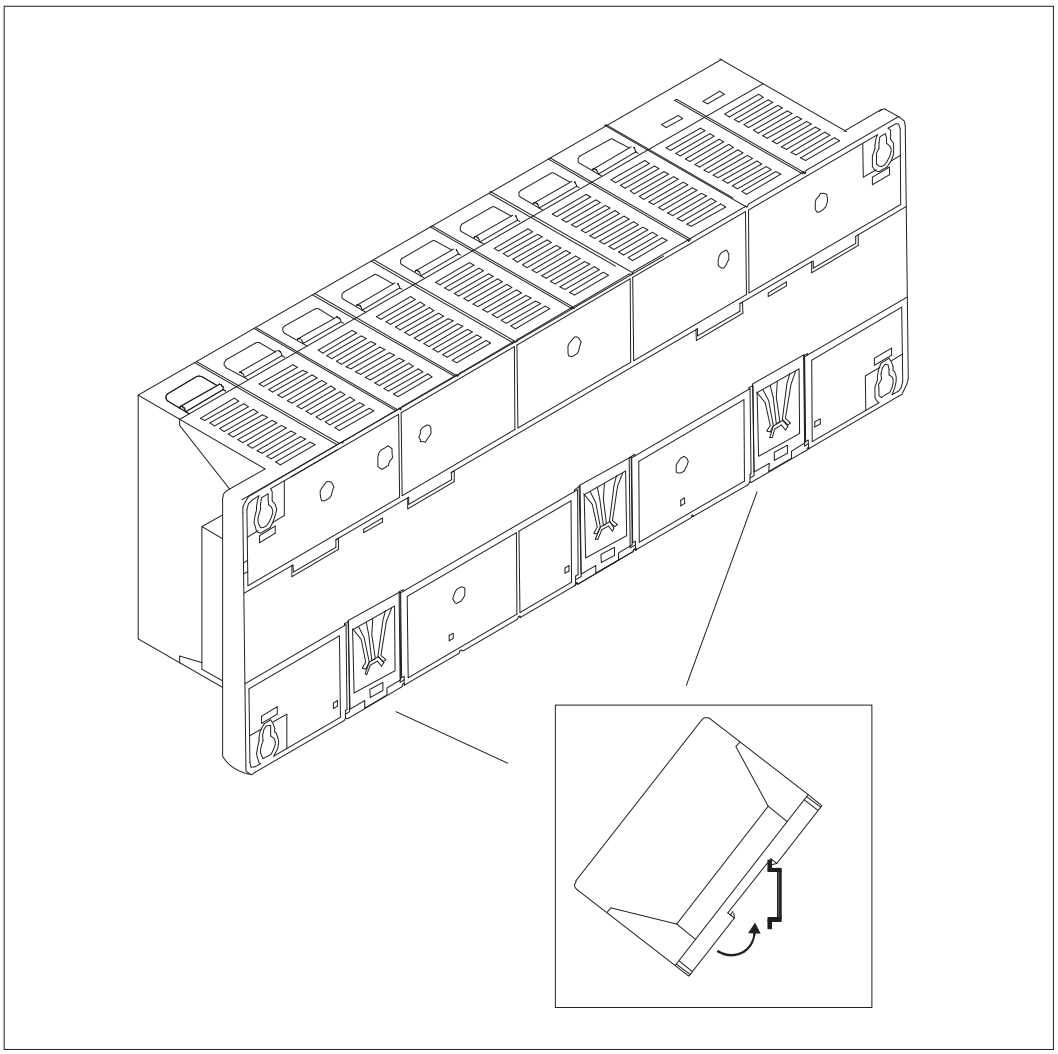


Figure 3.5 Secure ADAM-5000/TCP System to a DIN rail

3.4 Wiring and Connections

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

3.4.1 Power supply wiring

Although the ADAM-5000/TCP Series systems are designed for a standard industrial unregulated 24 V DC power supply, they accept any power unit that supplies within the range of +10 to +30 VDC. The power supply ripple must be limited to 200 mV peak-to-peak, and the immediate ripple voltage should be maintained between +10 and +30 VDC. Screw terminals +Vs and GND are for power supply wiring.

Note! *The wires used should be sized at least 2 mm.*

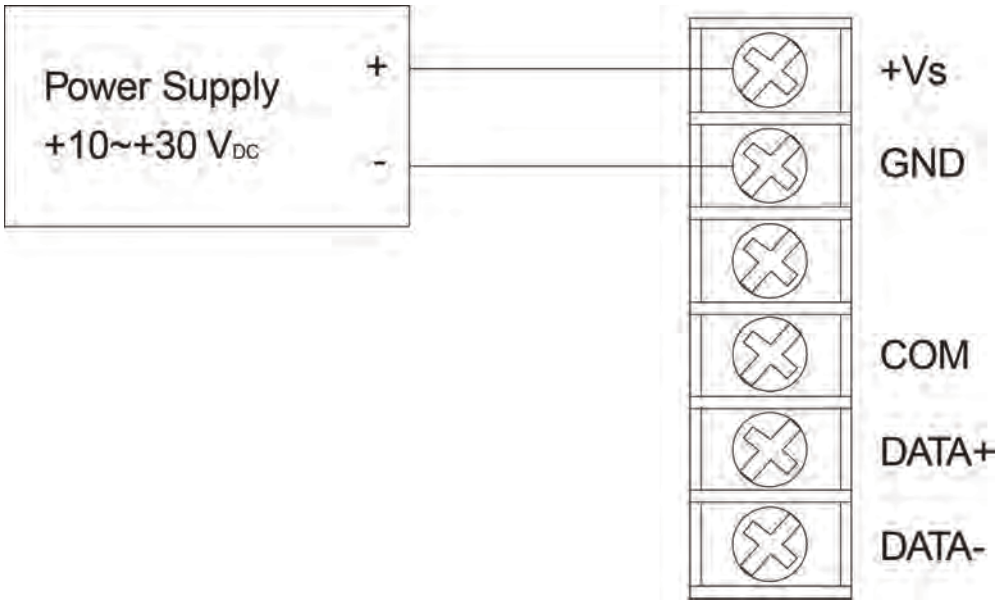


Figure 3.6 ADAM-5000/TCP power wiring

3.4.2 I/O modules wiring

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

1. The terminal block accepts wires from 0.5 mm to 2.5 mm.
2. Always use a continuous length of wire. Do not combine wires to make them longer.
3. Use the shortest possible wire length.
4. Use 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.

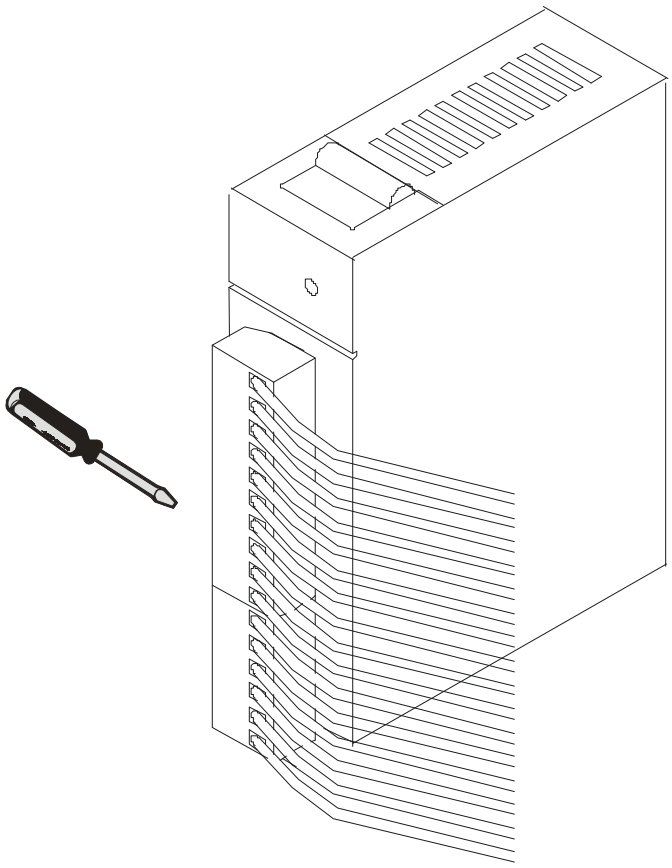


Figure 3.7 ADAM-5000 I/O Module Terminal Block wiring

3.4.3 System Network Connections

Ethernet Network

The ADAM-5000/TCP Series has an Ethernet communication port that allows you to program, configure, monitor, and integrate into a SCADA system. Figure 3-8 is a guideline to complete the system network connection.

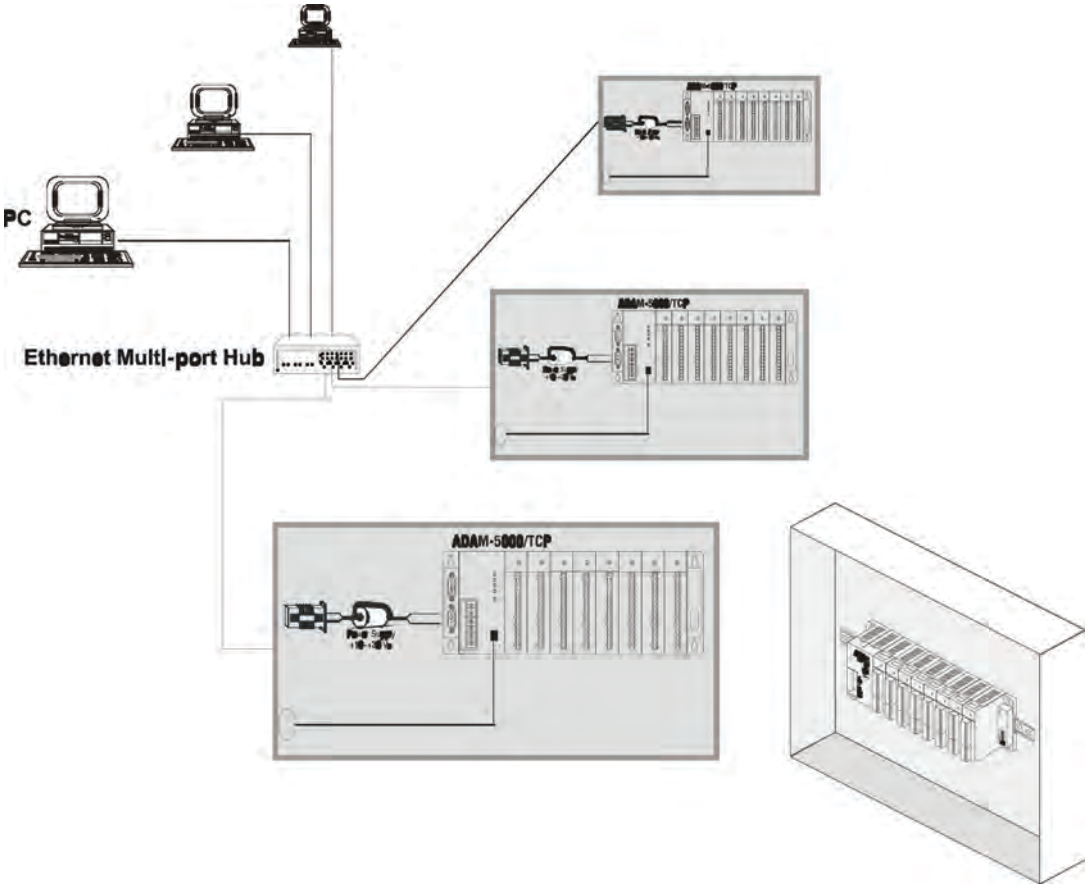


Figure 3.8 System network connection

Serial Network

Working as an Ethernet Data Gateway, the ADAM-5000/TCP Series provides an RS-485 interface to integrate serial devices for various applications. Adopting the Modbus standard protocol, it solves the communication problem between different networks and different devices. Meanwhile, users can extend their system scope by integrating up to 32 nodes of ADAM-5000/485 or other Modbus products, such as meters, card readers, and so on.

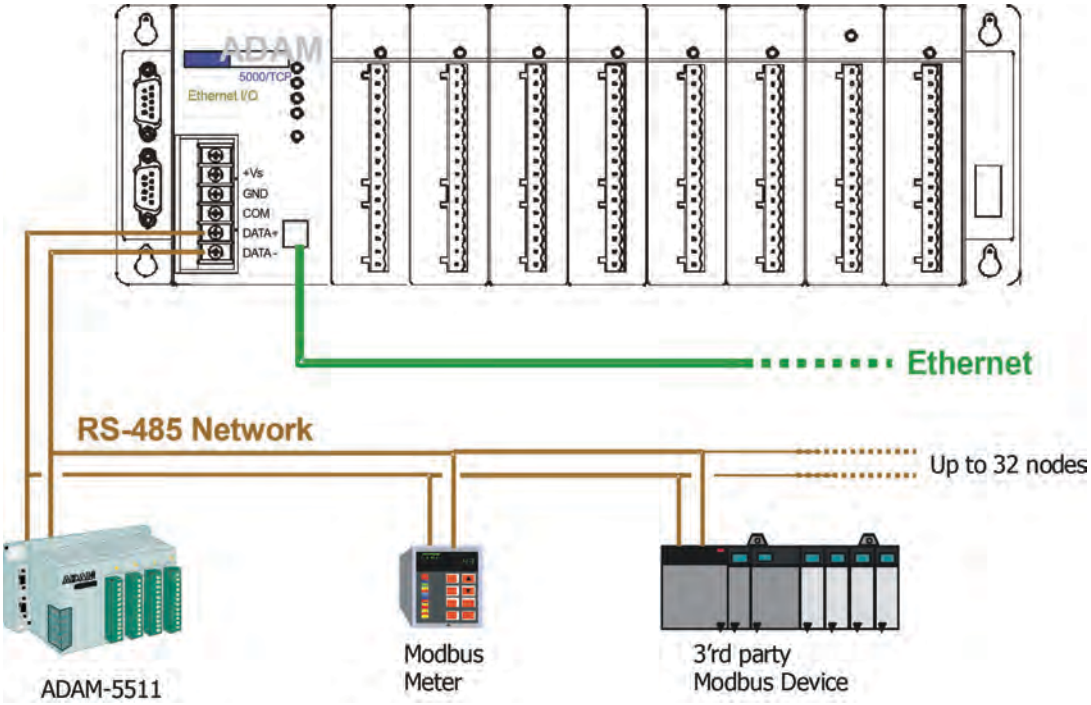


Figure 3.9 Serial Network Connection

Note! *The address of ADAM-5000/TCP Series on the RS-485 network will be always node 1. Any Modbus devices integrated in this network should be addressed from node 2 to 33.*

3.5 Assigning address for I/O Modules

Based on the Modbus standard, the addresses of the I/O modules you place into the ADAM-5000/TCP series system are defined by a simple rule. Refer to figures 3-9 to map the I/O address.

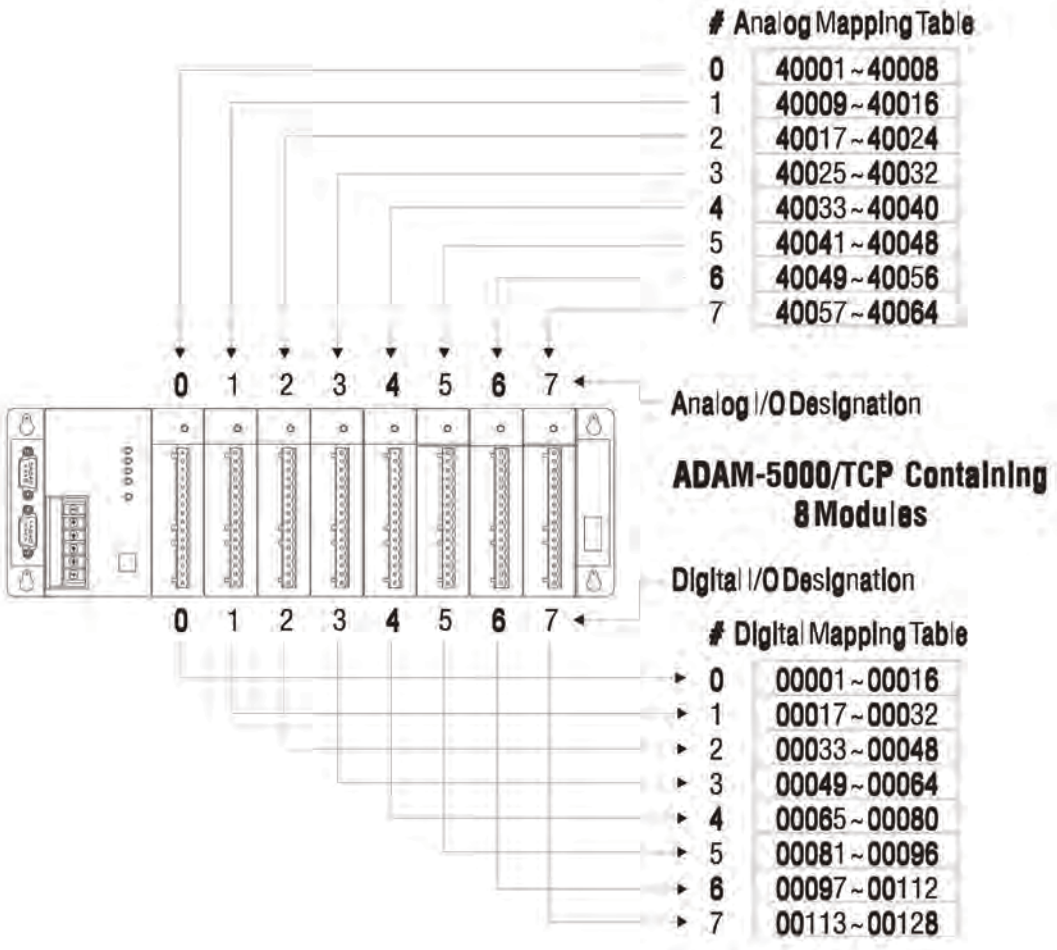


Figure 3.10 I/O Modules Address Mapping

For example, if there is a ADAM-5024 (4-channel AO Module) in slot 2, the address of this module should be 40017~40024.

Note! *ADAM-5080 is a special 4-channel counter module. The data type is designed as "unsigned long". When you insert an ADAM-5080 in slot 0, the address should be 40001, 40003, 40005 and 40007.*

Chapter 4

I/O modules

4.1 I/O Modules

This manual introduces the detailed specifications, functions, and application wiring of each ADAM-5000 I/O module.To organize an ADAM-5000 series and 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 20 types of ADAM-5000 I/O modules for various applications so far. The following table is the I/ O modules support list we provide to help you choose. **For more detailed specification and user guides, Refer the user manual of ADAM-5000 IO Module.**

Table 4.1: I/O Module Support List

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-5017P	8-ch Analog Input Module with Independent Input	Isolated
	ADAM-5018	7-ch. Thermocouple input	Isolated
	ADAM-5018P	7-ch Thermocouple Input Module with Independent	Isolated
Digital I/O	ADAM-5024	4-ch. AO	Isolated
	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 Module w/ LED	Isolated
	ADAM-5052	8-ch. DI	Isolated
	ADAM-5053S	32-ch Digital Input Module	Isolated
	ADAM-5055S	16-Ch Isolated DI/O Module w/ LED	Isolated
	ADAM-5056	16-ch. DO	Non-isolated
	ADAM-5056D	16-ch. DO W/LED	Non-isolated
	ADAM-5056S	16-Ch Sink Type Isolated DO Module w/ LED	Isolated
	ADAM-5056SO	16-Ch Source Type Isolated DO Module w/ LED	Isolated
	ADAM-5060	6-ch. Relay output	Isolated
Relay Output	ADAM-5069	8-Ch Power Relay Output Module w/ LED	Isolated
Counter/Frequency	ADAM-5080	4-ch. Counter/Frequency	Isolated
	ADAM-5081	4-ch High Speed Counter/Frequency Module	Isolated

Chapter 5

System Hardware Configuration

This chapter explains how to use Windows Utility to configure the ADAM-5000/TCP Series system for various applications. Users can learn the hardware connection, software installation, communication setting and every procedure for system configuration from these sections.

5.1 System Hardware Configuration

As we mentioned in chapter 3-1, you will need the following items to complete your system hardware configuration.

System Requirement

- Host computer
  - IBM PC compatible computer with 486 CPU (Pentium is recommended)
  - Microsoft XP/7 32, 64-bits or higher versions
  - At least 32 MB RAM
  - 20 MB of hard disk space available
  - VGA color monitor
  - 2x or higher speed CD-ROM
  - Mouse or other pointing devices
  - 10 or 100 Mbps Ethernet Card
- 10 or 100 Mbps Ethernet switch (at least 2 ports)
- Two Ethernet Cable with RJ-45 connector
- Power supply for ADAM-5000/TCP Series (+10 to +30 V unregulated) Make sure to prepare all of the items above, then connect the power and network wiring as figure 5-1.

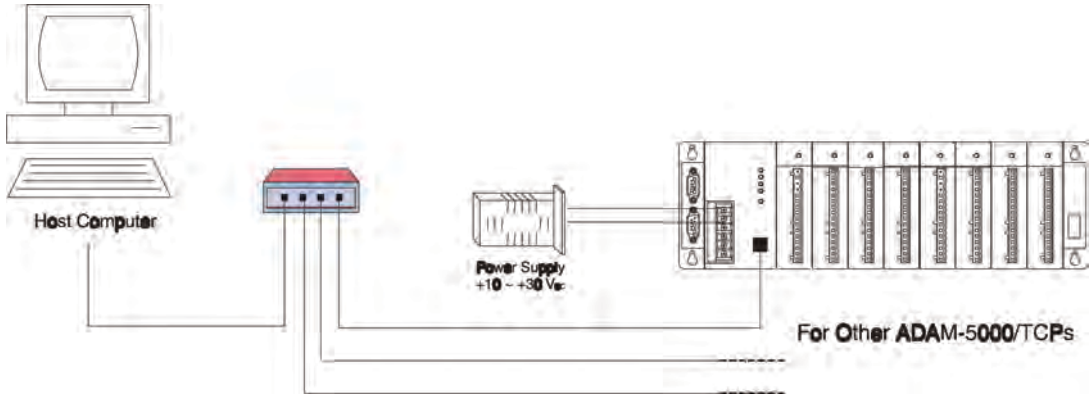


Figure 5.1 Hardware Configuration

5.2 Install Utility Software on Host PC

Please go to our web site to get the latest version of AdamApax .NET Utility.

Click AdamApax .NET Utility icon to execute the setup program. There will be a shortcut of the Utility executive program on Windows' desktop after completing the installation.

5.3 AdamApax .NET Utility Overview

The Windows Utility offers a graphical interface that helps you configure the ADAM-5000/TCP Series main unit and I/O modules. It is also very convenient to test and

monitor your DA&C System. The following guidelines will give you some brief instructions on how to use this Utility.

- Main Menu
- Ethernet Network Setting
- Adding Remote Station
- I/O Module Configuration
- Alarm Setting
- Firmware Update
- Security Setting
- Data Stream
- RS-485 Modbus Network Setting

5.3.1 Main Menu

Double Click the icon of ADAM/APAX.NET Utility shortcut, the operation screen will pop up as Figure 5-2.

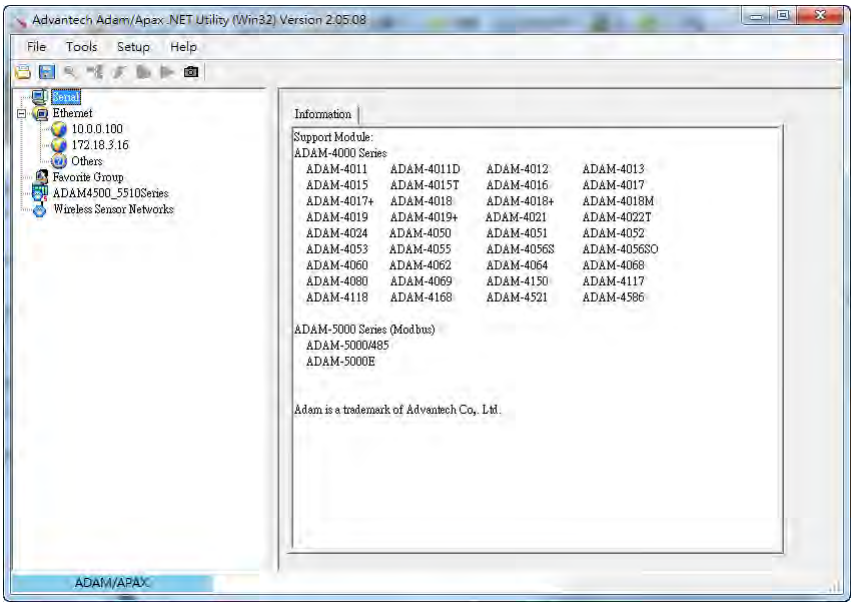
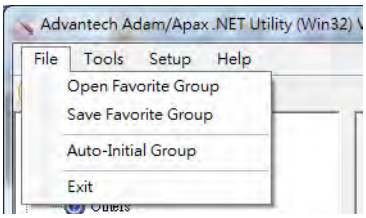


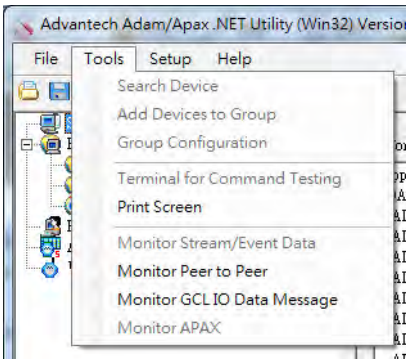
Figure 5.2 Operation Screen

The top of the screen consists of a function menu and a tool bar for commonly used-functions.

Function menu

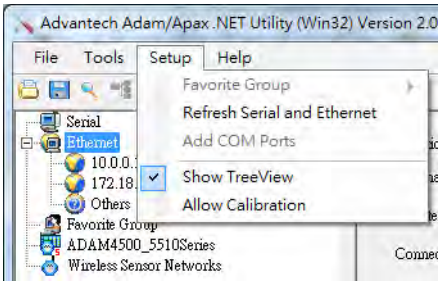


File / Exit will exit this Utility program.



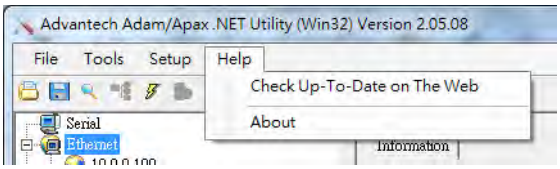
Tool content functions shown below:

Search Device:	Search all ADAM and APAX units in the specific Ethernet domain. (the same as the host PC's Ethernet domain).
Terminal for Command Testing:	Call up the operations screen of Terminal emulation to do the request / response command execution.
Monitor Data Stream:	Call up the monitoring screen of the streaming data from the specific ADAM-5000/TCP.



Setup contents functions shown below:

Refresh Serial and Ethernet:	Refresh the specific interface to get an update of the device.



About menu item shows information about software version, release date, and support modules.

Tool Bar

There are five push buttons in the tool bar.

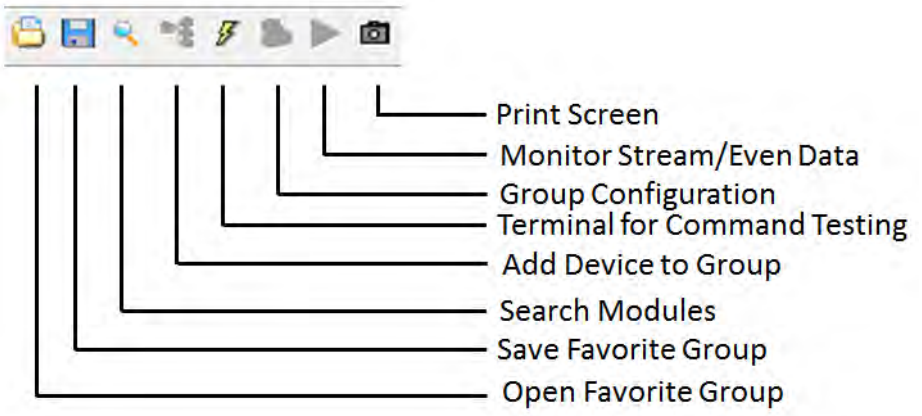


Figure 5.3 Tool Bar

5.3.2 Ethernet Network Setting

As soon as you click the search button, it will search all ADAM-5000/TCP Series on the host PC's domination Ethernet network automatically. Then the tree-structure display area will appear with the searched items and relative IP addresses.

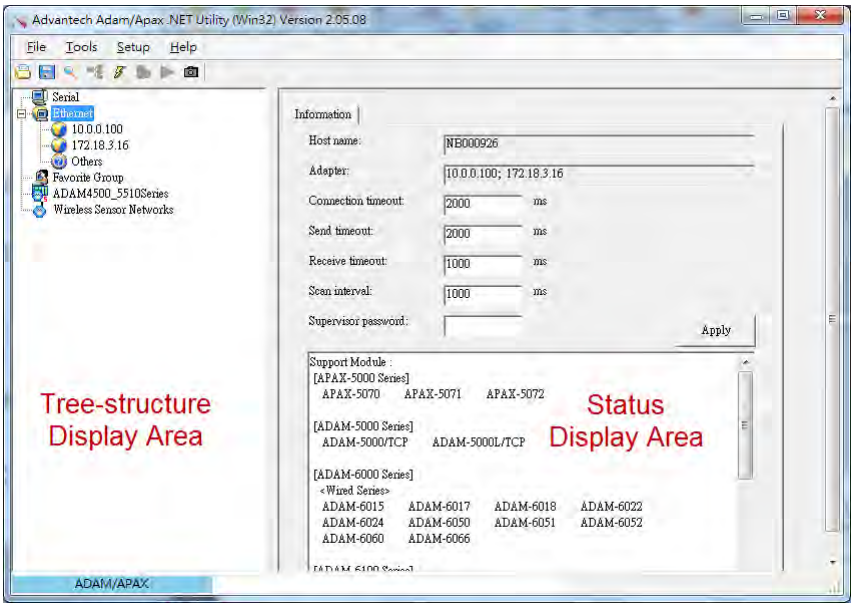


Figure 5.4 Network Setting

In Figure 5-4, there is host PC information in the status display area, including host name and IP address.

Since the utility software detects the ADAM-5000/TCP series on the network, the user can begin to setup each ADAM-5000/TCP series station individually following these steps.

Step 1. Choose any one station, all I/O modules plugged into the main unit will be listed on the tree-structure display area. Meanwhile, the "Device Name" and "Device Description" are editable by the operator.

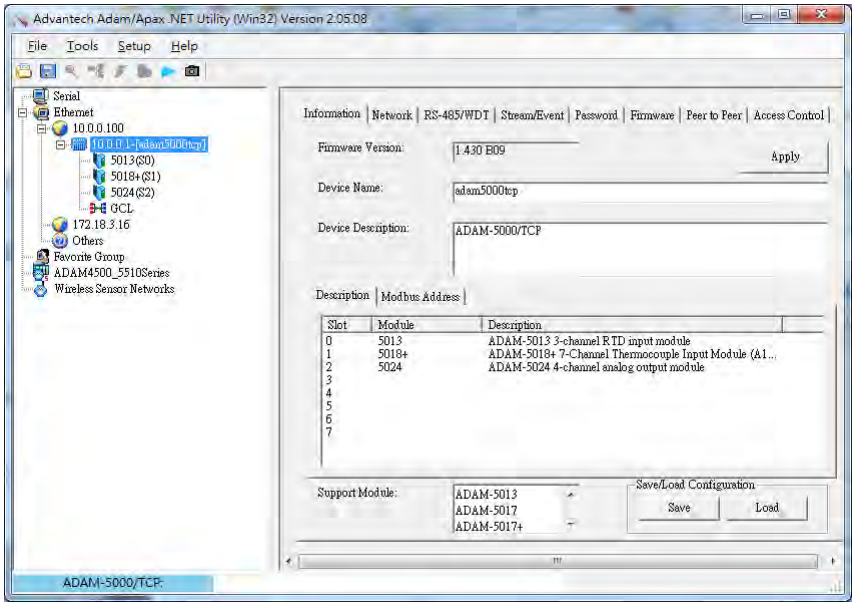


Figure 5.5 Define Device Name and Description

Step 2.Click the Network tip to configure the TCP/IP network setting.

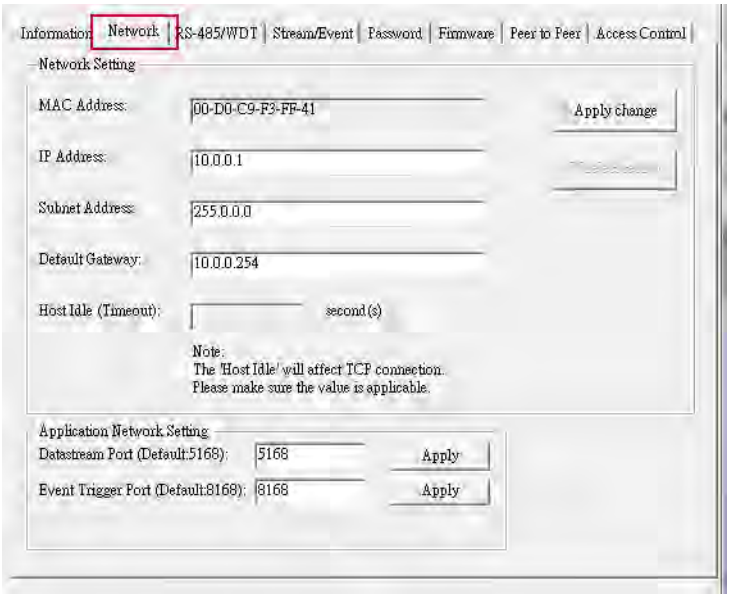


Figure 5.6 TCP/IP Network setting

MAC Address: This is also called the Ethernet address and needs no further configuration.

IP Address, Subnet Mask, Default Gateway: The IP address identifies your ADAM-5000/TCP Series device on the global network. Each ADAM-5000/TCP has same default IP address 10.0.0.1. Therefore, please do not initial many ADAM-5000/TCP Series at the same time to avoid the Ethernet conflict.

If you want to configure the ADAM-5000/TCP Series in the host PC's domain network, only the IP address and Subnet Mask will need to be set (host PC and ADAM-5000/TCP Series must belong to same subnet mask). If you want to configure the ADAM-5000/TCP Series via Internet or other network domains, you have to ask your

network administrator to obtain a specific IP and Gateway address then configure each ADAM-5000/ TCP Series with an individual setting.

Note! *There are several conditions you need to be sure of before adding a remote ADAM-5000/TCP Series system in the windows Utility.*

1. *Be sure the specific IP exists and is available.*
2. *Be sure to complete the network linkage for both sides.*
3. *Be sure to adjust the best timeout setting.*
4. *Even if you are not sure whether the communication is workable or not, there is also a "PING" function for testing the network connection.*

5.3.3 I/O Module Configuration

Digital Input Output Module

ADAM-5000 Digital Modules include ADAM-5050/5051(D)/5051S/5052/5055S/5056(D)/5056S/5060/5068/5069, users can read following information from the Utility.

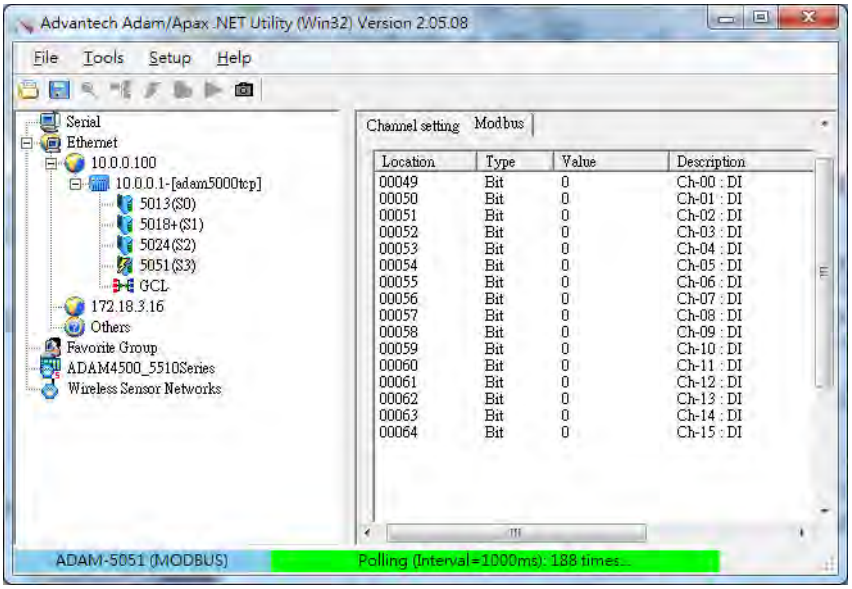


Figure 5.7 Digital I/O Module Configuration

Location: Standard Modbus address.Windows Utility shows the Modbus mapping address of each I/O channel. (Refer to chapter 3.5 Assigning address for I/O Modules) The addresses will be the indexes for applying to the database of the HMI or OPC Server.

Type: Data Type of the I/O channel. The data *type* of Digital I/O modules is always "Bit".

Value: The current status on each channel of I/O Module.
The value of digital I/O modules could be "0" (OFF) or "1" (ON).

Description: Describes the channel numbers and I/O types of the specific module.

In addition to monitoring the current DI/DO status, the Windows Utility offers a GUI interface as shown in figure 5-8. You can read the Digital input status through the

change of the indicator icons. Conversely, you can write the digital output status by clicking the indicator icons.

Figure 5.8 Operation and Indication Icons

Note!

1. *The indicator icons are only available to click for digital output channel.*
2. *The hexadecimal code will be calculated automatically for any status.*

Analog Input Module

ADAM-5000 Analog Input Modules include ADAM-5013/5017(H)/5018, users can read following information from the Utility.

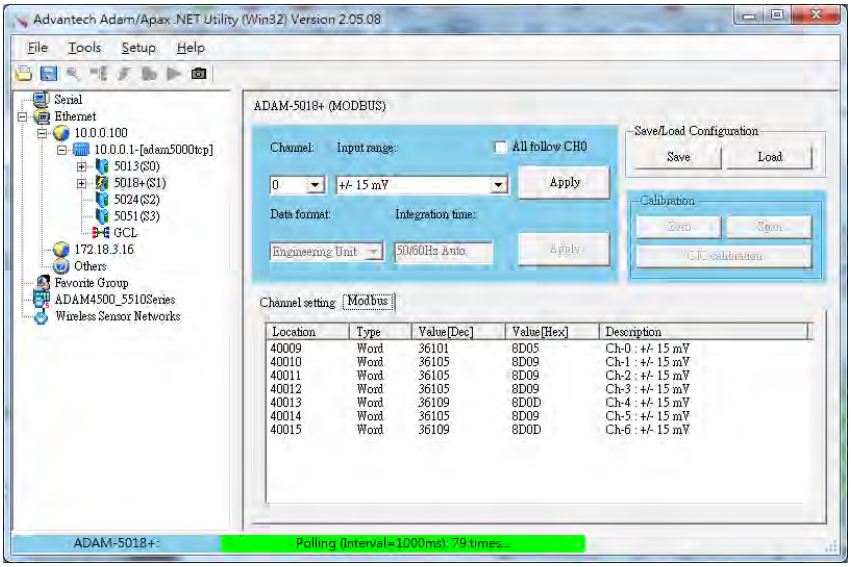


Figure 5.9 Current Analog Input Status

Location: Standard Modbus address. (Refer to chapter 3.5 Assigning addresses for I/O module).

Type: Data type of the I/O channel. The data type of analog Input modules is always “word”.

Value: The current status on each channel of I/O modules. Windows Utility provides both decimal and hexadecimal values used for different applications.

Description: Describes the channel numbers, sensor types, and measurement ranges of the specified module. Before acquiring the current data of an analog input module, you have to select the input range and integration time. Then the input data will be scaled as the specified range with the engineer unit.

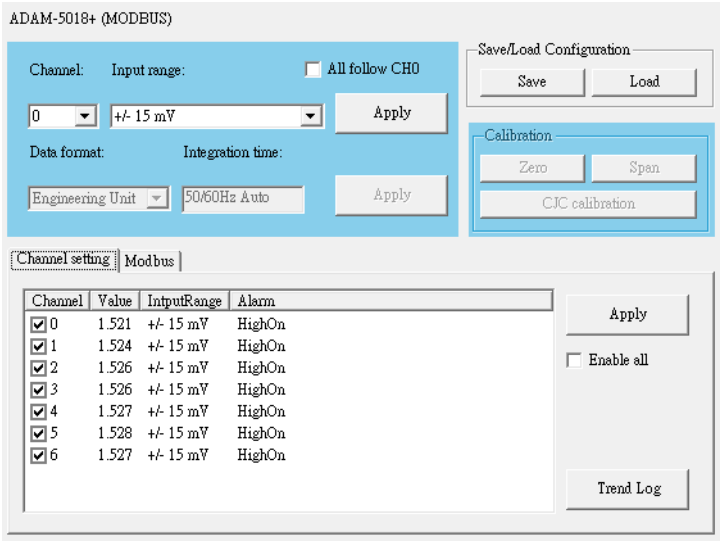


Figure 5.10 Setting range and integration time

Note! *Windows Utility allows the user to Enable / Disable the current status display.*

Analog Output Module

Selecting an ADAM-5024 Analog Output Module, users can read Value and Output Range information in the Channel setting table. ADAM-5024 is designed with four different outputs channels, so there are four channel configuration screens for signal range and output value setting in the Utility. Once the setting value sends out, the system will read back the value immediately to guarantee a correct analog output signal.

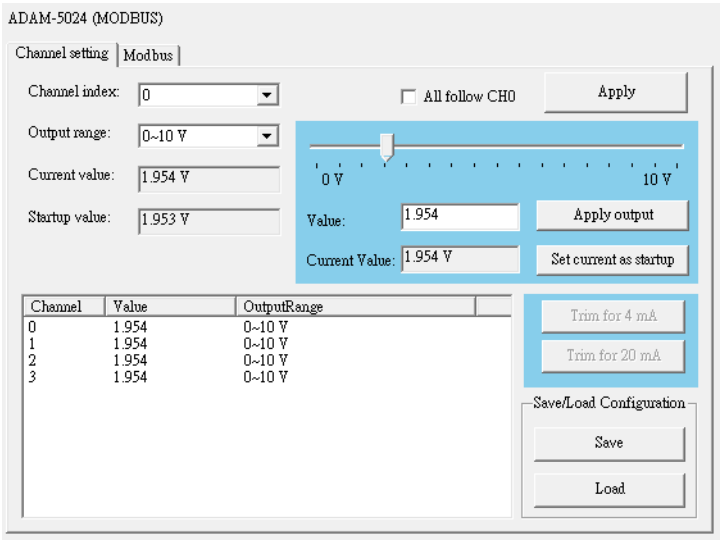


Figure 5.11 Analog Module Configuration Screen

Note! *Initial Setting function: Adjust a initial output value if you want to setup the specified channel and click the **set current as startup** button, the channel will output the same value each time when the system is initialized.*

Counter/Frequency Module

Selecting an ADAM-5080 Counter/Frequency Module, users also can read the information about location, type, value, and description from four individual channel configuration screens.

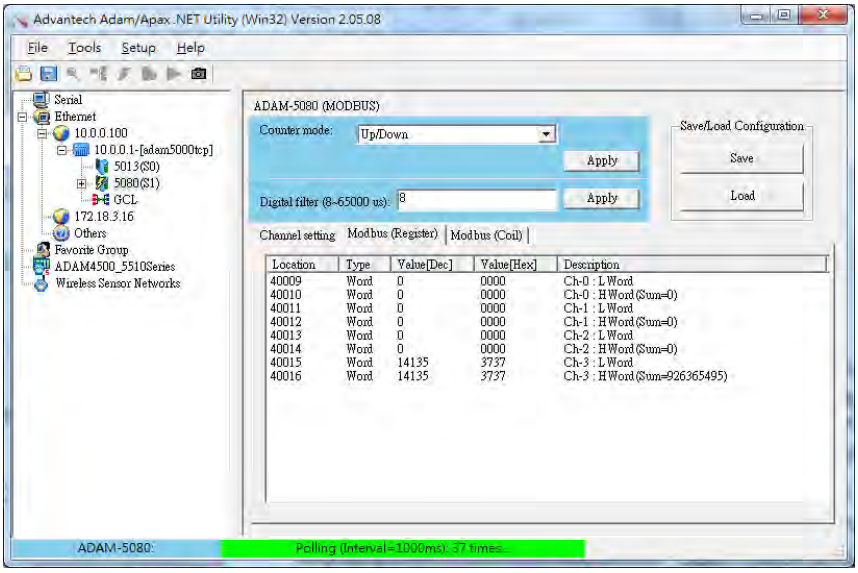


Figure 5.12 Counter/Frequency Module Configuration

However, the ADAM-5080 is a special module. Each channel is composed of an unsigned long and four bits. For example, if ADAM-5080 is plugged into Slot 6 on the ADAM-5000/ TCP system, the address locations should be:

	Channel 0	Channel 1	Channel 2	Channel 3
Unsigned Long	40049	40051	40053	40055
1st Bit	00097	00101	00105	00109
2nd Bit	00098	00102	00106	00110
3rd Bit	00099	00103	00107	00111
4th Bit	00100	00104	00108	00112

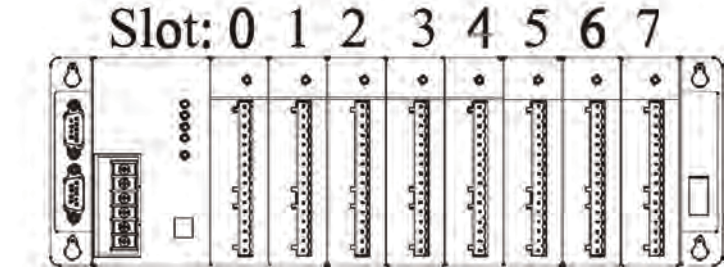


Figure 5.13 Location of Counter/Frequency Module

Note! ***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.*

3rd bit: *Normal OFF "0", only ON "1" when counter overflow. Users can write "0" to clear the overflow flag.*

4th bit: *Non used.*

5.3.4 Alarm Setting To satisfy the needs of various applications,

ADAM-5000/TCP Series system provides Alarm setting function for Analog Input and Counter Module. Users can set High/Low limit value to identify the alarm status and trigger a digital output as an event handling function.

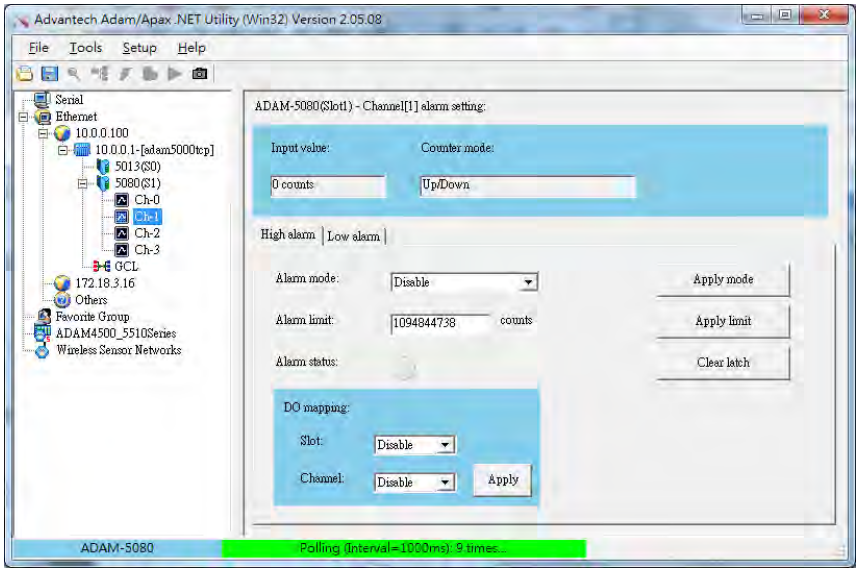


Figure 5.14 Alarm Setting for Analog Input and Counter Modules

There are three alarm types in Analog Input Modules:

Disable: ADAM-5000/TCP Series dose not executive alarm diagnosing function.

Momentary: When the Input value is over or under the High/Low limit, the alarm signal will be sent only once.

Latch: When the input value is over or under the High/Low limit, the alarm signal will be latched till clicking the “Clear Latch” button.

Note! *The alarm types of ADAM-5080 includes “Disable” and “Latch only.*

5.3.5 Firmware Update

ADAM-5000/TCP Series supports all ADAM-5000 series I/O modules and necessary operating functions so far. But Advantech always provides better hardware and software functions to improve the DA&C systems. Therefore, users will need to upgrade the firmware of ADAM-5000/TCP series to get the best performance. Select the Firmware Upgrade tab and click **Open** to find the specific firmware (*.bin) to upgrade.

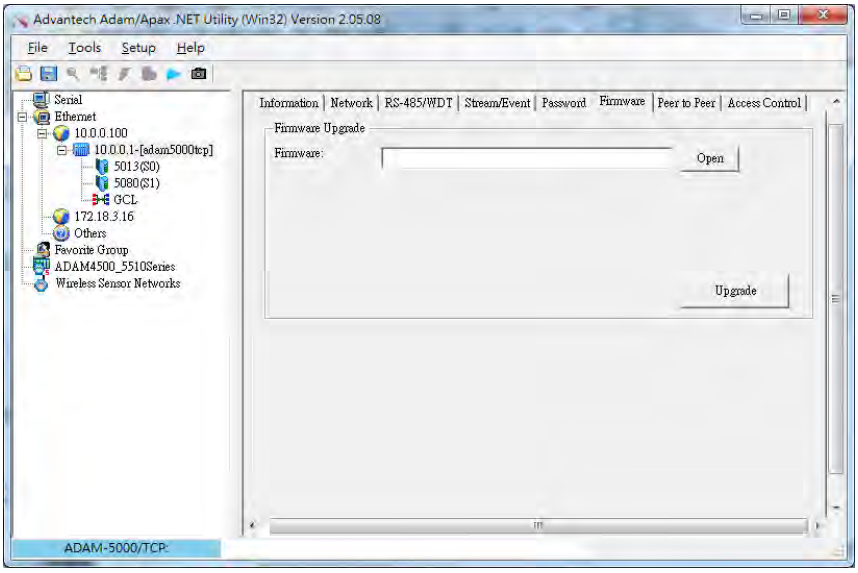


Figure 5.15 Firmware Upgrade

Click the upgrade button, then the new firmware will be downloaded into the ADAM-5000/TCP Series system.

5.3.6 Security Setting

Ethernet brings great benefits of speed and integration, there also exists network security risks. For this reason, the security protection is built into ADAM-5000/TCP Series. Once a user set the password into the ADAM-5000/TCP Series firmware, the important system configurations (Network, Firmware, Password) are only allowed to be changed via password verification.

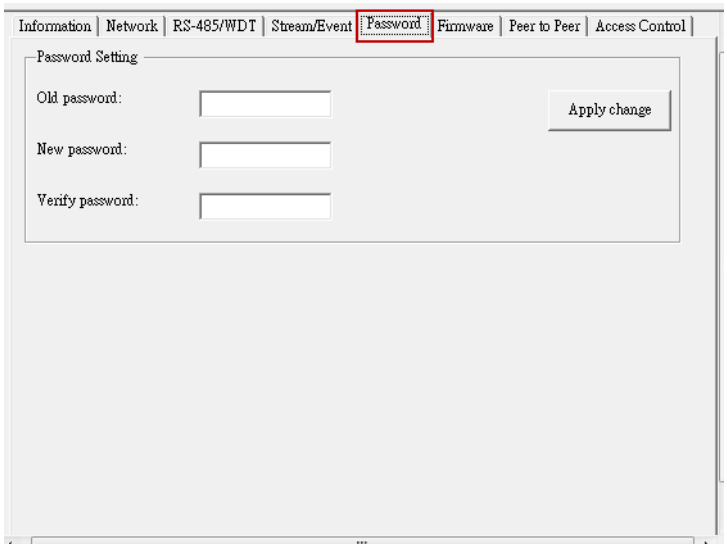


Figure 5.16 Password Setting

Note! *The default password of ADAM-5000/TCP Series is "**00000000**". Please make sure to change and keep the correct password by yourself. If you lose it, please contact Advantech's technical support center for help.*

5.3.7 Terminal Emulation

You can issue commands and receive a response by clicking the Terminal button on the tool bar. There are two kinds of command format supported by this emulation function. Users can choose ASCII or Hexadecimal mode as their communication base. If ASCII mode has been selected, the Windows Utility will translate the request and response string both in Modbus and ASCII format.

For example, select ASCII mode and key-in the ASCII command “$01M” (module name), then click **Send**.

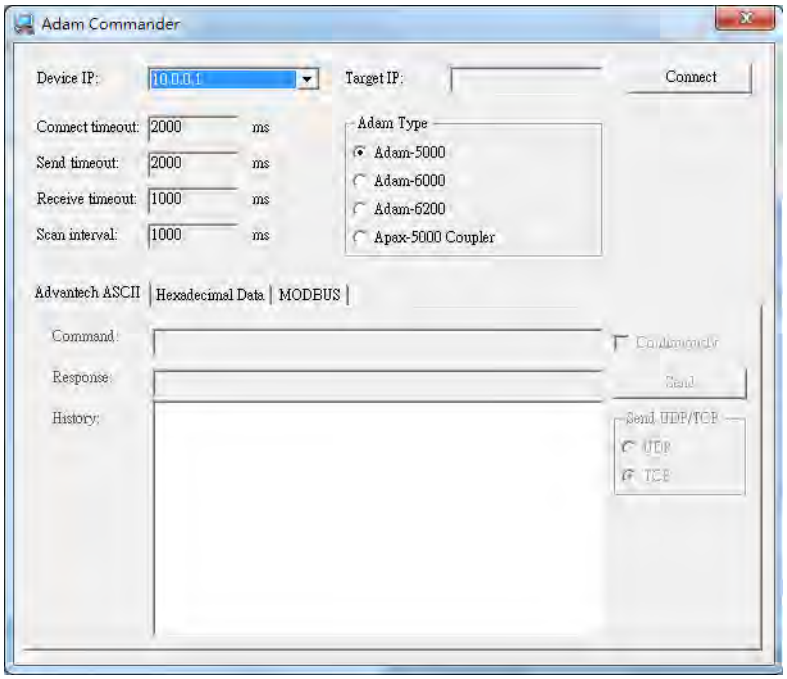


Figure 5.17 Command Emulation

5.3.8 Data Stream

Data Stream Configuration

In addition to TCP/IP communication protocol, ADAM-5000/TCP Series supports UDP communication protocol to regularly broadcast data to specific host PCs. Click the tip of Data stream, then configure the broadcasting interval and the specific IPs which need to receive data from the specific ADAM-5000/TCP Series. This UDP Data Stream function broadcasts up to 8 host PCs simultaneously, and the interval is user-defined from 50ms to 10 hours.

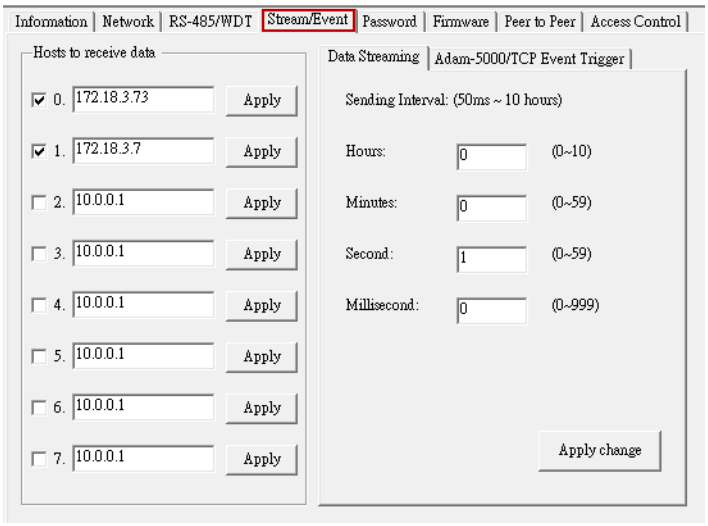


Figure 5.18 Data Stream Configuration

Data Stream Monitoring

After finishing the configuration of Data Stream, you can select the item “Monitor Data Stream” in the function bar or click icon to call up operation display as shown in Figure 5-19.

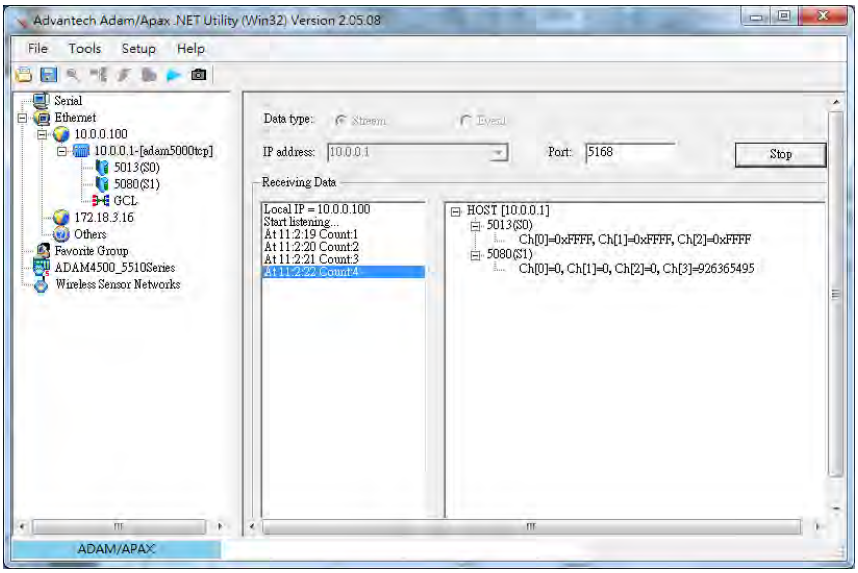


Figure 5.19 Data Stream Monitoring

Select the IP address of the ADAM-5000/TCP Series you want to read the data of, then click “Start” button. The Utility software will begin to receive the stream of data shown on this operation display.

5.3.9 Data Gateway Setting

ADAM-5000/TCP Series is designed with an RS-485 Modbus Interface. As a Data Gateway, it integrates serial Modbus devices into Ethernet applications easily.

Click the tip of “RS-485/Modbus” to configure the RS-485 network setting following these steps below.

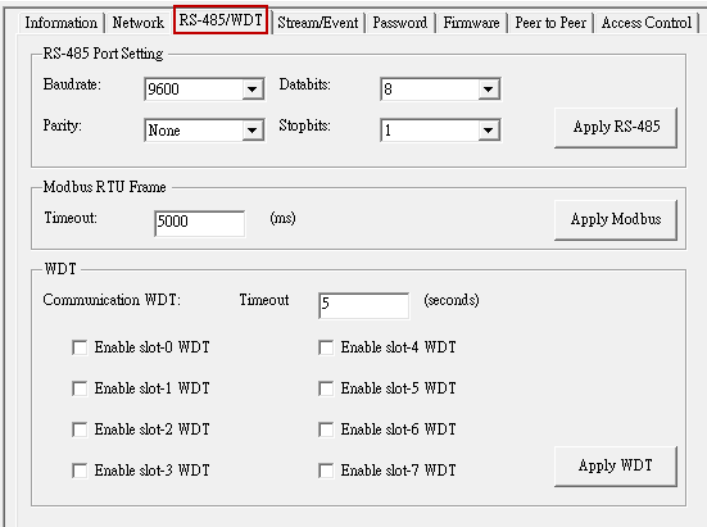


Figure 5.20 RS-485 Modbus Network Setting

1. Define the parameters of the network, including parity, stop bit, baud rate, (300~115200bps) and timeout.
2. Click the Apply button, the password verification dialog block will pop up.
3. Key in your specific password and click OK, The setting is now done.

Chapter 6

Programming

6.1 Introduction

After completing the system configuration, you can begin to plan the application program. This chapter introduces two programming tools for users to execute system data acquisition and control. The DLL drivers and command sets provide a friendly interface between your applications and ADAM-5000/TCP Series system.

6.2 DLL (Dynamic Link Library) Driver

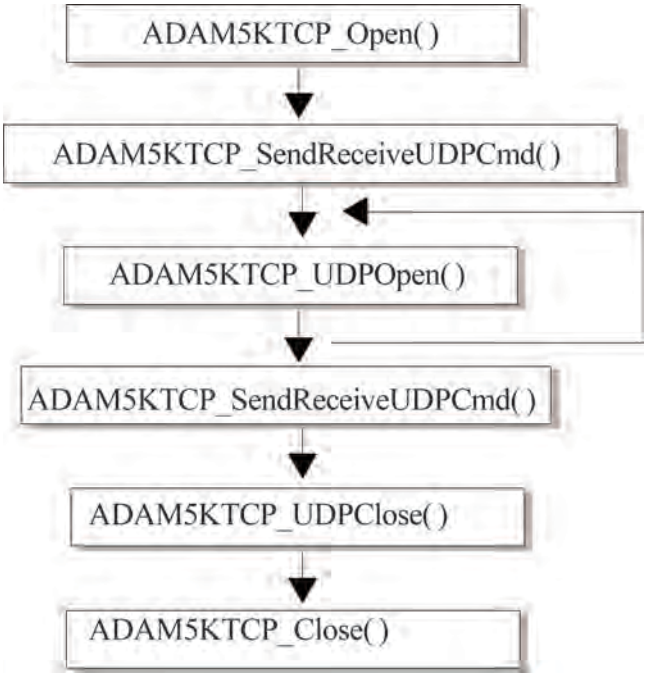
The Dynamic Link Library (DLL) enables you to quickly and easily write Windows applications for ADAM-5000/TCP Series systems. The library supports Borland C, Delphi, Visual C++, and Visual Basic. Since ADAM-5000/TCP systems communicate with a host computer through Ethernet, no additional driver needs to be installed. The DLL includes all necessary function calls to utilize the ADAM-5000/TCP Series systems to their fullest extent. In the same path with “ADAM 5000TCP Series” after completing S/W installation, you’ll find the relational example files for each kind of programming languages after setup the Windows Utility program. You can customize the source code to create your own tailor-made ADAM-5000/TCP Series setup program or monitoring system.

6.2.1 Index

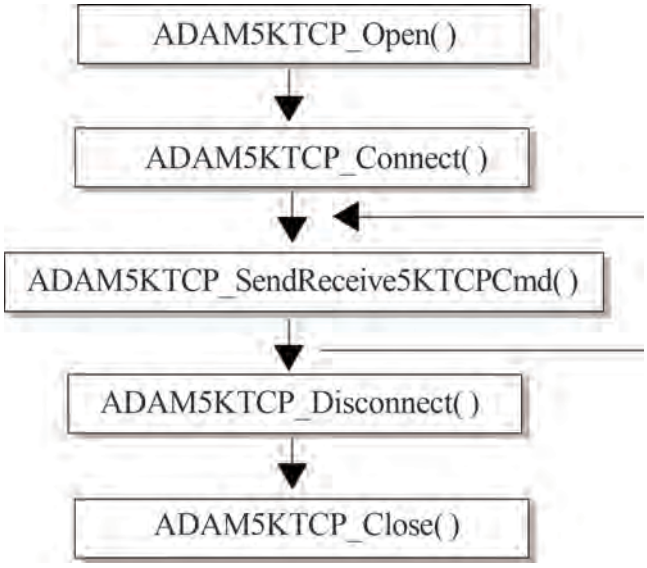
Function Libraries	Pages
ADAM5KTCP_Open	56
ADAM5KTCP_Close	56
ADAM5KTCP_Connect	56
ADAM5KTCP_Disconnection	56
ADAM5KTCP_GetDLLVersion	57
ADAM5KTCP_ReadReg	57
ADAM5KTCP_WriteReg	57
ADAM5KTCP_ReadCoil	58
ADAM5KTCP_WriteCoil	58
ADAM5KTCP_SendReceive5KTCPCmd	59
ADAM5KTCP_Add5KTCPForStream	59
ADAM5KTCP_ReadStreamData	60
ADAM5KTCP_ReadAlarmInfo	60
ADAM5KTCP_StartStream	61
ADAM5KTCP_StopStream	61
ADAM5KTCP_SetStreamAlarmState	62
ADAM5KTCP_Debug	62
ADAM5KTCP_UDPOpen	63
ADAM5KTCP_UDPClose	63
ADAM5KTCP_SendReceiveUDPCmd	63

6.2.2 Programming Flows

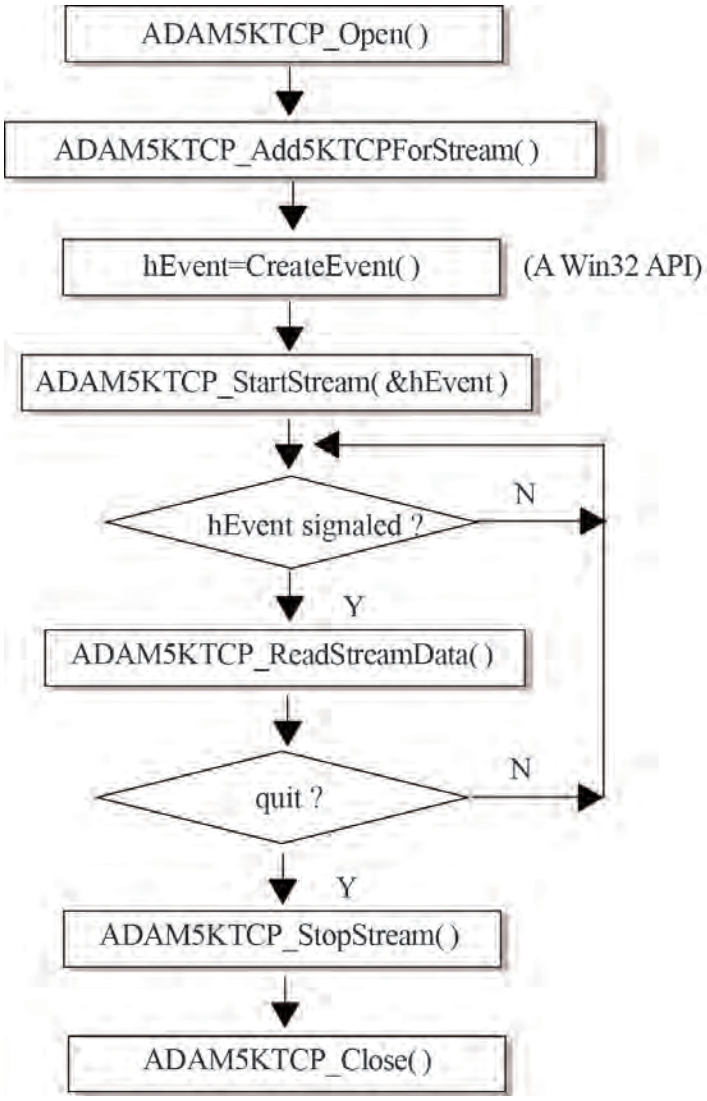
Sending a command and receiving a response by UDP.



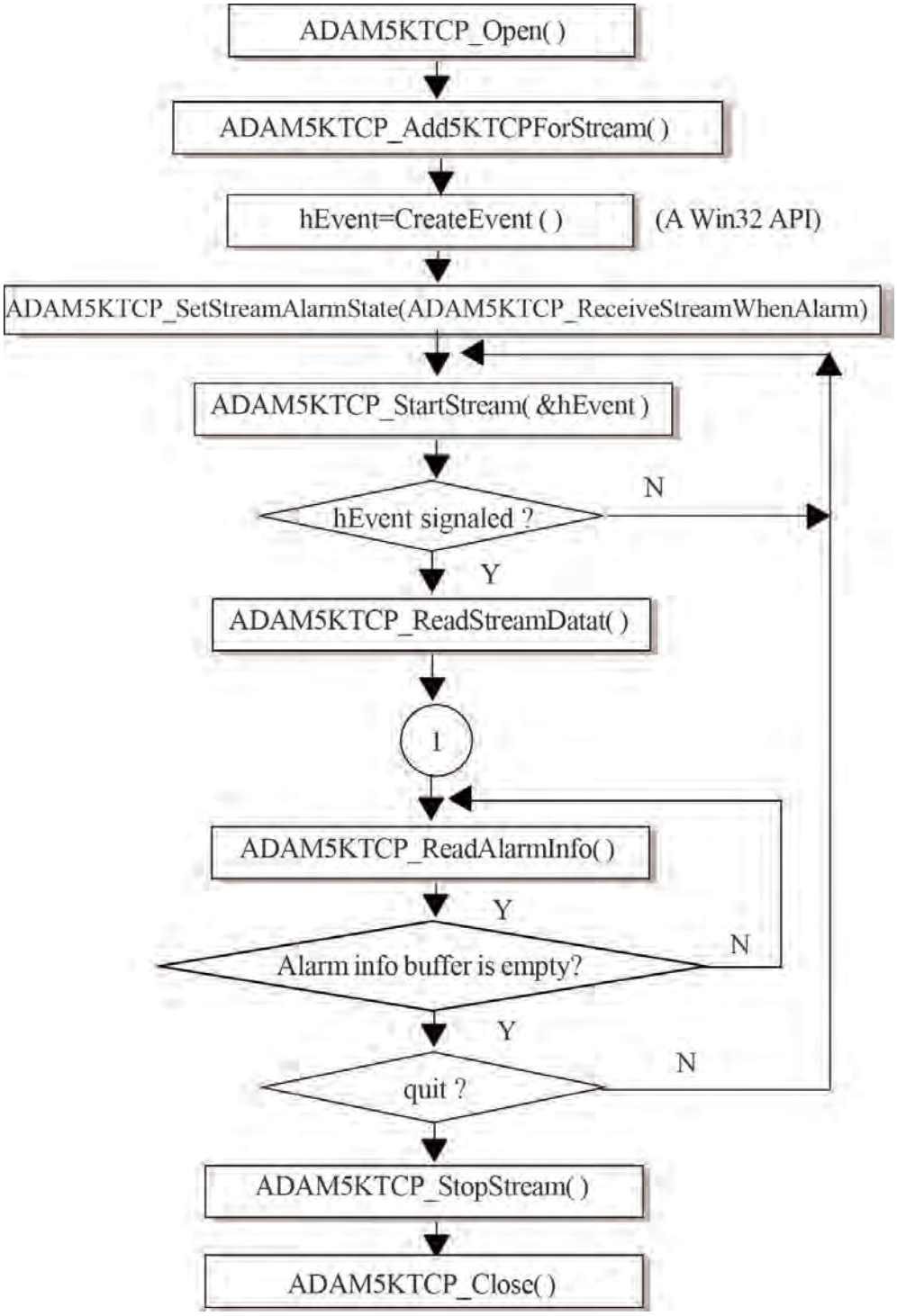
Sending a command and receiving a response by TCP



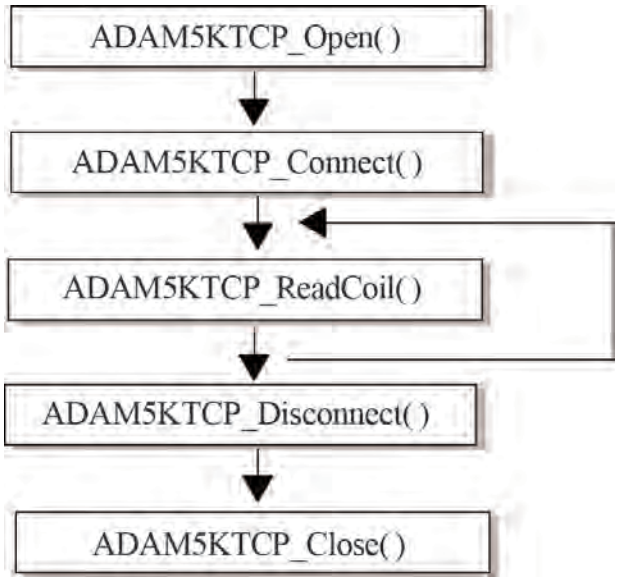
Receiving a data stream coming from ADAM-5000/TCP Series (s)



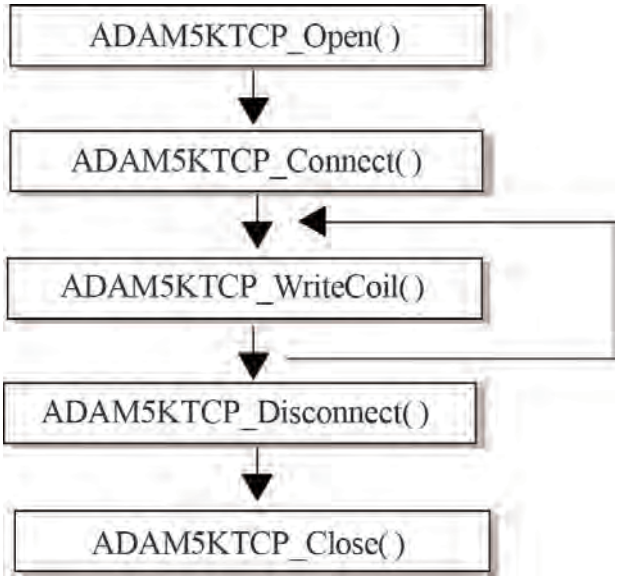
Receiving alarm information from ADAM-5000/TCP(s)



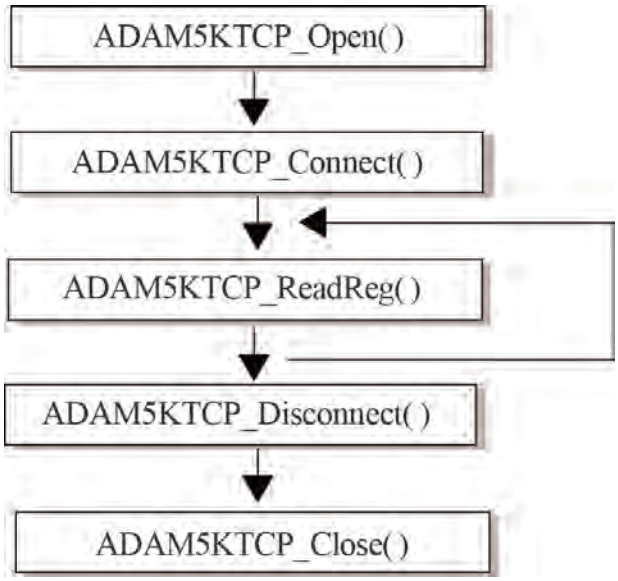
Reading coil values.



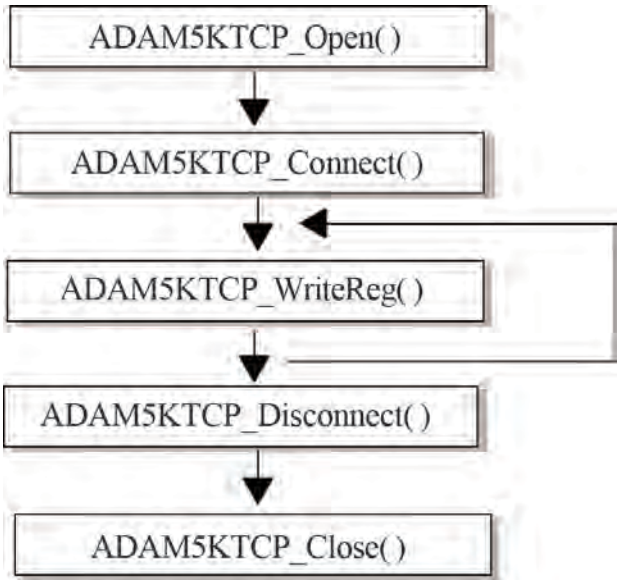
Writing values to coil.



Reading values from a holding register.



Writing values to a holding register.



6.2.3 Function Descriptions

ADAM5KTCP_Open

Description: Initiate the “adam5ktcp.dll” for use.

Syntax: int ADAM5KTCP_Open(void);

Parameters: void

Return: Refer to Chapter6-2-4 “Return Codes” for more detail information.

ADAM5KTCP_Close

Description: Terminates using the “adam5ktcp.dll”.

Syntax: void ADAM5KTCP_Close(void);

Parameters: void

Return: void

ADAM5KTCP_Connect

Description: Establishes a Windows Sockets connection in a specified ADAM-5000/TCP system.

Syntax: int ADAM5KTCP_Connect(char szIP[ ], unsigned short port, int iConnection-Timeout, int iSendTimeout, int iReceiveTimeout);

Parameters:

szIP[in]:	The IP Address of the ADAM-5000/TCP that is to be connected
port[in]:	The TCP/IP connection port used with Modbus/TCP,
iConnectionTime-out[in]:	The specified timeout interval for connecting to the ADAM-5000/TCP
iSendTimeout[in]:	The specified timeout interval for sending a command to the ADAM-5000/TCP
iReceiveTimeout[in]:	The specified timeout interval for receiving response from the ADAM-5000/TCP

Return: Refer to Chapter 6.2.4 “Return Codes” for more detailed information

ADAM5KTCP_Disconnect

Description: Disconnect the Windows Sockets connection of the specified ADAM-5000/TCP

Syntax: void ADAM5KTCP_Disconnect(void);

Parameters: void

Return: Refer to Chapter 6.2.4 “Return Codes” for more detailed information

DAM5KTCP_GetDLLVersion

Description: Read the version of ADAM-5000/TCP DLL driver

Syntax: int ADAM5KTCP_GetDLLVersion(void);

Parameters: void

Return: 0x150 means Version 1.50

ADAM5KTCP_ReadReg

Description: Reads the holding register value at a specified range described in parameters.

Syntax: int ADAM5KTCP_ReadReg(char szIP[], WORD wID, WORD wStartAddress, WORD wCount, WORD wData[]);

Parameters:

szIP[in]:	The IP Address of the ADAM-5000/TCP that to be connected
wID[in]:	The specific device ID for an Modbus/TCP device. The ADAM-5000/TCP is always assigned as 1
wStartAddress[in]:	The starting address to be read
wCount[in]:	How many holding registers to be read
wData[out]:	An unsigned 16 bit array that stores the read holding register values

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_WriteReg

Description: Write the holding register value at a specified range described in the parameters.

Syntax: int ADAM5KTCP_WriteReg(char szIP[], WORD wID, WORD wStartAddress, WORD wCount, WORD wData[]);

Parameters:

szIP[in]:	The IP Address of the ADAM-5000/TCP that to be connected
wID[in]:	The specific device ID for an Modbus/TCP device. The ADAM-5000/TCP is always assigned as 1
wStartAddress[in]:	The starting address to be written
wCount[in]:	How many holding registers to be written
wData[out]:	An unsigned 16 bit array that stores write values to a holding register

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_ReadCoil

Description: Reads the coils value at a specified range described in parameters.

Syntax: int ADAM5KTCP_ReadCoil(char szIP[], WORD wID, WORD wStartAddress, WORD wCount, BYTE byData[]);

Parameters:

szIP[in]:	The IP Address of the ADAM-5000/TCP that to be connected
wID[in]:	The specific device ID for an Modbus/TCP device. The ADAM-5000/TCP is always assigned as 1
wStartAddress[in]:	the starting address that to be read
wCount[in]:	How many coils to be read
byData[out]:	An 8-bit array that stores read coils

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_WriteCoil

Description: Writes coil values at a specified range described in parameters.

Syntax: int ADAM5KTCP_WriteCoil(char szIP[], WORD wID, WORD wStartAddress, WORD wCount, BYTE

Parameters:

szIP[in]:	The IP Address of the ADAM-5000/TCP that to be connected
wID[in]:	The specific device ID for an Modbus/TCP device. The ADAM-5000/TCP is always assigned as 1
wStartAddress[in]:	The starting address that to be written
wCount[in]:	How many coils to be written
byData[out]:	An unsigned 8-bit array that stored values written to the coil

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_SendReceive5KTCPCmd

Description: This function is designed for user's convenience, accepting the ASCII format string as a command. Then transforming it to meet the Modbus/TCP specifications.

Syntax: int ADAM5KTCP_SendReceive5KTCPCmd(char szIP[], char szSendToTCP[], char szReceiveFromTCP[], char szModbusSend[], char szModbusReceive[]);

Parameters:

szIP[in]:	The IP Address of the ADAM-5000/TCP that to be connected
szSendToTCP[in]:	The ASCII format string that send to a ADAM-5000/ TCP
szReceiveFromTCP[out]:	The ASCII format string that response from a ADAM-5000/TCP
szModbusSend[out]:	The Modbus/TCP format string that send to a ADAM-5000/TCP
szModbusReceive[out]:	The Modbus/TCP format string that response from a ADAM-5000/TCP

Return: Refer to Chapter 6.2.4 "Return Codes" for more detail information

ADAM5KTCP_Add5KTCPForStream

Description: Assigns a specified ADAM-5000/TCP to send stream data to the PC

Syntax: int ADAM5KTCP_Add5KTCPForStream(char szIP[]);

Parameters: szIP[in]: the IP Address of the ADAM-5000/TCP that assign to send stream data to the PC

Return: Refer to Chapter 6.2.4 "Return Codes" for more detail information

ADAM5KTCP_ReadStreamData

Description: Receives a data stream that comes from a specific ADAM-5000/TCP

Syntax: int ADAM5KTCP_ReadStreamData(char szIP[], struct _StreamData *pStreamData);

Parameters:

szIP[in]: Specifies the IP Address for a user to receive the stream data

*pStreamData[out]: the stream data stored in _StreamData structure Refer to Chapter 6-2-5 “Data Structure” for more detail information about _StreamData structure.

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_ReadAlarmInfo

Description: Receive alarm information that comes from the specific ADAM-5000/TCP

Syntax: int ADAM5KTCP_ReadAlarmInfo (struct _AlarmInfo *pAlarmInfo);

Parameters:

*pAlarmInfo[out]: The alarm information stored in _AlarmInfo structure Refer to Chapter 6-2-5 “Data Structure” for more detail information about _AlarmInfo structure.

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_StartStream

Description: Instruct the PC to start receiving stream data from the ADAM-5000/TCP

Syntax: int ADAM5KTCP_StartStream (HANDLE *EventFromApp);

Parameters:

*EventFromApp: The event object that would pass down to ADAM5KTCP.DLL This event object would be signaled either a stream data send to PC or an alarm status change in ADAM-5000/TCP.Refer to ADAM5KTCP_SetStream AlarmState for more detail information.

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_StopStream

Description: Instruct the PC to stop receiving stream data

Syntax: int ADAM5KTCP_StopStream( );

Parameters: void

Return: void

ADAM5KTCP_SetStreamAlarmState

Description: Set the criterion to signal the event object

Syntax: int ADAM5KTCP_SetStreamAlarmState(WORD wStreamAlarmState);

Parameters:

wStreamAlarmState[in]: When assigned to ADAM5KTCP_Receive StreamIngoreAlarm: means the ADAM5KTCP.DLL always signals event object when any stream data comes from an ADAM-5000/TCP. Then the application can receive the stream data by calling “ADAM5KTCP_ReadStreamData()” function.
When assigned to ADAM5KTCP_Receive Stream WhenAlarm: means ADAM5KTCP.DLL only signals event object when a alarm status is triggered. Then the application can receive the alarm information about the ADAM-5000/TCP by calling “ADAM5KTCP_ReadAlarmInfo()” function.

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_Debug

Description: Trace the executive information about streaming data mechanism in ADAM5KTCP.DLL
(It is convenient to troubleshooting of user’s applications.)

Syntax: int ADAM5KTCP_Debug(int *iMatchIndex, int *iReceiveCount, int *iThreadRun, int *iTotalStream, char szFromIP[]);

Parameters:

*iMatchIndex[out]: Indicates which ADAM-5000/TCP caused the signaling event object
0 means the first ADAM-5000/TCP, 1 means second,
2 means third, and so on. The ordinal is implied when calling “ADAM5KTCP_Add5KTCPForStream()” function.

*iReceiveCount[out]: Counts how many stream data have arrival

*iThreadRun[out]: Indicates the working thread status in ADAM5KTCP.DLL

*iTotalStream[out]: Reserved

szFromIP[out]: Specifies the IP Address of ADAM-5000/TCP which sends the stream data.

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_UDPOpen

Description: Opens a UDP socket and sets the timeout of send/receive interval to prepare send a command to ADAM-5000/TCP by UDP.

Syntax: int ADAM5KTCP_UDPOpen(int iSendTimeout, int iReceiveTimeout);

Parameters:

iSendTimeout[in]: The specified timeout interval for sending a command string to the ADAM-5000/TCP by UDP.

iReceiveTimeout[in]: The specified timeout interval for receiving a response string from the ADAM-5000/TCP by UDP.

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_UDPClose

Description: Closes the UDP socket that has been opened by “ADAM5KTCP_UDPOpen()”.

Syntax: int ADAM5KTCP_UDPClose();

Parameters: Void

Return: Refer to Chapter 6.2.4 “Return Codes” for more detail information

ADAM5KTCP_SendReceiveUDPCmd

Description: Sends a command to ADAM-5000/TCP and receives the response by UDP

Syntax: int ADAM5KTCP_SendReceiveUDPCmd(char szIP[], char szSend[], char szReceive[]);

Parameters:

szIP[in]: The IP Address of the ADAM-5000/TCP that send/receive the command/response

szSend[in]: The string in ASCII format that send to the ADAM-5000/TCP

szReceive[out]: The string in ASCII format that response from the ADAM-5000/TCP

6.2.4 Return Codes

Using these function libraries, you can read the error message and the against response from the returning codes.

ADAM5KTCP_NoError	(0)
ADAM5KTCP_StartupFailure	(-1)
ADAM5KTCP_SocketFailure	(-2)
ADAM5KTCP_UdpSocketFailure	(-3)
ADAM5KTCP_SetTimeoutFailure	(-4)
ADAM5KTCP_SendFailure	(-5)
ADAM5KTCP_ReceiveFailure	(-6)
ADAM5KTCP_ExceedMaxFailure	(-7)
ADAM5KTCP_CreateWsaEventFailure	(-8)
ADAM5KTCP_ReadStreamDataFailure	(-9)
ADAM5KTCP_InvalidIP	(-10)
ADAM5KTCP_ThisIPNotConnected	(-11)
ADAM5KTCP_AlarmInfoEmpty	(-12)

6.2.5 Data Structure

```
struct _StreamData
{
    WORD DIO[8]; // DI/DO data for Slot0, Slot1,...., Slot7
    WORD Slot0[8]; // AI/AO data for slot0
    WORD Slot1[8]; // AI/AO data for slot1
    WORD Slot2[8]; // AI/AO data for slot2
    WORD Slot3[8]; // AI/AO data for slot3
    WORD Slot4[8]; // AI/AO data for slot4
    WORD Slot5[8]; // AI/AO data for slot5
    WORD Slot6[8]; // AI/AO data for slot6
    WORD Slot7[8]; // AI/AO data for slot6
}; //StreamData,*pStreamData;

struct _AlarmInfo
{
    BYTE bySlot;              // the Slot of 5000/TCP which cause the alarm change
    BYTE byChannel;           // the Channel of 5000/TCP which cause the alarm change
    BYTE byAlarmType;         // 0: Low Alarm, 1: High Alarm
    BYTE byAlarmStatus;       // 0: Alarm Off, 1: Alarm On
    BYTE byIndexOf5KTCP;      // indicate the index 5000/TCP which cause the alarm change, zero-based
    char szIP[20];            // the IP address which cause the alarm change
    char szDateTime[48];      // e.x 2001/09/23 10:12:34:567 (Year/Month/Day Hour:Minute:Second:mSecond)
};
```

6.3 ADAM-5000/TCP Command

ADAM-5000/TCP system accepts a command/response form with the host computer. 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. This chapter explains the structure of the commands with Modbus/TCP protocol, and guides to use these command sets to implement user's programs.

6.3.1 Command Structure

It is important to understand the encapsulation of a Modbus request or response carried on the Modbus/TCP network. A complete command is consisted of command head and command body. The command head is prefixed by six bytes and responded to pack Modbus format; the command body defines target device and requested action. Following ex- ample will help you to realize this structure quickly.

Example:

If you want to read the value of ADAM-5017 in ADAM-5000/TCP's slot 0(2 channels; address: 40001~40002), the request command should be:

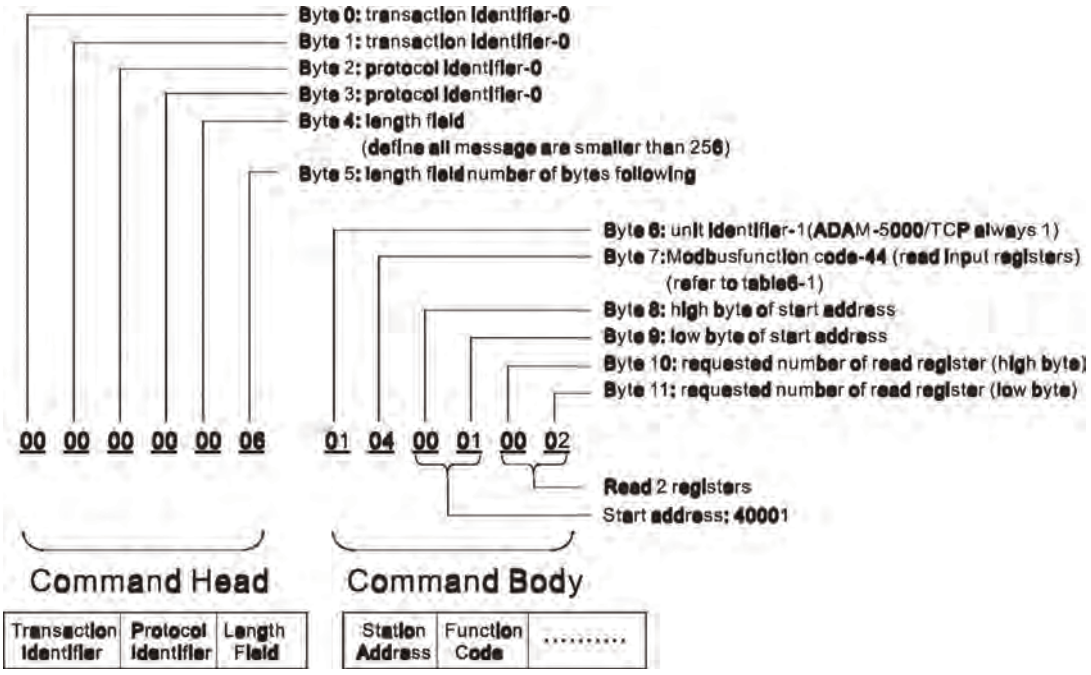


Figure 6.1 Request Comment Structure

And the response should be:

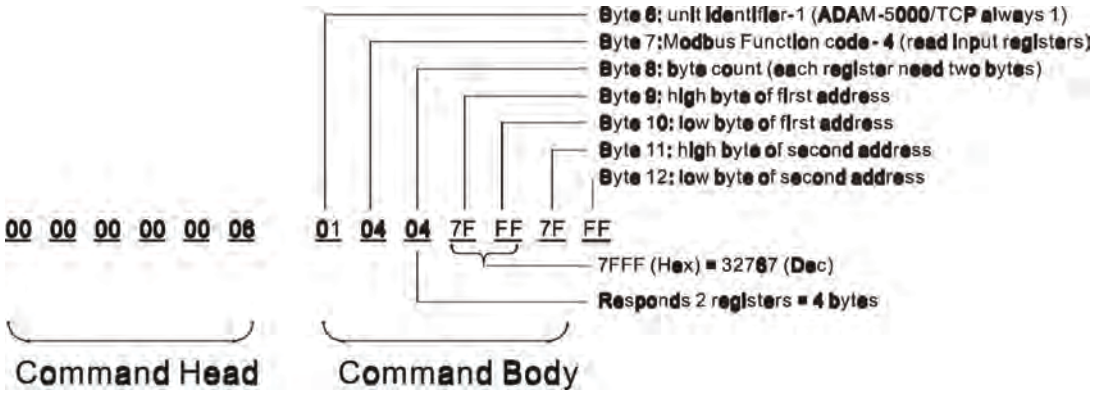


Figure 6.2 Response Comment Structure

6.3.2 Modbus Function Code Introduction

To fulfill your programming requirements, there is a series of function code standards for your reference.

Table 6.1: Response Comment Structure

Code (Hex)	Name	Usage
01	Read Coil Status	Read Discrete Output Bit
02	Read Input Status	Read Discrete Input Bit
03	Read Holding Registers	Read 16-bit register. Used to read integer or floating point process data.
04	Read Input Registers	
05	Force Single Coil	Write data to force coil ON/OFF
06	Preset Single Register	Write data in 16-bit integer format
08	Loopback Diagnosis	Diagnostic testing of the communication port
15	Force Multiple Coils	Write multiple data to force coil ON/OFF
16	Preset Multiple Registers	Write multiple data in 16-bit integer format

Function Code 01

The function code 01 is used to read the discrete output's ON/OFF status of ADAM-5000/TCP in a binary data format.

Request message format for function code 01:

	Command Body				
Station Address	Function Code	Start Address High Byte	Start Address Low Byte	Requested Number of Coil High Byte	Requested Number of Coil Low Byte

Example: Read coil number 1 to 8 (address number 10001 to 10008) from ADAM-5000/TCP
01 01 00 01 00 08

Response message format for function code 01:

	Command Body				
Station Address	Function Code	Byte Count	Data	Data	…

Example: Coils number 2 and 7 are on, all others are off.
01 01 01 42
In the response the status of coils 1 to 8 is shown as the byte value 42 hex, equal to 0100 0010 binary.

Function Code 02

The function code 02 is used to read the discrete input's ON/OFF status of ADAM-5000/TCP in a binary data format.

Request message format for function code 02:

	Command Body				
Station Address	Function Code	Start Address High Byte	Start Address Low Byte	Requested Number of Input High Byte	Requested Number of Input Low Byte

Example: Read coil number 1 to 8 (address number 10001 to 10008) from ADAM-5000/TCP
01 01 00 01 00 08

Response message format for function code 02:

Command Body					
Station Address	Function Code	Byte Count	Data	Data	…

Example: input number 2 and 3 are on, all others are off.
01 01 01 60
In the response the status of input 1 to 8 is shown as the byte value 60 hex, equal to 0110 0000 binary.

Function Code 03/04

The function code 03 or 04 is used to read the binary contents of input registers.

Request message format for function code 03 or 04:

Command Body					
Station Address	Function Code	Start Address High Byte	Start Address Low Byte	Requested Number of Register High Byte	Requested Number of Register Low Byte

Example: Read Analog inputs #1 and #2 in addresses 40001 to 40004 as floating point value from ADAM-5000/TCP
01 04 00 01 00 04

Response message format for function code 03 or 04:

Command Body					
Station Address	Function Code	Byte Count	Data	Data	…

Example: Analog input #1 and #2 as floating point values where AI#1=100.0 and AI#2=55.32
01 04 08 42 C8 00 00 47 AE 42 5D

Function Code 05

Force a single coil to either ON or OFF. The requested ON/OFF state is specified by a constant in the query data field. A value of FF 00 hex requests it to be ON. A value of 00 00 hex requests it to be OFF. And a value of FF FF hex requests it to release the force.

Request message format for function code 05:

Command Body					
Station Address	Function Code	Coil Address High Byte	Coil Address Low Byte	Force Data High Byte	Force Data Low Byte

Example: Force coil 3 (address 00003) ON in ADAM-5000/TCP
01 05 00 03 FF 00

Response message format for function code 05: The normal response is an echo of the query, returned after the coil state has been forced.

Command Body					
Station Address	Function Code	Coil Address High Byte	Coil Address Low Byte	Force Data High Byte	Force Data Low Byte

Function Code 06

Presets integer value into a single register.

Request message format for function code 06:

Command Body					
Station Address	Function Code	Register Address High Byte	Register Address Low Byte	Preset Data High Byte	Preset Data Low Byte

Example: Preset register 40002 to 00 04 hex in ADAM-5000/TCP
01 06 00 02 00 04

Response message format for function code 06: The normal response is an echo of the query, returned after the coil state has been preset.

Command Body					
Station Address	Function Code	Register Address High Byte	Register Address Low Byte	Preset Data High Byte	Preset Data Low Byte

Function Code 08

Echoes received query message. Message can be any length up to half the length of the data buffer minus 8 bytes.

Request message format for function code 08:

Command Body		
Station Address	Function Code	Any data, length limited to approximately half the length of the data buffer

Response message format for function code 08:

Command Body		
Station Address	Function Code	Data bytes received

Example: 01 08 00 02 00 04

Function Code 15 (0F hex)

Forces each coil in a sequence of coils to either ON or OFF.

Request message format for function code 15:

Command Body								
Station Address	Function Code	Address High Byte	Address Low Byte	Requested Number of Coil High Byte	Requested Number of Coil Low Byte	Byte Count	Force Data High Byte	Force Data Low Byte

Example: Request to force a series of 10 coils starting at address 00020 (14 hex) in ADAM-5000/TCP.
01 0F 00 14 00 0A 02 CD 01
The query data contents are two bytes: CD 01 hex, equal to 1100 1101 0000 0001 binary.
The binary bits are mapped to the addresses in the following way.
Bit:1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1
Address (000XX): 27 26 25 24 23 2221 20 ----------2928

Response message format for function code 15:

The normal responses return the station address, function code, start address, and requested number of coil forced.

Command Body					
Station Address	Function Code	Address High Byte	Address Low Byte	Requested Number of Coil High Byte	Requested Number of Coil Low Byte

Example: 01 0F 00 14 00 0A

Function Code 16 (10 hex)

Preset values into a sequence of holding registers.

Request message format for function code 16:

Command Body							
Station Address	Function Code	Start Address High Byte	Start Address Low Byte	Requested Number of Register High Byte	Requested Number of Register Low Byte	Byte Count	Data

Example: Preset constant #1 (address 40009) to 100.0 in ADAM-5000/TCP.
01 10 00 09 00 02 04 42 C8 00 00

Response message format for function code 16:

The normal responses return the station address, function code, start address, and requested number of registers preset.

Station Address	Function Code	Start Address High Byte	Start Address Low Byte	Requested Number of Register High Byte	Requested Number of Register Low Byte

Example: 01 10 00 09 00 02

6.4 Apply with ASCII Command for ADAM-5000/TCP System

For users not familiar with the Modbus protocol, Advantech offers a function library as a protocol translator, integrating ASCII command into Modbus/TCP structure. Therefore, users familiar with ASCII commands can access ADAM-5000/TCP easily. Before explaining the structure of ASCII commands packed with Modbus/TCP format. Let's see how to use an ASCII command and how many commands are available for your program.

Figure 6.3 ASCII Command Structure in ADAM-5000/TCP

6.4.1 Sytax of ASCII

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 (hex-decimal) that specifies the target system. The two characters following the address specified 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 four categories:

- System Command Set
- Analog Input 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 sets for each type of modules are listed along with a description of the effect the command has on the given module.

6.4.2 System Command Set

Table 6.2: CPU Command Set

Command Syntax	Command Name	Description
%aannccff	Configuration	Set the baudrate and checksum status for a specified ADAM-5000 system
$aaM	Read Module Name	Returns the module name from a specified ADAM-5000/TCP system
$aaF	Read Firmware	Returns the firmware version code from a specified ADAM-5000/TCP system Version
$aaT	Read I/O Type	Returns the I/O model number of all slots for a specified ADAM-5000/TCP system

%aannccff

Name Configuration

Description Sets RS-485 network baud rate and checksum status for a specified ADAM-5000/TCP system.

Syntax **%aannccff(cr)**

% is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of the ADAM-5000/TCP 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 invalid.

aa (range 00-FF) represents the 2-character hexadeci-network address of an ADAM-5000/ TCP system.

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

Example command: **%01000A40(cr)**

response: **!01(cr)**

The ADAM-5000/TCP system with address 01h 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.

Table 6.3: Baud rate codes	
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

$aaM

Name Read Module Name

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

Syntax **$aaM(cr)**

$ is a delimiter character.

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

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

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

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

response: **!015000(cr)** The command requests the system at address 01h to send its module name.

The system at address 01h responds with module name 5000/TCP indicating that there is an ADAM-5000/TCP at address 01h.

$aaF

Name Read Firmware Version

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

Syntax **$aaF(cr)**

$ is a delimiter character.

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

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

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

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

response: **!01A1.01(cr)** The command requests the system at address 01h to send its firmware version.

The system responds with firmware version **A1.01**.

$aaT

Name Read I/O Type

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

Syntax **$aaT(cr)**

$ is a delimiter character.

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

aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000/TCP system. bb, cc,

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

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

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

response: **!0118245160(cr)** The command requests the ADAM-5000/TCP system at address 01h to send all existing I/O module numbers. The system at address 01h responds with I/O module numbers 18, 24, 51 and 60 in slots 0-3. This means that the ADAM-5000/TCP system contains an ADAM-5018,ADAM-5024, ADAM-5051 and ADAM-5060 in slots 0 thru 3.

6.4.3 Analog Input Command Set

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 or ADAM-5000/TCP is being used, the “i” in Si can be set at 0 to 7.

Table 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
$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.

$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/TCP system you want to configure.

Si identifies the desired slot i (i:0 to 7). 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 6-4. 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 address of an ADAM-5000/TCP system.

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

Example command: **$01S3A2000(cr)**

response: **!35(cr)** The RTD input module in slot 3 of the ADAM-5000/ TCP system at address 01h 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/TCP system you want to interrogate.

Si identifies the desired slot i (i:0 to 7)

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/TCP 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).

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

Example command: **$01S3B(cr)**

response: **!012000(cr)** The RTD input module inslot 3 of the ADAM-5000/ TCP system at address 01h 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 Modbus network address of the ADAM-5000/TCP system you want to interrogate.

Si is the I/O slot of the ADAM-5000/TCP 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 Modbus network address of the ADAM-5000/TCP 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).

Example command: **$01S3(cr)**

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

The command requests the RTD input module in slot 3 of the ADAM-5000/TCP system at address 01h 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 Modbus network address of the ADAM-5000/TCP system you want to interrogate.

SiCj identifies the desired slot i (i:0 to 7) 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 Modbus network address of an ADAM-5000/TCP 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).

Example command: **$01S3C0(cr)**

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

The command requests the RTD input module in slot 3 of the ADAM-5000/TCP system at address 01h 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 Modbus network address of the ADAM-5000/TCP 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 Modbus network address of an ADAM-5000/TCP 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/TCP 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 invalid.

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

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

Example command: **$01S1501(cr)**

response: **!01(cr)** The command enables/disables the channels of the analog input module in slot 1 of the system at address 01h. 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/TCP 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/TCP 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.

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

Example command: **$01S16(cr)**

response: **!0101(cr)** The command asks the analog input module in slot 1 of the system at address 01h to send the status of its input channels. The analog input module responds that channel 0 of its multiplex channels is enabling, 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/TCP system which contains the RTD module.

Si identifies the slot i (i:0 to 7) 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/TCP 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/TCP system which contains the module which is to be calibrated.

Si identifies the slot i (i:0 to 7) 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/TCP 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 Modbus network address of the ADAM-5000/TCP system which contains the module to be calibrated.

Si identifies the desired slot i (i:0 to 7) 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 Modbus network address of an ADAM-5000/TCP system.

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

Table 6.5: ADAM-5017/5018 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

$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/TCP 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)

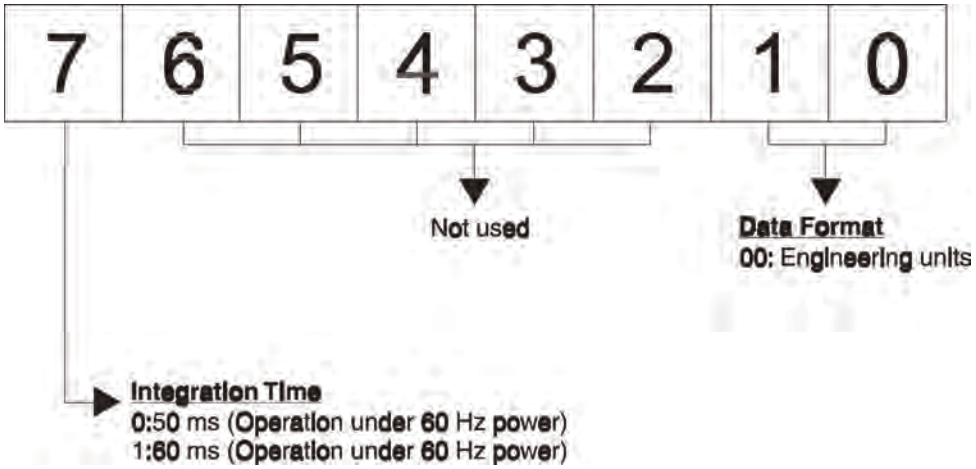


Figure 6.4 Data format for 8-bit parameters

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/TCP system.

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

Example command: **$01S3A0000(cr)**

response: **!01(cr)**

The analog input module in slot 3 of the ADAM-5000/TCP system at address 01h 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/TCP 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/TCP 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).

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

Example

command: **$01S1B**

response: **!010000**

The ADAM-5018 analog input module in slot 1 of the ADAM-5000/TCP system at address 01h 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/TCP 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)

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

Note! *Bit 7 cannot be enabled in the ADAM-5018 since the module is physically limited to 7 channels.*

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/TCP system.

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

Example command: **$01S1581(cr)**

response: **!01(cr)** The command enables/disables channels of the analog input module in slot 1 of the system at address 01h. 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/TCP 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/TCP 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 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: **$01S16(cr)**

response: **!01FF(cr)** The command asks the analog input module in slot 1 of the system at address 01h to send the status of its input channels. The analog input module responds that all its multiplex channels are enabling (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 Modbus network address of the ADAM-5000/TCP system you want to interrogate.

Si is the I/O slot of ADAM-5000/TCP 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)

Example command: **#01S1(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/TCP system at address 01h 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 channel 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 Modbus network address of the ADAM-5000/TCP 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)

Example command: **#01S2C2(cr)**

response: **>+1.4567**

The command requests the analog input module in slot 2 of the ADAM-5000/TCP system at address 01h 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 Modbus network address of the ADAM-5000/TCP 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 Modbus address of an ADAM-5000/TCP system.

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

$aaSi0

Name Span Calibration

Description Calibrates a specified analog input module 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/TCP system which is to be calibrated.

Si identifies the I/O slot which is 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/TCP 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.*

$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/TCP 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/TCP 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.*

$aaSi3

Name CJC Status Command (ADAM-5018 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/TCP 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.

> is a 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/TCP 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)

Example command: **$01S13(cr)**

response: **>+0136.8(cr)** The command requests the analog input module in slot 1 of the ADAM-5000/TCP system at address 01h to read its CJC sensor and return the data. The ana- log input module responds with 36.8°C.

$aaSi9shhhh

Name CJC Zero Calibration (ADAM-5018 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/TCP 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/TCP system.

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

Example command: **$01S29+0042(cr)**

response: **!01(cr)**
The command increases the CJC offset value of the analog input module in slot 2 of the system at address 01h 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.*

Table 6.6: ADAM-5017H /5017UH Analog Input command Set

Command Syntax	Command Name	Description
$aaSiCjArrFF	Set Input Range	Sets input range for a specified channel of an analog input module in a specified system
$aaSiCjB	Read Input Range	Returns the input range for a specified channel of a specified analog input module ina specified system
$aaSiAFFff	Set Data Format	Sets data format in engineering units or two's complement for a specified analog input module ina specified system
$aaSiB	Read Data Format	Returns the data format for a specified analog input module ina specified system
$aaSi5mm	Enable/Disable Channels for Multiplexing	Enables/Disables multiplexing for separate channels of the specified input module
$aaSi6	Read Channels Status	Asks the 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 currently configured data format
#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
$aaSiER	Initialize EEPROM Data	Initializes all EEPROM data in a specified analog input module to their default values.
$aaSi0	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

Note! *The command sets " $aasi5mm, $aasi6, $aasi0, $aasi1" for ADAM-5017H/5017UH are the same with ADAM-5017. Refer the preceding pages to learn the detail.*

$aaSiCjArrFF

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/TCP system you want to configure. SiCj identifies the slot i (i:0 to 7) of the ADAM-5000/ TCP 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)

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

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.*

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/TCP system.

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

Example command: **$01S3C1A0bFF(cr)**

response: **!01(cr)**

Channel 1 of the ADAM-5017H/5017UH module in slot 3 of the ADAM-5000/TCP system at address 01h is set to the input range 0-20 mA, engineering unit data for- mat. 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/TCP system you want to interrogate. SiCj identifies the slot i (i:0 to 7) of the ADAM-5000/ TCP 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/TCP system. **rr** represents the 2-character hexadecimal code of the input range. (See Appendix B)

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

Example command: **$01S3C1B(cr)**

response: **!010b00(cr)**

Channel 1 of the ADAM-5017H/5017UH module in slot 3 of the ADAM-5000/TCP system at address 01h 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/TCP system you want to configure. Si identifies the I/O slot of the ADAM-5000/TCP system containing the ADAM-5017H/5017UH 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.

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

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.*

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/TCP system.

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

Example command: **$01S3AFF00(cr)**

response: **!01(cr)**

The data format of the ADAM-5017H /5017UH module in slot 3 of the ADAM-5000/TCP system at address 01h 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/TCP system you want to interrogate. Si identifies the I/O slot of the ADAM-5000/TCP 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/TCP 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).

Example command: **$01S3B(cr)**

response: **!01FF00(cr)**

The ADAM-5017H /5017UH module in slot 3 of the ADAM-5000/TCP system at address 01h 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 Modbus network address of the ADAM-5000/TCP system you want to interrogate.

Si identifies the I/O slot (i:0 to 7) of ADAM-5000/TCP 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 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: **#01S3(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/TCP system at address 01h 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/TCP 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 Modbus network address of the ADAM-5000/TCP system you want to configure.

Si identifies the I/O slot (i:0 to 7) of ADAM-5000/TCP 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) 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. 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: **#01S3C2(cr)**

response: **+9.750 (cr)**

The command requests the ADAM-5017H/5017UH module in slot 3 of the ADAM-5000/TCP system at address 01h to return the input value of channel 2.

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

Table 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! *his command set applies to the ADAM-5013, ADAM-5017, ADAM-5017H/5017UH and the ADAM-5018.*

$aaSiCjAhs

Name Set Alarm Mode

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

Syntax **$aaSiCjAhs(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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/TCP system.

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

Example command: **$01S0C1AHL(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system.

Syntax **$aaSiCjAh(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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/TCP system.

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

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

Example command: **$01S0C1AL(cr)**

response: **!01M(cr)** Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjAhEs(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) and the desired channel j (j : 0 to 7). **AhE**s 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/TCP system.

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

Example command: **$01S0C1ALEE(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjCh(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus network address of the corresponding ADAM-5000 system.

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

Example command: **$01S0C1CL(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h is instructed to set its Low alarm state to OFF. The system confirms it has done so accordingly.

$aaSiCjAhCSkCn

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/TCP system

Syntax **$aaSiCjAhCSkCn(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 7) 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01S0C1ALCS1C0(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h is instructed to connect its Low alarm to the digital output of point 0 of slot 1 in the same ADAM-5000/TCP 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/TCP system

Syntax **$aaSiCjRhC(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus network address of the corresponding ADAM-5000/TCP system. **SkCn** identifies the desired slot k (k : 0 to 7) 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)

Example command: **$01S0C1RLC(cr)**

response: **!01S1C0(cr)** Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h is instructed to read its Low alarm output connection. The system responds that the Low alarm output con- nects to the digital output at point 0 of slot 1 in the same ADAM-5000/TCP 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/TCP system.

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

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01S0C1AHU+080.00(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjRhU(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

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

Example command: **$01S0C1RHU(cr)**

response: **!01+2.0500(cr)** Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjS(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus of the corresponding ADAM-5000/TCP 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)

Example command: **$01S0C1S(cr)**

response: **!0101(cr)** The command instructs the system at address 01h 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.

Table 6.8: 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/5017UH and the ADAM-5018.*

$aaSiCjAhs

Name Set Alarm Mode

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

Syntax **$aaSiCjAhs(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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/TCP system.

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

Example command: **$01S0C1AHL(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system.

Syntax **$aaSiCjAh(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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/TCP system.

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

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

Example command: **$01S0C1AL(cr)**

response: **!01M(cr)**
Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjAhEs(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) and the desired channel j (j : 0 to 7). **AhE**s 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/TCP system.

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

Example command: **$01S0C1ALEE(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjCh(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus network address of the corresponding ADAM-5000 system.

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

Example command: **$01S0C1CL(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h is instructed to set its Low alarm state to OFF. The system confirms it has done so accordingly.

$aaSiCjAhCSkCn

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/TCP system

Syntax **$aaSiCjAhCSkCn (cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 7) and the desired digital output point n (n : 0 to F). To discon- nect 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01S0C1ALCS1C0(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h is instructed to connect its Low alarm to the digital output of point 0 of slot 1 in the same ADAM-5000/TCP 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/TCP system

Syntax **$aaSiCjRhC(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus network address of the corresponding ADAM-5000/TCP system. **SkCn** identifies the desired slot k (k : 0 to 7) 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)

Example command: **$01S0C1RLC(cr)**

response: **!01S1C0(cr)**
Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP 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/TCP system.

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

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01S0C1AHU+080.00(cr)**

response: **!01(cr)**

Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjRhU(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

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

Example command: **$01S0C1RHU(cr)**

response: **!01+2.0500(cr)**
Channel 1 of slot 0 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjS(cr)**

$ is a delimiter character.

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

SiCj identifies the desired slot i (i : 0 to 7) 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 Modbus of the corresponding ADAM-5000/TCP 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)

Example command: **$01S0C1S(cr)**

response: **!0101(cr)**

The command instructs the system at address 01h 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.

6.4.4 Analog Output Command Set

Table 6.9: 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/TCP system you want to configure. **SiCj** identifies the I/O slot i (i : 0 to 7) 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. 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)

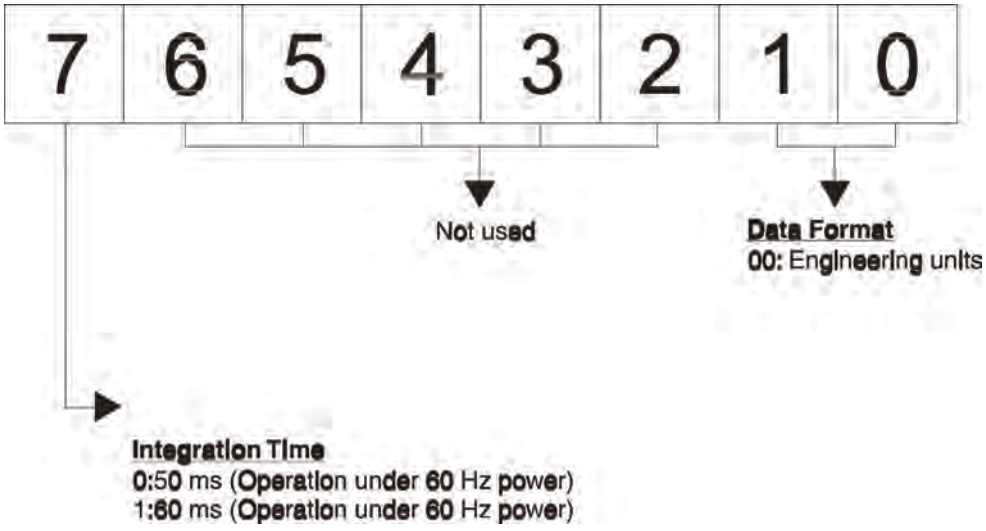


Figure 6.5 The other bits are not used and are set to 0.

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/TCP system.

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

Example

command: **$01S3C0A3110(cr)**

response: **!01(cr)**

The analog output channel 0 in slot 3 of the ADAM-5000/TCP system at address 01h 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/TCP system you want to interrogate. **SiCj** identifies the I/O slot i (i : 0 to 7) 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/TCP 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. 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: **$01S1C1B**

response: **!013210**

The analog output channel 1 in slot 1 of the ADAM-5000/TCP system at address 01h 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/TCP 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 Modbus network address of the ADAM-5000/TCP system. SiCj identifies the I/O slot i (i : 0 to 7) 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.

? delimiter character indicating the command was invalid.

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

Example command: **#01S1C106.000(cr)**

response: **>(cr)**

The command instructs the module in slot 1 of the ADAM-5000/TCP system at address 01h to output a value of 6 mA from it's channel 1. The module should be an analog output module with it's channel 1 con- figured 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 ?01(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/TCP system. **SiCj** identifies the I/O slot i (i : 0 to 7) 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/TCP system.

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

Example command: **$01S1C14(cr)**

response: **!01(cr)**

Presume the present output value of channel 1 of slot 1 in the ADAM-5000/TCP system at address 01h is 9.4 mA. The command asks 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/TCP system at address 01h 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/TCP system. **SiCj** identifies the I/O slot i (i : 0 to 7) 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/TCP 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 be connected to the module's output.*

$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/TCP system. **SiCj** identifies the I/O slot i (i : 0 to 7) 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/TCP 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 be connected to the module's output.*

$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/TCP system. **SiCj** identifies the I/O slot i (i : 0 to 7) 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/TCP system.

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

Example command: **$01S1C2314(cr)**

response: **!01(cr)**
The command tells channel 2 of the analog output module in slot 1 of the ADAM-5000/TCP system at address 01h 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.*

$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 Modbus network address of the ADAM-5000/TCP system. **SiCj** identifies the I/O slot i (i : 0 to 7) and the channel j (j : 0 to 3) for the module you want to return a prior value.

6 is the last value read-back 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 Modbus network address of an ADAM-5000/TCP system.

(data) is the value that is returned by the analog output module. The format of the data depends on the module's configuration data format.

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

Example command: **$01S2C16(cr)**

response: **!0103.000(cr)**
The command tells

channel 1 of the analog output module in slot 2 of the ADAM-5000/TCP system at address 01h 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).

6.4.5 Digital Input/Output Command Set

Table 6.10: Digital Input/Output Command Set

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/TCP system at address aa return the status of its digital input channels and a readback value of its digital output channels.

Syntax **$aaSi6(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of the ADAM-5000/TCP 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-5051/5050/5055)

!aa(dataoutput)(dataoutput)00(cr) if the command is valid. (ADAM-5050/5055/5056)

!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 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 Modbus network address of an ADAM-5000/TCP 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: **$01S26(cr)**

response: **!01112200(cr)**
The command asks

the digital input module in slot 2 of the ADAM-5000/TCP system at address 01h to return the values of all of its channels.

The first 2-character portion of the response indicates the address of the ADAM-5000/TCP system.

The second 2-character 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 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 simultaneously.

Syntax **#aaSiBB(data)(cr)**

is a delimiter character.

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

Si identifies the slot i (i:0 to 7) of the ADAM-5000/TCP 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-5055 can range from 0h to 7h, the ADAM-5056 can range from 0h to Fh, and the ADAM-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) 2 or 4-characters are significant. The digital equivalent of these hexadecimal characters represent the channels' values.

Note that the number of channels on the ADAM-5056 and ADAM-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-5055/5068/5069.

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 Modbus network address of an ADAM-5000/TCP 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/ TCP system at address 15h - either ADAM-5056 or ADAM-5050/5055/5060/5068/5069. Channel 2 of the digital output module is set to ON.

command: **#01S1001234(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/TCP system at address 01h. Channels 2, 4, 5, 9 and 12 will be set to ON, and all other channels are set to OFF.

command: **#01S0003A(cr)**

response: **>(cr)** An output byte with value 3Ah (00111011) is sent to the digital output module (ADAM-5060) in slot 0 of the ADAM-5000/TCP system at address 01h. Chan- nels 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.*

Read Channel Masking Status of ADAM-5050/5051/5052/5056/5060/5068/5069 Command Set

$aaSiM

Name Read Channel Masking Status

Description Asks 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/TCP 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/TCP system that is responding.

(data) is the hexadecimal value representing the status of all digital output channels. 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. And a 2-character value represents the output channels in sequence from 7 thru 0 in ADAM-5068/5069 module. Each bit represents a channel. A value of 1 means the channel is masked, while a value of 0 means the channel is valid.

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

Example command: **$01S1M(cr)**

response: **!011322(cr)**
The command asks the

digital output module in slot 1 of the ADAM-5000/TCP system at address 01h 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/TCP 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)

\

ADAM-5080 Counter/Frequency Command Set

Table 6.11: ADAM-5080 Counter/Frequency Command Set

Command Syntax	Command Name	Description
$aaT	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
$aaSiArrff	Set Configuration	Set slot index and Counter mode
$aaSiB	Read Configuration	The command requests the Configuration of slot
#aaSi	Read All Channel Counter (Frequency) Data	Returns the input value of all channels for the specified input module for a specified system in engineering unit only.
#aaSiCj	Read One Channel Counter (Frequency) Data	The command will return the input value from one of the four channels of a specified module.
$aaSiØ(data)	Set Digital filter Scale	Set the filter seconds to start to measure the input signal.
$aaSiØ	Read Digital filter scale	Read the filter seconds to start to measure the input signal
$aaSiCj5s	Set Counter Start/Stop	Request the addressed counter/frequency module to start or stop the counting.
$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
$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/TCP system.

Syntax **$aaT (cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of the ADAM-5000/TCP 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 Modbus network address of an ADAM-5000/TCP system. **FFFFFFFF** indicates the I/O slot which ADAM-5080 module is in.

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

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

Response: **!01FF80FFFF(cr)**

ADAM-5080 is plugged in slot 1 and the command requests the system at address 01h to send its module name.

$aaF

Name Read Firmware Version

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

Syntax **$aaF(cr)**

$ is a delimiter character.

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

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

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

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

response: **!01A1.1(cr)**
The command requests the system at address 01h to send its firmware version. The system responds with firmware version A1.1.

$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/TCP 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 represents UP/DOWN counter mode.

rr=02 represents Frequency mode. **ff** indicates which format is

ff=00 represents the engineer format.

ff=02 represents the hexadecimal 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/TCP system.

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

Example command: **$01S1A0002(cr)**

response: **!01(cr)**

The ADAM-5080 in Slot 1 of ADAM-5000 system at address 01h is in Bi-direction mode and configured for hexadecimal 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/TCP 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/TCP system.

rr=00 represents Bi-direction counter mode.

rr=01 represents UP/DOWN counter mode.

rr=02 represents Frequency mode. **ff** indicates which format is

ff=00 represents the engineer format.

ff=02 represents the hexadecimal format.

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

Example command: **$01S3B(cr)**

response: **!010100(cr)**

The ADAM-5080 in Slot 3 of ADAM-5000/TPC system at address 01h 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 Modbus network address of the ADAM-5000/TCP 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 hexadecimal format. If (data) = " ", it means the channel is invalid.

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

Example command: **#01S2(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/TCP system at address 01h.

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/TCP system at address 01h to return the input values of all channels. The module response that input values if all channels are hexadecimal:

0e88fa63,c33697b5,2a68d61f,e2ca6915

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/TCP system at address 01h 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 ex- ample of the value above, the actual frequency should be:

actual frequency = 98700/100 = 987

If the response is:

>0000F1000002000000031000000DD400(cr)

The command requests the module in slot 2 of the ADAM-5000/TCP system at address 01h 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 (hexdecimal)=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 Modbus address of the ADAM-5000/TCP 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 exists.

> 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)

Example command: **$01S3C2(cr)**

response: **>0000000451(cr)**
The command requests the ADAM-5080 module in slot 3 of the ADAM-5000/TCP system at address 01h to return the input value of channel 2. The counter module responds that the input value of channel 2 is 451.

$aaSi0(data)

Name Set Digital filter Scale

Description Set the filter seconds to start to measure the input signal.

Syntax **$aaSi0(data)(cr)**

$ is a delimiter character.

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

Si identifies the specified slot.

0 is the command for setting digital filter scale.

(data) represents filter seconds 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/TCP system.

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

Example command: **$01S3000765(cr)**

response: **!01(cr)**

The ADAM-5080 in slot 3 of the ADAM-5000/TCP system at address 01h needs 765m seconds to start to measure the input.

$aaSi0

Name Read Digital filter scale

Description Read the filter seconds to start to measure the input signal.

Syntax **$aaSi0(cr)**

$ is a delimiter character.

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

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

0 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/TCP system.

(data) represents filter seconds from 8 μs~65000 μs. Be aware that (data) has 5 characters.

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

Example command: **$01S30(cr)**

response: **!0100765(cr)**

The command requests the ADAM-5080 in slot 3 of the ADAM-5000/TCP system at address 01h to read the filter seconds. The module responds with 765m 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 Modbus network address of the ADAM-5000/TCP 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 Modbus network address of an ADAM-5000/TCP system.

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

Example command: **$01S3C251(cr)**

response: **!01(cr)**
The command requests channel 2 of ADAM-5080 in slot 3 in ADAM-5000/TCP system at address 01h 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 Modbus network address of the ADAM-5000/TCP 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 Modbus network address of an ADAM-5000/TCP system.

s represents start/stop command.

s=0 indicate stop counter. s=1 indicate start counter.

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

Example command: **$01S3C25(cr)**

response: **!011(cr)**

The channel 2 of ADAM-5080 in slot 3 in ADAM-5000/TCP system at address 01h 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 Modbus network address of the ADAM-5000/TCP 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 Modbus network address of an ADAM-5000/TCP system.

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

Example command: **$01S3C26(cr)**

response: **!01(cr)**

The command requests the channel 2 of ADAM-5080 in slot 3 in ADAM-5000/TCP system at address 01h to clear counter value.

$aaSi7

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 Modbus network address of the ADAM-5000/TCP system.

Si identifies the I/O slot i (i : 0 to 7).

7 is the command for the last value read-back.

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 Modbus network address of an ADAM-5000/TCP 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.*

Example command: **$01S37(cr)**

response: **!0100000001(cr)**
The command requests the ADAM-5080 of slot 3 in ADAM-5000/TCP system at address 01h 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/TCP 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/TCP system.

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

Example command: **@01S3C2P0000004369(cr)**

response: **!01(cr)**

The channel 2 of ADAM-5080 in slot 3 in ADAM-5000/TCP system at address 01h is instructed to set initial counter value. The initial counter value is 4369.

@aaSiCjG

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/TCP 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/TCP system.

(data) is initial value from 0 to 4294967295.Be aware that (data) has 10 characters.

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

Example command: **@01S3C2G(cr)**

response: **!010000004369(cr)**

The channel 2 of ADAM-5080 in slot 3 in ADAM-5000/TCP system at address 01h 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 Modbus network address of an ADAM-5000/TCP system. **SiCj** identifies the desired slot i and the desired channel j. **AhE**s 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01S0C1ALED(cr)**

response: **!01(cr)**

Channel 1 of slot 0 of ADAM-5080 in ADAM-5000/TCP system at address 01h is instructed to disable its Low alarm function. The module confirms that its Low alarm function has been disabled.

$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 Modbus network address of an ADAM-5000/TCP 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 Modbus address of the corresponding ADAM-5000/TCP system.

p indicates alarm mode.

p=D, if alarm is Disable.

P=L, if alarm is Latch.

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

Example command: **$01S0C1AL(cr)**

response: **!01L(cr)**

Channel 1 of slot 0 of ADAM-5080 in ADAM-5000/TCP system at address 01h is instructed to return its Low alarm mode. The system responds thatit 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 Modbus network address of an ADAM-5000/ TCP 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 Modbus network address of the corresponding ADAM-5000/ TCP system.
(cr) represents terminating character, carriage return (0Dh)

Example command: $01S0C1CL(cr)
response: !01(cr)
Channel 1 of slot 0 of ADAM-5080 in ADAM-5000 system at address 01h is instructed to set its Low alarm state to normal. The system confirms it has done so accordingly.

$aaSiCjAhCSkCn

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/TCP system

Syntax **$aaSiCjAhCSkCn(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01S0C1ALCS1C0(cr)**

response: **!01(cr)**

Channel 1 of slot 0 of ADAM-5080 in ADAM-5000/TCP system at address 01h is instructed to connect its Low alarm to the digital output of point 0 of slot 1 in the same ADAM-5000/TCP 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/TCP system

Syntax **$aaSiCjRhC(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP 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)

Example command: **$01S0C1RLC(cr)**

response: **!01SØC1(cr)**
Channel 1 of slot 0 of ADAM-5080 in ADAM-5000/ TCP system at address 01h 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/ TCP 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/TCP system.

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

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01SØC1AHU0000000020(cr)**

response: **!01(cr)**

The channel 1 of slot 0 of ADAM-5080 in ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjRhU(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP 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)

Example command: **$01SØC1RHU(cr)**

response: **!010000000026(cr)**

The channel 1 of slot 0 of ADAM-5080 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjS(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP 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)

Example command: **$01SØC1S**

response: **!0111(cr)**

The channel 1 of slot 0 of ADAM-5080 in the ADAM-5000/TCP system at address 01h is configured to read alarm status. The High alarm has occurred and low alarm has oc-

6.4.6 WatchDog Timer Command Set

Table 6.12: WatchDog Timer Command Set

Command Syntax	Command Name	Description
&AAXdddd(cr) Set WDT timeout value	**Success: !AA(cr)** **Fail: ?AA(cr)**	**dddd** is the SDT 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 SDT 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 **$AAXEWmm(cr)**
&AAXSiDmmmm(cr) Set WDT timeout channel enable mask	**Success: !AA(cr)** **Fail: ?AA(cr)**	**mmmm** indicates a 4-character hexadecimal value representing the SDT timeout channel enable mask of the DIO module.
&AAXSi(cr) Get WDT timeout channel enable mask	**Success: !AAmmmm(cr)** **Fail: ?AA(cr)**	The same as **$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

ADAM-5081 Counter/Frequency Command Set

Table 6.13: ADAM-5081 Counter/Frequency Command Set

Command Syntax	Command Name	Description
$aaT	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
$aaSiArrff	Set Configuration	Set slot index and Counter mode
$aaSiB	Read Configuration	The command requests the Configuration of slot
#aaSi	Read All Channel Counter (Frequency) Data	Returns the input value of all channels for the specified input module for a specified system in engineering unit only.
#aaSiCj	Read One Channel Counter (Frequency) Data	The command will return the input value from one of the four channels of a specified module.
$aaSiØ(data)	Set Digital filter Scale	Set the filter seconds to start to measure the input signal.
$aaSiØ	Read Digital filter scale	Read the filter seconds to start to measure the input signal
$aaSiCj5s	Set Counter Start/Stop	Request the addressed counter/frequency module to start or stop the counting.
$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
$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/TCP system.

Syntax **$aaT (cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of the ADAM-5000/TCP 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 Modbus network address of an ADAM-5000/TCP system. **FFFFFFFF** indicates the I/O slot which ADAM-5081 module is in.

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

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

Response: **!01FF80FFFF(cr)**

ADAM-5081 is plugged in slot 1 and the command requests the system at address 01h to send its module name.

$aaF

Name Read Firmware Version

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

Syntax **$aaF(cr)**

$ is a delimiter character.

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

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

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

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

response: **!01A1.1(cr)**
The command requests the system at address 01h to send its firmware version. The system responds with firmware version A1.1.

$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/TCP 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 represents UP/DOWN counter mode.

rr=02 represents Frequency mode. **ff** indicates which format is

ff=00 represents the engineer format.

ff=02 represents the hexadecimal 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/TCP system.

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

Example command: **$01S1A0002(cr)**

response: **!01(cr)**

The ADAM-5081 in Slot 1 of ADAM-5000 system at address 01h is in Bi-direction mode and configured for hexadecimal 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/TCP 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/TCP system.

rr=00 represents Bi-direction counter mode.

rr=01 represents UP/DOWN counter mode.

rr=02 represents Frequency mode. **ff** indicates which format is

ff=00 represents the engineer format.

ff=02 represents the hexadecimal format.

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

Example command: **$01S3B(cr)**

response: **!010100(cr)**

The ADAM-5081 in Slot 3 of ADAM-5000/TPC system at address 01h 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 Modbus network address of the ADAM-5000/TCP 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 hexadecimal format. If (data) = " ", it means the channel is invalid.

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

Example command: **#01S2(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/TCP system at address 01h.

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/TCP system at address 01h to return the input values of all channels. The module response that input values if all channels are hexadecimal:

0e88fa63,c33697b5,2a68d61f,e2ca6915

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/TCP system at address 01h 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 ex- ample of the value above, the actual frequency should be:

actual frequency = 98700/100 = 987

If the response is:

>0000F1000002000000031000000DD400(cr)

The command requests the module in slot 2 of the ADAM-5000/TCP system at address 01h 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 (hexdecimal)=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 Modbus address of the ADAM-5000/TCP 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 exists.

> 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)

Example command: **$01S3C2(cr)**

response: **>0000000451(cr)**
The command requests the ADAM-5081 module in slot 3 of the ADAM-5000/TCP system at address 01h to return the input value of channel 2. The counter module responds that the input value of channel 2 is 451.

$aaSi0(data)

Name Set Digital filter Scale

Description Set the filter seconds to start to measure the input signal.

Syntax **$aaSi0(data)(cr)**

$ is a delimiter character.

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

Si identifies the specified slot.

0 is the command for setting digital filter scale.

(data) represents filter seconds 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/TCP system.

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

Example command: **$01S3000765(cr)**

response: **!01(cr)**

The ADAM-5081 in slot 3 of the ADAM-5000/TCP system at address 01h needs 765m seconds to start to measure the input.

$aaSi0

Name Read Digital filter scale

Description Read the filter seconds to start to measure the input signal.

Syntax **$aaSi0(cr)**

$ is a delimiter character.

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

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

0 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/TCP system.

(data) represents filter seconds from 8 μs~65000 μs. Be aware that (data) has 5 characters.

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

Example command: **$01S30(cr)**

response: **!0100765(cr)**

The command requests the ADAM-5081 in slot 3 of the ADAM-5000/TCP system at address 01h to read the filter seconds. The module responds with 765m 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 Modbus network address of the ADAM-5000/TCP 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 Modbus network address of an ADAM-5000/TCP system.

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

Example command: **$01S3C251(cr)**

response: **!01(cr)**
The command requests channel 2 of ADAM-5081 in slot 3 in ADAM-5000/TCP system at address 01h 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 Modbus network address of the ADAM-5000/TCP 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 Modbus network address of an ADAM-5000/TCP system.

s represents start/stop command.

s=0 indicate stop counter. s=1 indicate start counter.

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

Example command: **$01S3C25(cr)**

response: **!011(cr)**

The channel 2 of ADAM-5081 in slot 3 in ADAM-5000/TCP system at address 01h 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 Modbus network address of the ADAM-5000/TCP 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 Modbus network address of an ADAM-5000/TCP system.

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

Example command: **$01S3C26(cr)**

response: **!01(cr)**

The command requests the channel 2 of ADAM-5081 in slot 3 in ADAM-5000/TCP system at address 01h to clear counter value.

$aaSi7

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 Modbus network address of the ADAM-5000/TCP system.

Si identifies the I/O slot i (i : 0 to 7).

7 is the command for the last value read-back.

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 Modbus network address of an ADAM-5000/TCP 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.*

Example command: **$01S37(cr)**

response: **!0100000001(cr)**
The command requests the ADAM-5081 of slot 3 in ADAM-5000/TCP system at address 01h 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/TCP 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/TCP system.

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

Example command: **@01S3C2P0000004369(cr)**

response: **!01(cr)**

The channel 2 of ADAM-5081 in slot 3 in ADAM-5000/TCP system at address 01h is instructed to set initial counter value. The initial counter value is 4369.

@aaSiCjG

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/TCP 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/TCP system.

(data) is initial value from 0 to 4294967295.Be aware that (data) has 10 characters.

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

Example command: **@01S3C2G(cr)**

response: **!010000004369(cr)**

The channel 2 of ADAM-5081 in slot 3 in ADAM-5000/TCP system at address 01h 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 Modbus network address of an ADAM-5000/TCP system. **SiCj** identifies the desired slot i and the desired channel j. **AhE**s 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01S0C1ALED(cr)**

response: **!01(cr)**

Channel 1 of slot 0 of ADAM-5081 in ADAM-5000/TCP system at address 01h is instructed to disable its Low alarm function. The module confirms that its Low alarm function has been disabled.

$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 Modbus network address of an ADAM-5000/TCP 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 Modbus address of the corresponding ADAM-5000/TCP system.

p indicates alarm mode.

p=D, if alarm is Disable.

P=L, if alarm is Latch.

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

Example command: **$01S0C1AL(cr)**

response: **!01L(cr)**

Channel 1 of slot 0 of ADAM-5081 in ADAM-5000/TCP system at address 01h is instructed to return its Low alarm mode. The system responds thatit 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 Modbus network address of an ADAM-5000/ TCP 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 Modbus network address of the corresponding ADAM-5000/ TCP system.
(cr) represents terminating character, carriage return (0Dh)

Example command: $01S0C1CL(cr)
response: !01(cr)
Channel 1 of slot 0 of ADAM-5081 in ADAM-5000 system at address 01h is instructed to set its Low alarm state to normal. The system confirms it has done so accordingly.

$aaSiCjAhCSkCn

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/TCP system

Syntax **$aaSiCjAhCSkCn(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01S0C1ALCS1C0(cr)**

response: **!01(cr)**

Channel 1 of slot 0 of ADAM-5081 in ADAM-5000/TCP system at address 01h is instructed to connect its Low alarm to the digital output of point 0 of slot 1 in the same ADAM-5000/TCP 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/TCP system

Syntax **$aaSiCjRhC(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP 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)

Example command: **$01S0C1RLC(cr)**

response: **!01SØC1(cr)**
Channel 1 of slot 0 of ADAM-5081 in ADAM-5000/ TCP system at address 01h 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/ TCP 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/TCP system.

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

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP system.

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

Example command: **$01SØC1AHU0000000020(cr)**

response: **!01(cr)**

The channel 1 of slot 0 of ADAM-5081 in ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjRhU(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP 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)

Example command: **$01SØC1RHU(cr)**

response: **!010000000026(cr)**

The channel 1 of slot 0 of ADAM-5081 in the ADAM-5000/TCP system at address 01h 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/TCP system

Syntax **$aaSiCjS(cr)**

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal Modbus network address of an ADAM-5000/TCP 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 Modbus network address of the corresponding ADAM-5000/TCP 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)

Example command: **$01SØC1S**

response: **!0111(cr)**

The channel 1 of slot 0 of ADAM-5081 in the ADAM-5000/TCP system at address 01h is configured to read alarm status. The High alarm has occurred and low alarm has oc-

Appendix A

Design Worksheets

An organized system configuration will lead to efficient performance and reduce engineer effort. This Appendix provides the necessary worksheet, helping users to configure their DA&C system in order. Follow these working steps to build up your system relational document:

Step 1: Asking questions and getting answers for your control strategy.

1. What will be monitored and controlled? (List the equipment)
2. What will be monitored and controlled separately? (Divide the function area)
3. What will be monitored and controlled by ADAM-5000/TCP? (List the target equipment in different function areas)

Step 2: Identify the I/O types of each equipment and full-fill Table A-1 to establish the I/O data base

Table A.1: I/O Data Base

Function Area	Equipment	Input or Output	I/O Module Type	I/O Module Product No.	Voltage of Range	Current of Special	Range Requirements

Table A.1: I/O Data Base							

Step 3: Mapping the I/O data base into ADAM-5000/TCP system.

1. In column A, note the ADAM-5000/TCP IP addresses mapped for individual function areas.
2. In column B, list the I/O module's product number.
3. In column C, enter the maximum number of I/O points available per module.
4. In column D, total the number of the I/O point you need.
5. In column E, calculate the total number of these modules that you will need for these ADAM-5000/TCP systems.
6. In column F, enter the number of spare modules that you may need for future expansion in these ADAM-5000/TCP systems.
7. In column G, enter the total number (Required + Spare) of these modules that you need for these ADAM-5000/TCP systems.

Table A.2: Summary Required Modules						
<A>	<B>	<C>	<D>	<E>	<F>	<G>
ADAM-5000/TCP IP Address	I/O Module Product No.	I/O Points per Module	Total I/O Points Required	I/O Module Required	Spare I/O Modules	Total I/O Modules

Table A.2: Summary Required Modules						

Step 4: Implement the Modbus address in to the I/O table.

Table A.3: Table for Programming						
ADAM-5000/TCP IP Address	I/O Module Product No.	I/O Points per Module	Total I/O Points Required	I/O Module Required	Spare I/O Modules	Total I/O Modules

Table A.3: Table for Programming						

These several worksheets are very useful to hardware wiring and software integration, please make copies to establish your own system configuration documentation.

Appendix B

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	0.01º C (two decimal places)
Type K, E, R, S, and B thermocouple	0.1º C (one decimal places)

Example 1

The input value is -2.65 V 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-5017

Module	Range Code	Input Range Description	Data Formats	+F.S.	Zero	-F.S.	Displayed Resolution	Actual Value
ADAM-5017	08h	±10 V	Engineering Units	+10.000	±00.000	-10.000	1 mV	Reading/1000
			% of FSR	+100.00	±000.00	-10.000	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	09h	±5 V	Engineering Units	+5.0000	±0.0000	-5.0000	100.00 µV	Reading/1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	0Ah	±1 V	Engineering Units	+10.000	±0.0000	-1.0000	100.00 µV	Reading/10000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	0Bh	±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	
	0Ch	±150 mV	Engineering Units	+150.0	±000.00	-150.00	10 µV	Reading/100
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	0Dh	±20 V	Engineering Units	+20.000	±00.000	-20.000	1 µV	Reading/1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	

B.3 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 mV	Engineering Units	+20.000	±00.000	-20.000	1 µV	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	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	

Appendix C

Grounding Reference

Field Grounding and Shielding Application Overview

Unfortunately, it's impossible to finish a system integration task at one timC. We always meet some trouble in the field. A communication net- work or system isn't stable, induced noise or equipment is damaged or there are storms. However, the most usual issue is just simply improper wiring, ie, grounding and shielding. You know the 80/20 rule in our life: we spend 20% time for 80% work, but 80% time for the last 20% of the work. So is it with system integration:

we pay 20% for Wire / Cable and 0% for Equipment. However, 80% of reliability depends on Grounding and Shielding. In other words, we need to invest more in that 20% and work on these two issues to make a highly reliable system. This application note brings you some concepts about field grounding and shielding. These topics will be illustrated in the following pages.

C.1 Grounding

C.1.1 The 'Earth' for reference

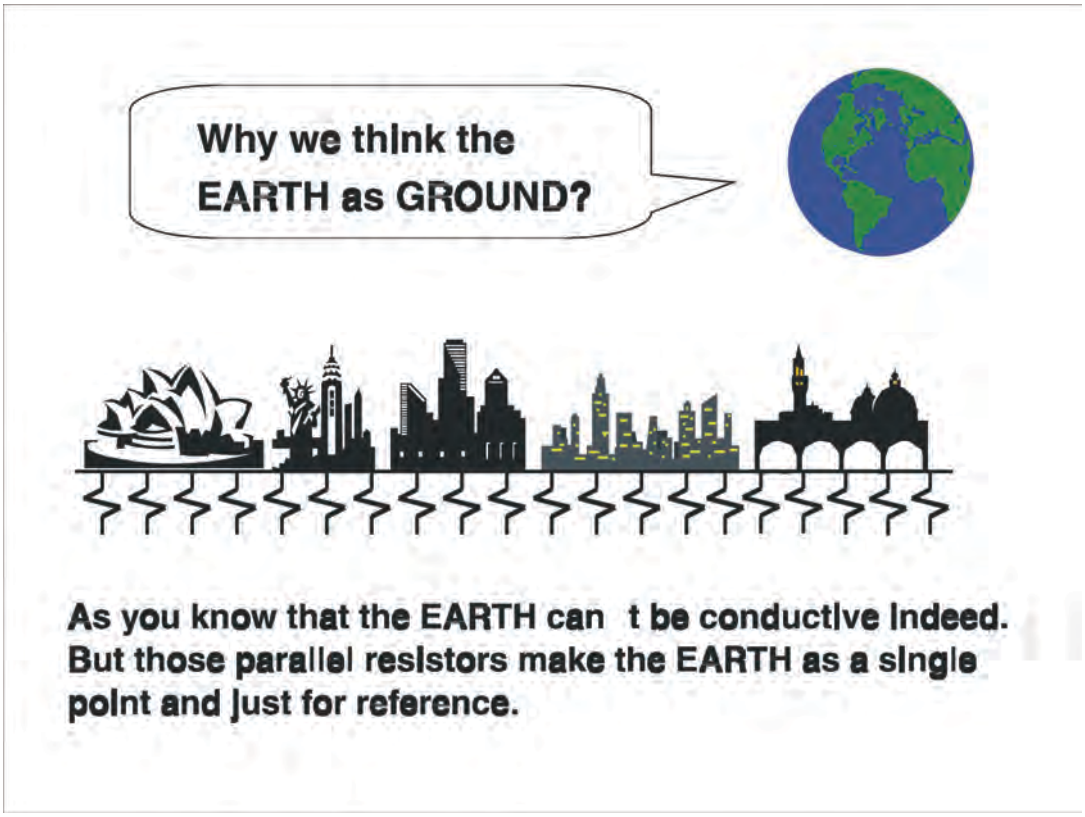


Figure C.1 Think the EARTH as GROUND.

As you know, the EARTH cannot be conductivC. However, all buildings lie on, or in, the EARTH. Steel, concrete and associated cables (such as lighting arresters) and power system were connected to EARTH. Think of them as resistors. All of those infinite parallel resistors make the EARTH as a single reference point.

C.1.2 The 'Frame Ground' and 'Grounding Bar'

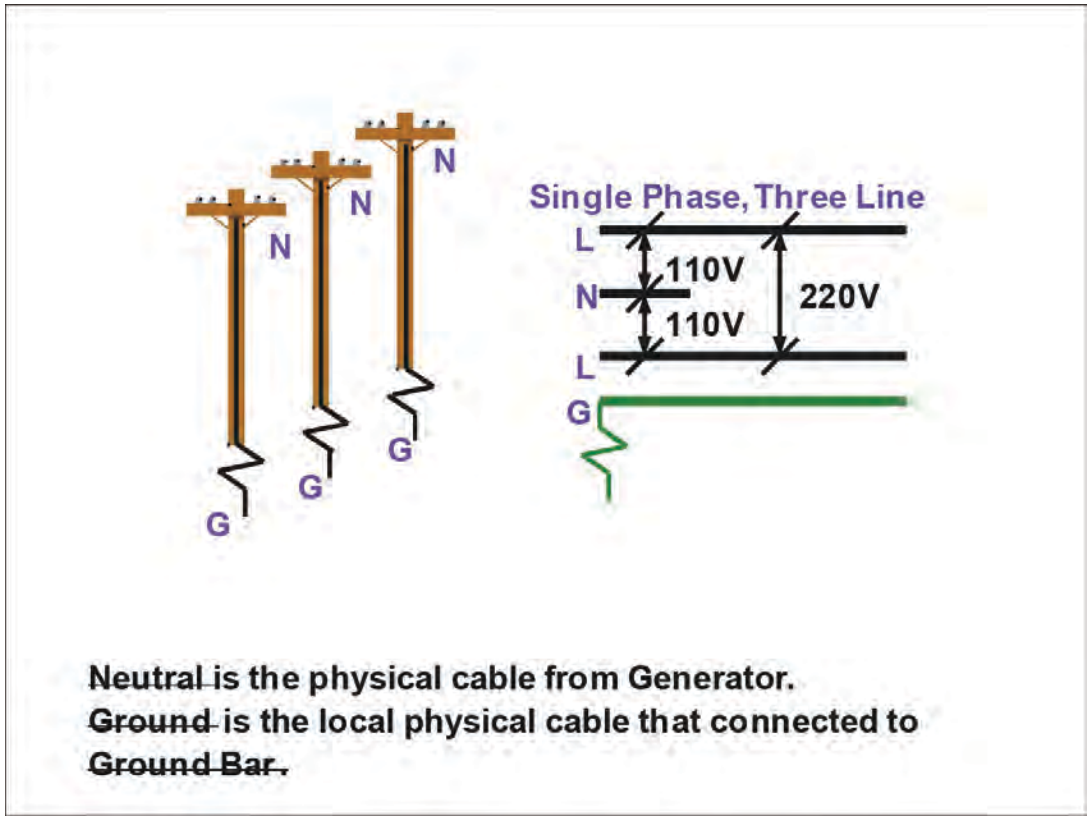


Figure C.2 Grounding Bar

Grounding is one of the most important issues for our system. Just like Frame Ground of the computer, this signal offers a reference point of the electronic circuit inside the computer. If we want to communicate with this computer, both Signal Ground and Frame Ground should be connected to make a reference point of each other's electronic circuit. Generally speaking, it is necessary to install an individual grounding bar for each system, such as computer networks, power systems, tele-communication networks, etc. Those individual grounding bars not only provide the individual reference point, but also make the earth a our ground!

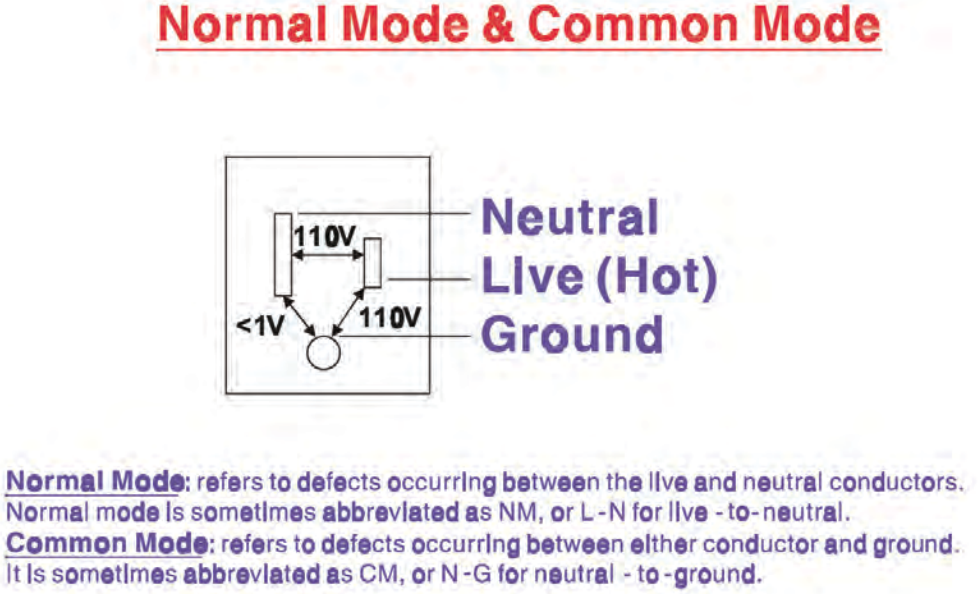


Figure C.3 Normal mode and Common modC

C.1.3 Normal Mode and Common Mode

Have you ever tried to measure the voltage between a live circuit and a concrete floor? How about the voltage between neutral and a concrete floor? You will get non-sense values. 'Hot' and 'Neutral' are just relational signals: you will get 110VAC or 220VAC by measuring these signals. Normal mode and common mode just show you that the Frame Ground is the most important reference signal for all the systems and equipment.

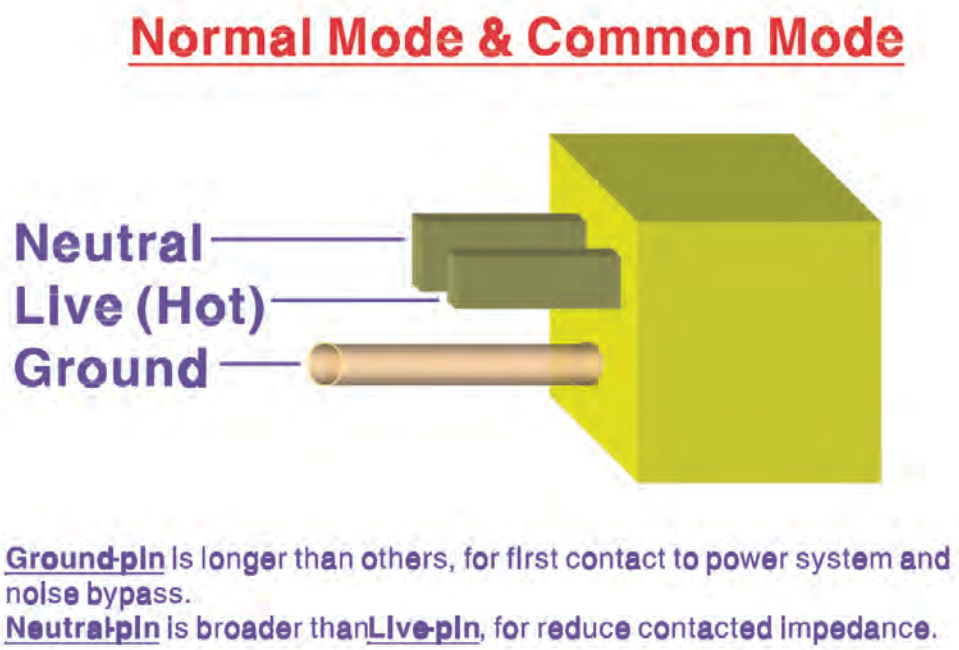


Figure C.4 Normal mode and Common modC

- Ground-pin is longer than others, for first contact to power system and noise bypass.
- Neutral-pin is broader than LivC-pin, for reducing contact impedancC.

C.1.4 Wire impedance

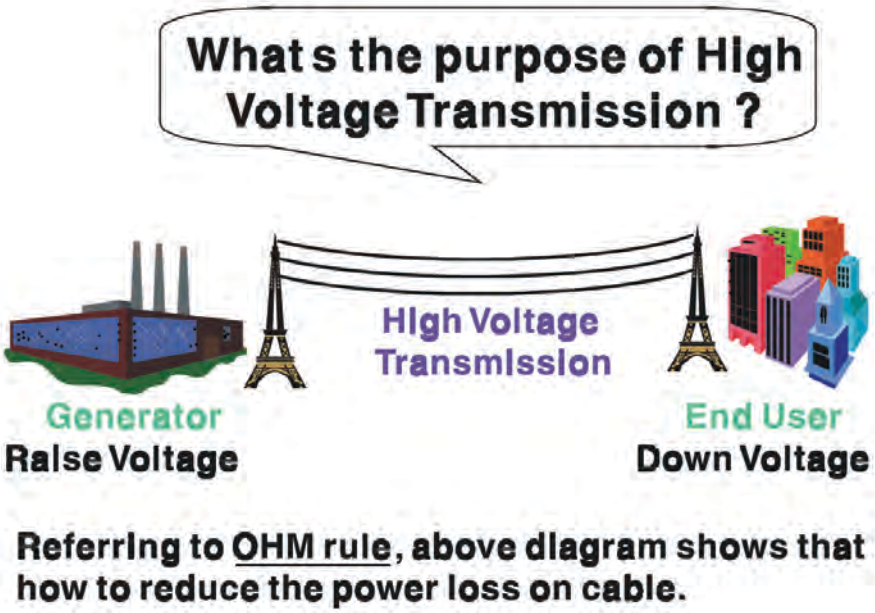


Figure C.5 The purpose of high voltage transmission

- What's the purpose of high voltage transmission?
  We have all seen high voltage transmission towers. The power plant raises the voltage while generating the power, then a local power station steps down the voltagC. What is the purpose of high voltage transmission wires ? According to the energy formula, P = V * I, the current is reduced when the voltage is raised. As you know, each cable has impedance because of the metal it is made of. Referring to Ohm's Law, (V = I * R) this decreased current means lower power losses in the wirC. So, high voltage lines are for reducing the cost of moving electrical power from one place to another.

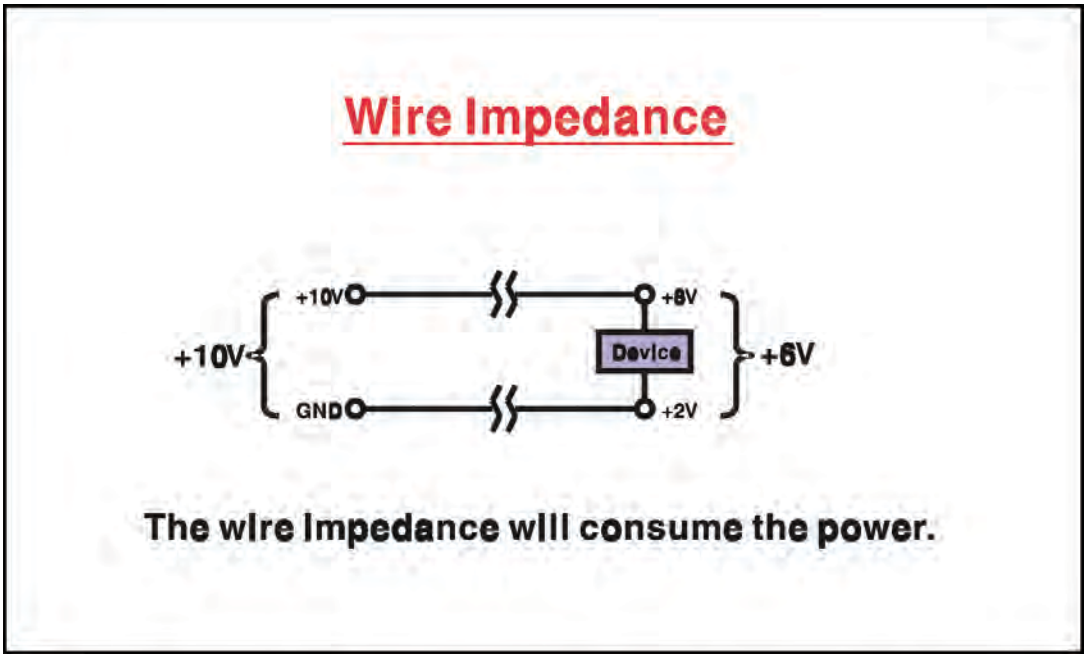


Figure C.6 Wire impedancC

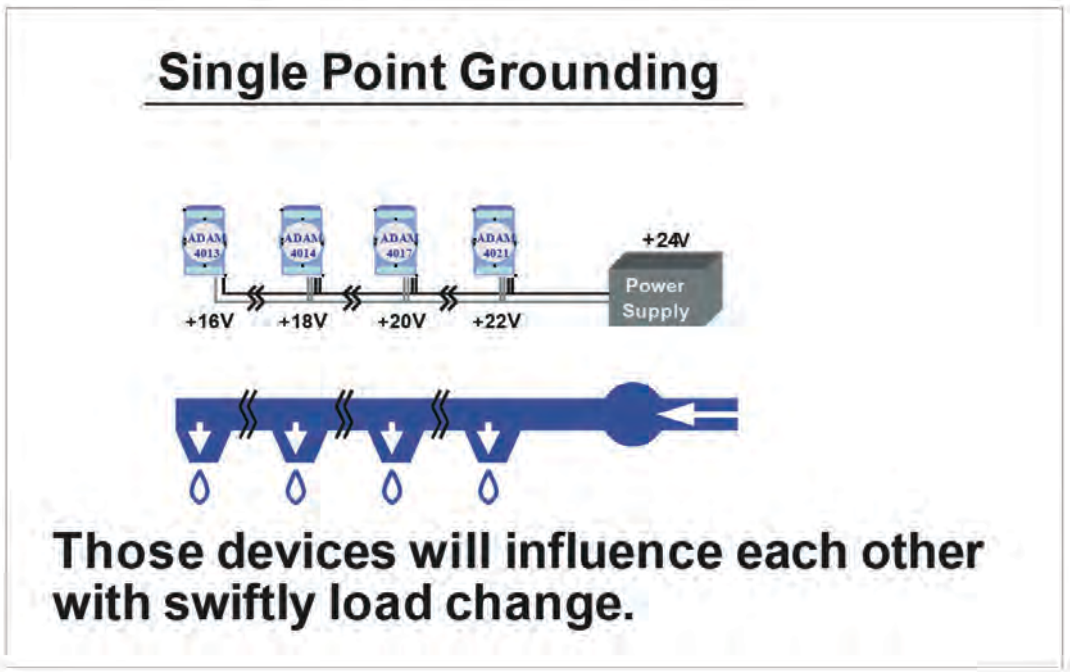


Figure C.7 Single point grounding (1)

- What's Single Point Grounding?
  Maybe you have had an unpleasant experience while taking a hot shower in Winter. Someone turns on a hot water faucet somewhere else. You will be impressed with the cold water! The bottom diagram above shows an example of how devices will influence each other with swift load change. For example, normally we turn on all the four hydrants for testing. When you close the hydrant 3 and hydrant 4, the other two hydrants will get more flow. In other words, the hydrant cannot keep a constant flow rate.

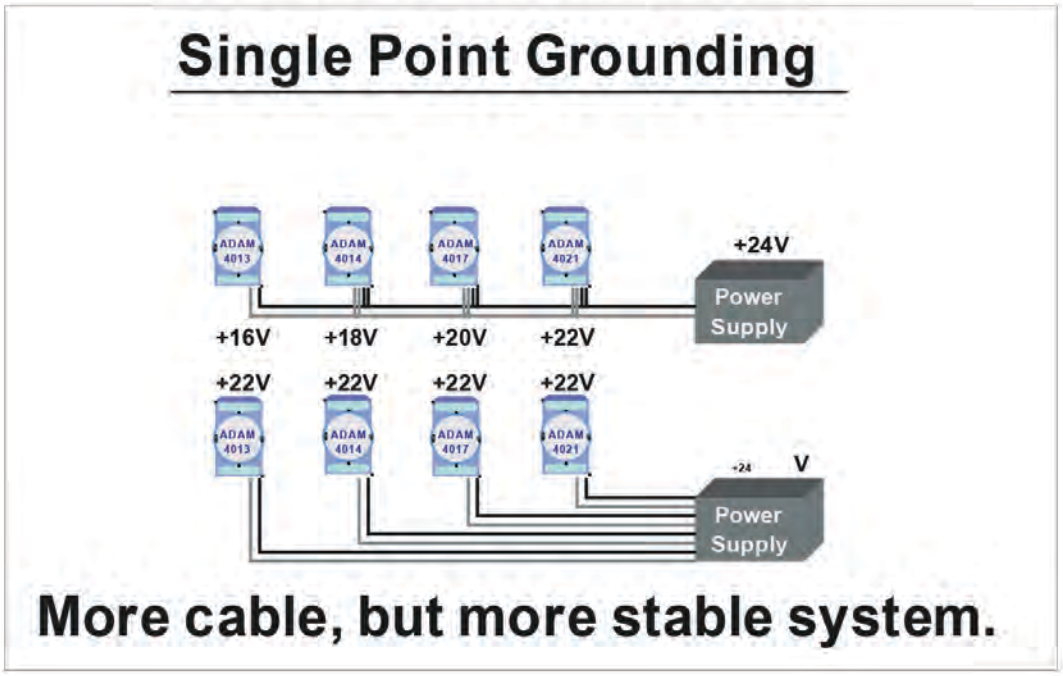


Figure C.8 Single point grounding (2)

The above diagram shows you that a single point grounding system will be a more stable system. If you use thin cable for powering these devices, the end device will actually get lower power. The thin cable will consume the energy.

C.2 Shielding

C.2.1 Cable Shield

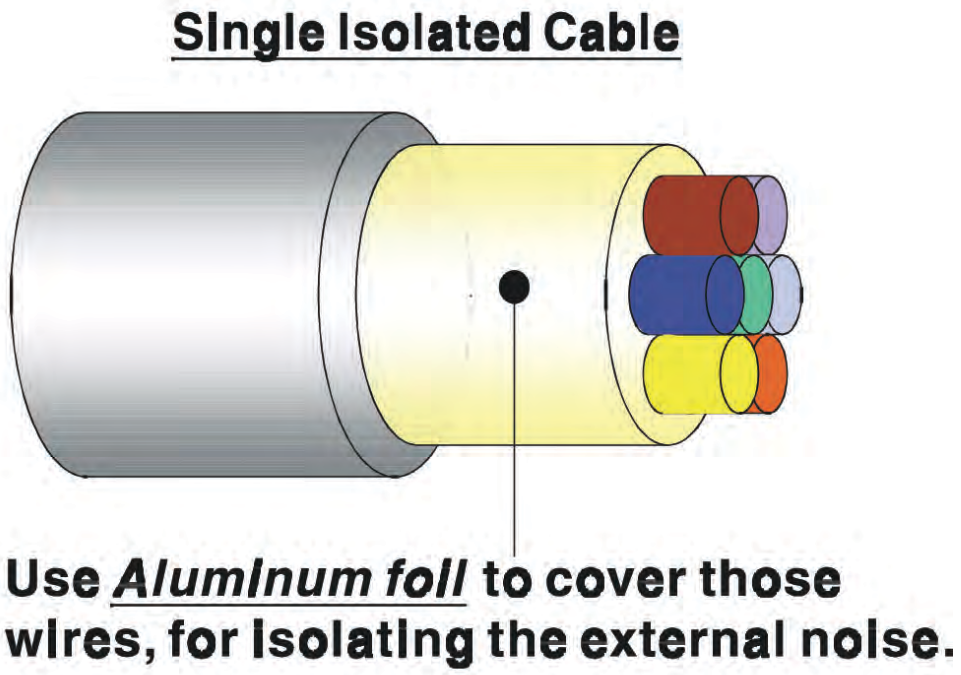


Figure C.9 Single isolated cable

- Single isolated cable The diagram shows the structure of an isolated cablC. You see the isolated layer which is spiraled Aluminum foil to cover the wires. This spiraled structure makes a layer for shielding the cables from external noisC.

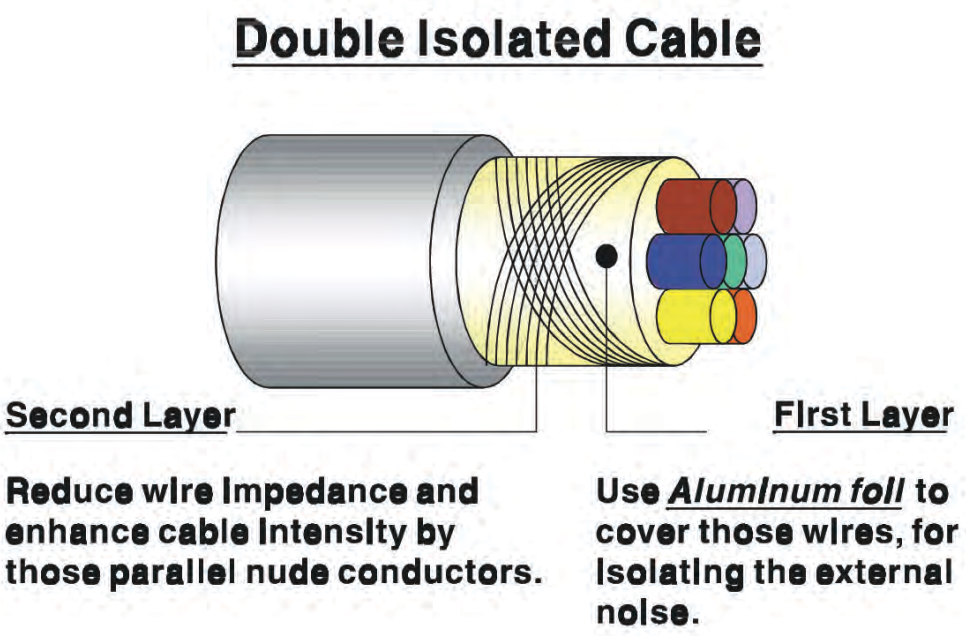


Figure C.10 Double isolated cable

- Double isolated cable Figure 10 is an example of a double isolated cablC. The first isolating layer of spiraled aluminum foil covers the conductors. The second isolation layer is several bare conductors that spiral and cross over the first shield layer. This spiraled structure makes an isolated layer for reducing external noisC.

Additionally, follow these tips just for your referencC.

- The shield of a cable cannot be used for signal ground. The shield is designed for carrying noise, so the environment noise will couple and interfere with your system when you use the shield as signal ground.
- The higher the density of the shield - the better, especially for communication network.
- Use double isolated cable for communication network / AI / AO.
- Both sides of shields should be connected to their frame while inside the devicC. (for EMI consideration)
- Don’t strip off too long of plastic cover for soldering.

C.2.2 System Shielding

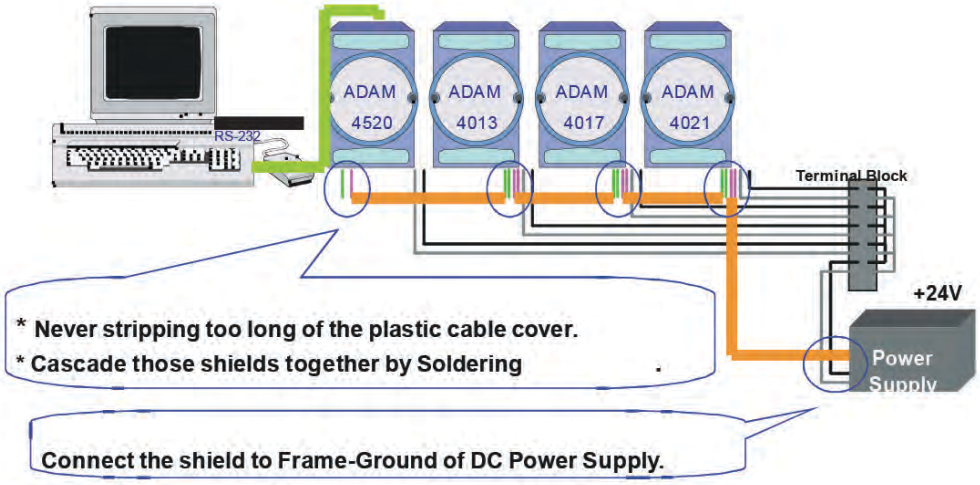


Figure C.11 System Shielding

- Never strip too much of the plastic cable cover. This is improper and can destroy the characteristics of the shielded-twisted-pair cable. Besides, the bare wire shield easily conducts the noise.
- Cascade these shields together by soldering. Refer to following pages for further detailed explanation.
- Connect the shield to Frame Ground of DC power supply to force the conducted noise to 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)

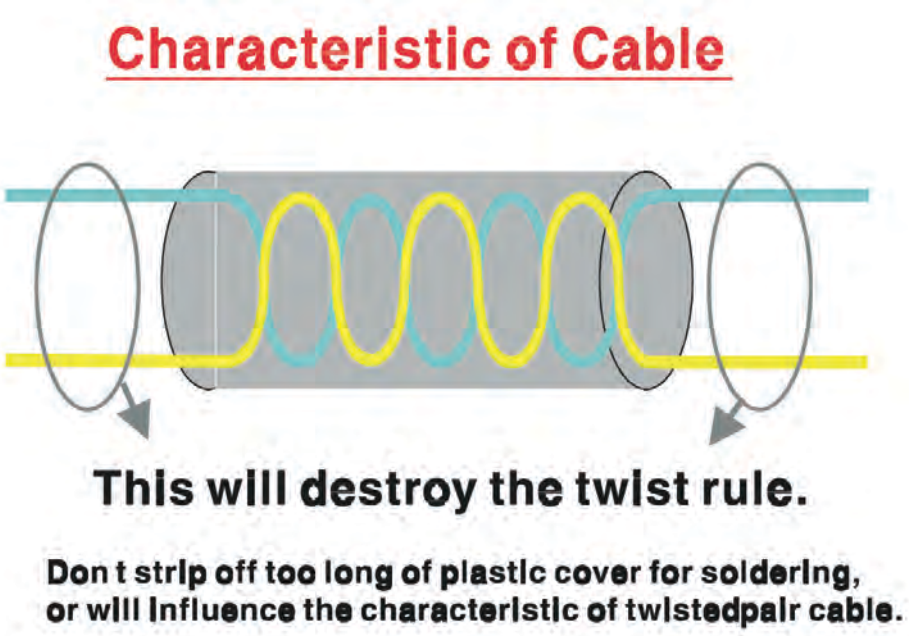


Figure C.12 The characteristic of the cable

- The characteristic of the cable Don't strip off too much insulation for soldering. This could change the effectiveness of the Shielded-Twisted-Pair cable and open a path to introduce unwanted noise.

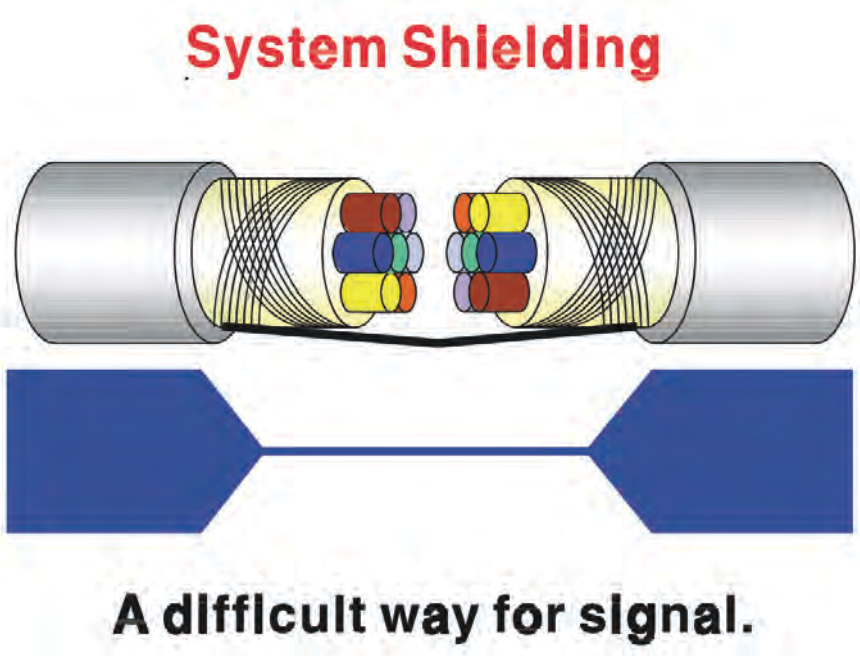


Figure C.13 System Shielding (1)

- Shield connection (1)

If you break into a cable, you might get in a hurry to achieve your goal. As in all electronic circuits, a signal will use the path of least resistance. If we make a poor connection between these two cables we will make a poor path for the signal. The noise will try to find another path for easier flow.

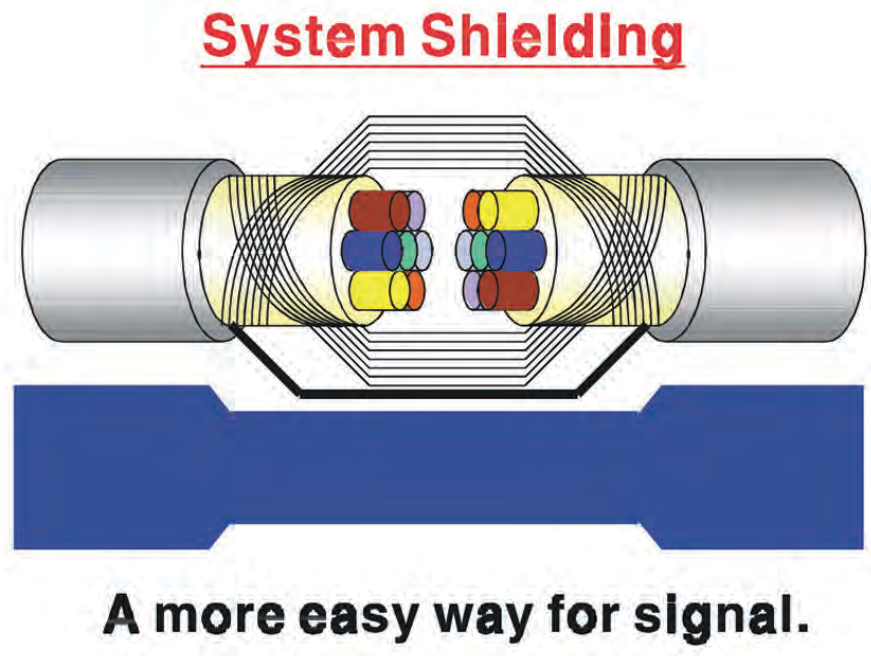


Figure C.14 System Shielding (2)

- Shield connection (2) The previous diagram shows you that the fill soldering just makes an easier way for the signal.

C.3 Noise Reduction Techniques

- Isolate noise sources in shielded 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.
- Check for stability in communication lines.
- Add another Grounding Bar if necessary.
- The diameter of power cable must be over 2.0 mm2.
- Independent grounding is needed for A/I, A/O, and communication network while using a 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 noisC.

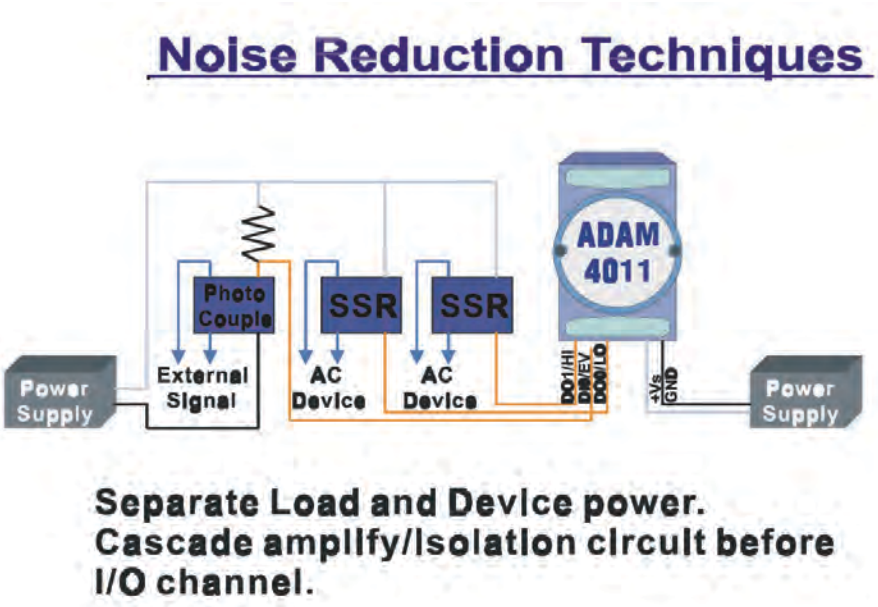


Figure C.15 Noise Reduction Techniques

C.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 correctly?
- Wire size is correct?
- Soldered connections are good?
- The terminal screw are tight?

ADVANTECH

Enabling an Intelligent Planet

www.advantech.com