ADAM-5000/TCP
Distributed DA&C System Based on Ethernet

ADVANTECH
Industrial Automation
Copyright Notice
This document is copyrighted, 2001, by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd., reserves the right to make improvements to the products described in this manual at any time without notice.
No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, nor for any infringements upon the rights of third parties which may result from its use.

Acknowledgments
IBM and PC are trademarks of International Business Machines Corporation.
Product Warranty

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 product which have been repaired or altered by other than repair personnel authorized by Advantech, or which have been subject to misuse, abuse, accident or improper installation. Advantech assumes no liability as a consequence of such events under the terms of this Warranty.

Because of Advantech’s high quality-control standards and rigorous testing, most of our customers never need to use our repair service. If an Advantech product ever does prove defective, it will be repaired or replaced at no charge during the warranty period. For out-of-warranty repairs, you will be billed according to the cost of replacement materials, service time and freight. Please consult your dealer for more details.

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

1. Collect all the information about the problem encountered (e.g. type of PC, CPU speed, Advantech products used, other hardware and software used etc.). Note anything abnormal and list any on-screen messages you get when the problem occurs.
2. Call your dealer and describe the problem. Please have your manual, product, and any helpful information readily available.
3. If your product is diagnosed as defective, you have to request an RAM number. When requesting an RMA (Return Material Authorization) number, please access ADVANTECH’s RMA web site: http://www.advantech.com.tw/rma. If the web server is shut down, please contact our office directly. You should fill in the “Problem Repair Form”, describing in detail the application environment, configuration, and problems encountered. Note that error descriptions such as “does not work” and “failure” are so general that we are then required to apply our internal standard repair process.
4. Carefully pack the defective product, a completely filled-out Repair and Replacement Order Card and a photocopy of dated proof of purchase (such as your sales receipt) in a shippable container. A product returned without dated proof of purchase is not eligible for warranty service.
5. Write the RMA number visibly on the outside of the package and ship it prepaid to your dealer.
Technical Support

We want you to get the maximum performance from your products. So if you run into technical difficulties, we are here to help. For most frequently asked questions you can easily find answers in your product documentation. Moreover, there are a huge database about trouble-shooting and knowledge Base as technical reference on our website. These answers are normally a lot more detailed than the ones we can give over the phone.

So please consult this manual or the web site first. If you still cannot find the answer, gather all the information or questions that apply to your problem and, with the product close at hand, call your dealer. Our dealers are well trained and ready to give you the support you need to get the most from your Advantech products. In fact, most problems reported are minor and are able to be easily solved over the phone.

In addition, free technical support is available from Advantech engineers every business day. We are always ready to give advice on application requirements or specific information on the installation and operation of any of our products.

Website information:

You can access the most current support on our website:

http://www.advantech.com/support/

If you find a problem with our documentation, please let us know by completing and returning the “Support Request Form” on our website:

http://www.advantech.com/support/request_dir.htm
## Organization of this manual

This Manual has six chapters, three appendices.

The following table lists each chapter or appendices with its corresponding title and a brief overview of the topics covered in it.

<table>
<thead>
<tr>
<th>Chapter / Appendix</th>
<th>Title</th>
<th>Topics Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Understanding Your System</td>
<td>Introduces the suitable applying industries and the position in a SCADA system. Summarize the features and the specification of ADAM-5000/TCP. Explains the functions of the LED indicators.</td>
</tr>
<tr>
<td>2</td>
<td>Selecting Your Hardware</td>
<td>Provides a briefly selection chart and specification table of ADAM-5000 I/O modules for users to organize their system easily. Give a direction to calculate system capacity and select a certain power supply. Recommend a standard for communication cable and connector.</td>
</tr>
<tr>
<td>3</td>
<td>Hardware Installation Guide</td>
<td>Lists the necessary components and proper environment in installing process. Describes the Hardware dimension and the way to place or mount it. Explains the rule of mapping I/O address. Describes the wiring and connecting detail for ADAM-5000/TCP.</td>
</tr>
<tr>
<td>4</td>
<td>I/O Module Introduction</td>
<td>Introduces the detail specifications functions and application wiring of each ADAM-5000 I/O modules.</td>
</tr>
<tr>
<td>5</td>
<td>System Configuration Guide</td>
<td>Guides users to use Windows Utility for network &amp; security setting, I/O range configuration, accuracy calibration, command setting, and so on.</td>
</tr>
<tr>
<td>6</td>
<td>Planning Your Application Program</td>
<td>Introduces the functions and structure of DLL drivers and command sets. Explain how to integrate these programming tools to plan your application program.</td>
</tr>
<tr>
<td>A</td>
<td>Design Worksheets</td>
<td>Provides organized worksheets for users to establish system configuration document in order.</td>
</tr>
<tr>
<td>B</td>
<td>Data Formats and I/O Range</td>
<td>Provides detail information about Data formats and I/O Range of Analog Module.</td>
</tr>
<tr>
<td>C</td>
<td>Grounding Reference</td>
<td>Explains the concepts about field grounding and shielding.</td>
</tr>
</tbody>
</table>
How to use this manual

The following flow chart demonstrates a thought process that you can use when you plan your ADAM-5000/TCP system.
Contents

Chapter 1 Understanding Your System .......... 1-1
  1-1 Introduction ................................................................. 1-2
  1-2 Major Features ............................................................... 1-3
      1-2-1 Communication Network ........................................... 1-3
      1-2-2 Modbus/TCP Protocol ................................................. 1-3
      1-2-3 Hardware Capacity & Diagnostic .................................. 1-3
      1-2-4 Communicating Isolation ............................................. 1-4
      1-2-5 Completed set of I/O modules for total solutions .......... 1-4
      1-2-6 Built-in real-time OS and watchdog timer .................... 1-4
      1-2-7 Software Support ...................................................... 1-4
      1-2-8 Security Setting ........................................................ 1-5
      1-2-9 UDP Data Stream ...................................................... 1-5
      1-2-10 Modbus Ethernet Data Gateway .................................. 1-5
  1-3 Technical specification of ADAM-5000/TCP System .. 1-6
      1-3-1 System .................................................................... 1-6
      1-3-2 Ethernet Communication ............................................ 1-6
      1-3-3 Serial Communication ............................................... 1-6
      1-3-4 Power ..................................................................... 1-7
      1-3-5 Isolation ................................................................... 1-7
      1-3-6 Mechanical ................................................................ 1-7
      1-3-7 Environment ............................................................. 1-7
      1-3-8 Dimensions ............................................................... 1-7
      1-3-9 Basic Function Block Diagram .................................... 1-8
  1-4 LED Status of ADAM-5000/TCP main unit ............. 1-8

Chapter 2 Selecting Your Hardware Components .... 2-1
  2-1 Selecting I/O Module ...................................................... 2-2
  2-2 Selecting Power Supply .................................................. 2-6
  2-3 Selecting Link Terminal and Cable ......................... 2-8
  2-4 Selecting Operator Interface ....................................... 2-10

Chapter 3 Hardware Installation Guide ............. 3-1
  3-1 Determining the proper environment ................. 3-2
      3-1-1 Check the content of shipping box ......................... 3-2
      3-1-2 System Requirement ................................................ 3-2
      3-1-3 I/O modules ........................................................... 3-2
  3-2 Installing your main unit and module ............... 3-3
<table>
<thead>
<tr>
<th>3-3 Mounting</th>
<th>3-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-3-1 Panel mounting</td>
<td>3-4</td>
</tr>
<tr>
<td>3-3-2 DIN rail mounting</td>
<td>3-4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3-4 Wiring and Connections</th>
<th>3-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-4-1 Power supply wiring</td>
<td>3-6</td>
</tr>
<tr>
<td>3-4-2 I/O modules wiring</td>
<td>3-7</td>
</tr>
<tr>
<td>3-4-3 System Network Connections</td>
<td>3-8</td>
</tr>
</tbody>
</table>

| 3-5 Assigning address for I/O Modules | 3-10 |

Chapter 4 I/O Module Introduction .......................... 4-1

Chapter 5 System Hardware Configuration .......... 5-1

<table>
<thead>
<tr>
<th>5-1 System Hardware Configuration</th>
<th>5-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-2 Install Utility Software on Host PC</td>
<td>5-3</td>
</tr>
<tr>
<td>5-3 ADAM-5000/TCP Windows Utility Overview</td>
<td>5-3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5-3-1 Main Menu</th>
<th>5-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-3-2 Ethernet Network Setting</td>
<td>5-5</td>
</tr>
<tr>
<td>5-3-3 Add Remote Station</td>
<td>5-8</td>
</tr>
<tr>
<td>5-3-4 I/O Module Configuration</td>
<td>5-9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5-3-5 Alarm Setting</th>
<th>5-14</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-3-6 I/O Module Calibration</td>
<td>5-15</td>
</tr>
<tr>
<td>5-3-7 Firmware Update</td>
<td>5-17</td>
</tr>
<tr>
<td>5-3-8 Security Setting</td>
<td>5-18</td>
</tr>
<tr>
<td>5-3-9 Terminal Emulation</td>
<td>5-19</td>
</tr>
<tr>
<td>5-3-10 Data Stream</td>
<td>5-20</td>
</tr>
<tr>
<td>5-3-11 Data Gateway Setting</td>
<td>5-22</td>
</tr>
</tbody>
</table>

Chapter 6 Planning Your Application Program....... 6-1

<table>
<thead>
<tr>
<th>6-1 Introduction</th>
<th>6-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-2 DLL (Dynamic Link Library) Driver</td>
<td>6-2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6-2-1 Index</th>
<th>6-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-2-2 Programming Flow</td>
<td>6-4</td>
</tr>
<tr>
<td>6-2-3 Function Descriptions</td>
<td>6-10</td>
</tr>
<tr>
<td>6-2-4 Return Codes</td>
<td>6-30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6-3 ADAM-5000/TCP Command</th>
<th>6-32</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-3-1 Command Structure</td>
<td>6-32</td>
</tr>
<tr>
<td>6-3-2 Modbus Function Code Introduction</td>
<td>6-33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6-4 Apply with ASCII Command for ADAM-5000/TCP System</th>
<th>6-39</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-4-1 Syntax of ASCII</td>
<td>6-39</td>
</tr>
<tr>
<td>6-4-2 System Command Set</td>
<td>6-40</td>
</tr>
<tr>
<td>6-4-3 Analog Input Command Set</td>
<td>6-48</td>
</tr>
<tr>
<td>6-4-4 Analog Output Command Set</td>
<td>6-127</td>
</tr>
<tr>
<td>6-4-5 Digital Input/Output Command Set</td>
<td>6-141</td>
</tr>
</tbody>
</table>
Appendix A Design Worksheets ....................... A-1

Appendix B Data Formats and I/O Ranges ........ B-1
  B.1 Analog Input Formats ...........................................B-2
  B.2 Analog Input Ranges - ADAM-5017 ..................B-4
  B.3 Analog Input Ranges - ADAM-5018 ..................B-5
  B.4 Analog Input Ranges - ADAM-5017H ..................B-7
  B.5 Analog Output Formats .................................B-8
  B.6 Analog Output Ranges .....................................B-8
  B.7 ADAM-5013 RTD Input Format and Ranges .......B-9

Appendix C Grounding Reference ....................... C-1
  C.1 Grounding ....................................................C-3
  C.2 Shielding ....................................................C-11
  C.3 Noise Reduction Techniques .........................C-17
  C.4 Check Point List ............................................C-18
Figures

Figure 1-1: Apply to System Application .......................................................... 1-2
Figure 1-2: ADAM-5000/TCP system & I/O module dimensions .............. 1-7
Figure 1-3: Function block diagram ................................................................. 1-8
Figure 1-4: ADAM-5000/TCP LED Indicators ................................................ 1-8

Figure 2-1: ADAM-5000 I/O Module Selection Chart .................................. 2-3
Figure 2-2: Ethernet Terminal and Cable Connection ................................. 2-8
Figure 2-3 RS-485 Terminal and Cable Connection ....................................... 2-9

Figure 3-1: Module alignment and installation .............................................. 3-3
Figure 3-2: Secure the module to the system .................................................. 3-3
Figure 3-3: ADAM-5000/TCP panel mounting screw placement ............... 3-4
Figure 3-4: ADAM-5000/TCP DIN rail mounting ........................................ 3-5
Figure 3-5: Secure ADAM-5000/TCP System to a DIN rail ....................... 3-5
Figure 3-6: ADAM-5000/TCP power wiring ............................................... 3-6
Figure 3-7: ADAM-5000 I/O Module Terminal Block wiring ..................... 3-7
Figure 3-8: System network connection ....................................................... 3-8
Figure 3-9 Serial Network Connection ......................................................... 3-9
Figure 3-10: I/O Modules Address Mapping .................................................. 3-10

Figure 5-1: Hardware Configuration ............................................................... 5-2
Figure 5-2: operation Screen ......................................................................... 5-3
Figure 5-4: Network Setting ........................................................................... 5-5
Figure 5-3: Tool Bar ....................................................................................... 5-5
Figure 5-5: Communication testing function ................................................. 5-6
Figure 5-6: Define Device Name and Description .......................................... 5-6
Figure 5-7: TCP/IP Network setting ............................................................... 5-7
Figure 5-8: Adding ADAM-5000/TCP screen ............................................... 5-8
Figure 5-9: Digital I/O Module Configuration ................................................. 5-9
Figure 5-11: Current Analog Input Status ..................................................... 5-10
Figure 5-10: Operating and Indicating Icons ............................................... 5-10
Figure 5-12: setting range and integration time ............................................ 5-11
Figure 5-13: Analog Module Configuration Screen ....................................... 5-12
Figure 5-14: Counter/Frequency Module Configuration .............................. 5-12
Figure 5-15: Location of Counter/Frequency Module .................................. 5-13
Figure 5-16: Alarm Setting for Analog Input and Counter Modules ........... 5-14
Figure 5-17: Zero Calibration ........................................................................ 5-15
Figure 5-18: Span Calibration ........................................................................ 5-15
Tables

Table 2-1: I/O Selection Guidelines .......................................................... 2-2
Table 2-2: I/O Modules Selection Guide .................................................... 2-5
Table 2-3: Power Consumption of ADAM-5000 series .......................... 2-6
Table 2-4: Power Supply Specification Table ........................................... 2-7
Table 2-5: Ethernet RJ-45 port Pin Assignment ...................................... 2-8

Table 4-1: I/O module support List ....................................................... 4-1

Table 6-1: Response Comment Structure .............................................. 6-33
Table 6-2: CPU Command Set Table .................................................... 6-40
Table 6-3 Baud rate codes ..................................................................... 6-42
Table 6-4: ADAM-5013 RTD Input command Set Table ....................... 6-49
Table 6-5: ADAM-5017/5018 Analog Input command Set Table .......... 6-63
Table 6-6: ADAM-5017H Analog Input command Set Table ................. 6-80
Table 6-7 Analog Input alarm command set table ................................ 6-89
Table 6-8 Analog Input alarm command set table ............................... 6-108
Table 6-9: Analog Output command Set Table ...................................... 6-127
Table 6-10: Counter/Frequency Command Set Table ......................... 6-149

Table A-1: I/O Data Base ..................................................................... A-3
Table A-2: Summary Required Modules ................................................. A-5
Table A-3: Table for Programming ...................................................... A-6
# Chapter 1

**Understanding Your System**

## Using this Chapter

<table>
<thead>
<tr>
<th>If you want to read about</th>
<th>Go to page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1-2</td>
</tr>
<tr>
<td>Major Feature</td>
<td>1-3</td>
</tr>
<tr>
<td>Technical Specification</td>
<td>1-6</td>
</tr>
<tr>
<td>LED Status of ADAM-5000/TCP main unit</td>
<td>1-8</td>
</tr>
</tbody>
</table>
Chapter 1  Understanding Your System

1-1 Introduction

Undoubtedly, Ethernet connectivity is becoming to a big trend for industrial applications. Longer communication distances, faster communication speeds, and greater advantages attract people into developing their system based upon this network scenario. 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, 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. See figure 1-1.

Figure 1-1: Apply to System Application
1-2 Major Features

1-2-1 Communication Network
By adopting a 32-bit RISC CPU, the ADAM-5000/TCP 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 auto-switched 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 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 when they install the DA&C system with their own operating application. It will certainly reduce engineer effort. Moreover, the ADAM-5000/TCP 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 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 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 Ethernet port from dangerous voltage up to 1500V_{DC} power spikes and avoid surge damage to whole system.

1-2-5 Completed set of I/O modules for total solutions
The ADAM-5000/TCP 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 to support your applications (not include ADAM-5090).

Full ranges of digital module supports 10 to 30 V_{DC} 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. The 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 ideal for use in applications which require a high level of system performance and stability.

1-2-7 Software Support
Based on the Modbus standard, the ADAM-5000/TCP firmware is a built-in Modbus/TCP server. Therefore, Advantech provides the necessary DLL drivers, OCX component OPC Server, and Windows Utility for users for client data for the ADAM-5000/TCP. Users can configure this DA&C system via Windows Utility; integrate with HMI software package via Modbus/TCP driver or Modbus/TCP OPC Server. Even more, you can use the DLL driver or OCX component 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. 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 packs to update current data. It may cause to data collision and lower performance on the network, especially when there are frequent communication between multi-servers and I/O devices. To reduce the communication loading of the host computer on your Ethernet network, the ADAM-5000/TCP also supports UDP (User Datagram Protocol) protocol to broadcast the data packs to specific IPs without requesting commands. Users can apply this great feature to implement Data Stream, Event Trigger, and other advanced functions.

1-2-10 Modbus Ethernet Data Gateway

Much more than an I/O system, ADAM-5000/TCP provides an RS-485 network interface for other Modbus devices integration. It works as Ethernet Data Gateway, upgrading Modbus serial network devices up to Ethernet layer. Maximum 32 nodes of ADAM-5511 or 3’rd party products supported Modbus protocol are allowed to integrate with an ADAM-5000/TCP. This great feature enlarges your system scope, as opposed to other general dummy I/O system.
1-3 Technical specification of ADAM-5000/TCP System

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 (four ADAM-5024 allowed)
- **Status Indicator:**
  - Power (3.3V, 5V), CPU, Communication (Link, Collide, 10/100 Mbps, Tx, Rx)
- **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:** RJ45 modular jack
- **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$_{\text{DC}}$
• Protection: Over-voltage and power reversal

1-3-5 Isolation
• Ethernet Communication: 1500 V$_{\text{DC}}$
• I/O Module: 3000 V$_{\text{DC}}$

1-3-6 Mechanical
• Case: KJW with captive mounting hardware
• Plug-in Screw Terminal Block:
  Accepts 0.5 mm 2 to 2.5 mm 2, 1 - #12 or 2 - #14 to #22 AWG

1-3-7 Environment
• Operating Temperature: -10 to 70°C (14 to 158°F)
• Storage Temperature: -25 to 85°C (-13 to 185°F)
• 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 1-2: ADAM-5000/TCP system & I/O module dimensions
1-3-9 Basic Function Block Diagram

![Function block diagram](image)

Figure 1-3: Function block diagram

1-4 LED Status of ADAM-5000/TCP main unit

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

![ADAM-5000/TCP LED Indicators](image)

Figure 1-4: ADAM-5000/TCP LED Indicators
(1) **3.3V**: Red indicator. This LED is normal on when ARM CPU is powered on.

(2) **5V**: Red indicator. This LED is normal on when ADAM-5000/TCP system is powered on.

(3) **Run**: Green indicator. This LED is regularly blinks whenever the ADAM-5000/TCP system is running.

(4) **Link**: Green Indicator. This LED is normal on whenever the ADAM-5000/TCP’s Ethernet wiring is connected.

(5) **Tx**: Green indicator. This LED is designed for the spare function (COM port transit indicator) in the future.

(6) **Rx**: Green indicator. This LED is designed for the spare function (COM port receive indicator) in the future.

(7) **Collide**: Green indicator. This LED blinks whenever there is the Ethernet data pack collision.

(8) **Speed**: Green indicator. This LED is on when the Ethernet communication speed is 100 Mbps.

(9) **Rx (RJ-45)**: Green indicator. This LED blinks whenever the ADAM-5000/TCP transmitting data to Ethernet.

(10) **Tx (RJ-45)**: Yellow indicator. This LED blinks whenever the ADAM-5000/TCP receiving data from Ethernet.
Chapter 2
Selecting Your Hardware Components

Using this Chapter

If you want to read about | Go to page
--- | ---
Selecting I/O Module | 2-2
Selecting Power Supply | 2-6
Selecting Link Terminal & Cable (Ethernet) | 2-8
Selecting Operator Interface | 2-10
2-1 Selecting I/O Module

To organize an ADAM-5000/TCP 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 should be considered when you select the I/O modules.

What type of I/O signal is applied in your system?
How much I/O is required to your system?
How will you place the main unit for concentrate the I/O points of an entire process.
How many ADAM-5000/TCP 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 as module selection guidelines

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

Table 2-1: I/O Selection Guidelines
Advantech provides 15 types of ADAM-5000 I/O modules for various applications so far. The 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](image_url)
## Chapter 2  
Selecting Your Hardware Components

<table>
<thead>
<tr>
<th>Module</th>
<th>ADAM-5013</th>
<th>ADAM-5017</th>
<th>ADAM-5017H</th>
<th>ADAM-5018</th>
<th>ADAM-5024</th>
<th>ADAM-5050</th>
<th>ADAM-5051</th>
<th>ADAM-5051D</th>
<th>ADAM-5051S</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Analog Input</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>16 bit</td>
<td>16 bit</td>
<td>12 bit</td>
<td>16 bit</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Channel</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling Rate</td>
<td>10</td>
<td>10</td>
<td>8K</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voltage Input</td>
<td></td>
<td>±150 mV</td>
<td>±500 mV</td>
<td>±1 V</td>
<td>±5 V</td>
<td>±10 V</td>
<td>±15 mV</td>
<td>±50 mV</td>
<td>±100 mV</td>
</tr>
<tr>
<td>Current Input</td>
<td>-</td>
<td>±20 mA*</td>
<td>±20 mA*</td>
<td>±20 mA*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direct Sensor Input</td>
<td>Pt or Ni</td>
<td>-</td>
<td>-</td>
<td>J, K, T, E, R, S, B</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Analog Output</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>12 bit</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voltage Output</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0-10 V</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current Output</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0-20 mA</td>
<td>4-20 mA</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>Digital Input and Digital Output</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital Input Channels</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16 DIO (bit-wise selectable)</td>
<td>16</td>
<td>16</td>
<td>WLED</td>
</tr>
<tr>
<td>Digital Output Channels</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>Counter (32-bit)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channels</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Input Frequency</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>COMM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channels</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Isolation</td>
<td>3000 V&lt;sub&gt;AC&lt;/sub&gt;</td>
<td>3000 V&lt;sub&gt;AC&lt;/sub&gt;</td>
<td>3000 V&lt;sub&gt;AC&lt;/sub&gt;</td>
<td>3000 V&lt;sub&gt;AC&lt;/sub&gt;</td>
<td>3000 V&lt;sub&gt;AC&lt;/sub&gt;</td>
<td>-</td>
<td>-</td>
<td>2500 V&lt;sub&gt;AC&lt;/sub&gt;</td>
<td></td>
</tr>
</tbody>
</table>
## Chapter 2

### Selecting Your Hardware Components

#### Table 2-2: I/O Modules Selection Guide

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Input Channel</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sampling Rate</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Voltage Input</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Current Input</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Direct Sensor Input</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Analog Output</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Voltage Output</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Current Output</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Digital Input and Digital Output</td>
<td>8 8 - - - - -</td>
<td>8 8 - - - - -</td>
<td>8 8 - - - - -</td>
<td>8 8 - - - - -</td>
<td>8 8 - - - - -</td>
<td>8 8 - - - - -</td>
<td>8 8 - - - - -</td>
<td>8 8 - - - - -</td>
<td>8 8 - - - - -</td>
</tr>
<tr>
<td>Digital Input Channels</td>
<td>- 16 16 WLED</td>
<td>- 16 16 WLED</td>
<td>- 16 16 WLED</td>
<td>- 16 16 WLED</td>
<td>- 16 16 WLED</td>
<td>- 16 16 WLED</td>
<td>- 16 16 WLED</td>
<td>- 16 16 WLED</td>
<td>- 16 16 WLED</td>
</tr>
<tr>
<td>Digital Output Channels</td>
<td>8 8 16 16 WLED</td>
<td>8 8 16 16 WLED</td>
<td>8 8 16 16 WLED</td>
<td>8 8 16 16 WLED</td>
<td>8 8 16 16 WLED</td>
<td>8 8 16 16 WLED</td>
<td>8 8 16 16 WLED</td>
<td>8 8 16 16 WLED</td>
<td>8 8 16 16 WLED</td>
</tr>
<tr>
<td>Counter (32-bit)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channels</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>Input Frequency</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5000 Hz (max)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mode</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Frequency, Up/Down Counter, Bi-direction Counter</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>COMM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channels</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>Type</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>RS-232</td>
<td>-</td>
</tr>
<tr>
<td>Isolation</td>
<td>5000 Vrms 2500 Vdc</td>
<td>5000 Vrms 2500 Vdc</td>
<td>5000 Vrms 2500 Vdc</td>
<td>5000 Vrms 2500 Vdc</td>
<td>5000 Vrms 2500 Vdc</td>
<td>5000 Vrms 2500 Vdc</td>
<td>5000 Vrms 2500 Vdc</td>
<td>5000 Vrms 2500 Vdc</td>
<td>5000 Vrms 2500 Vdc</td>
</tr>
</tbody>
</table>
2-2 Selecting Power Supply

ADAM-5000/TCP 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 main unit and each I/O module.

<table>
<thead>
<tr>
<th>Main Description</th>
<th>Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAM-5000/485</td>
<td>1.0 W</td>
</tr>
<tr>
<td>ADAM-5000E</td>
<td>4.0 W</td>
</tr>
<tr>
<td>ADAM-5000/TCP</td>
<td>5.0 W</td>
</tr>
<tr>
<td>ADAM-5510</td>
<td>1.0 W</td>
</tr>
<tr>
<td>ADAM-5511</td>
<td>1.0 W</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I/O Modules</th>
<th>Description</th>
<th>Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAM-5013</td>
<td>3-Channel RTD Input Module</td>
<td>1.1 W</td>
</tr>
<tr>
<td>ADAM-5017</td>
<td>8-Channel Analog Input Module (mV, mA or High Voltage)</td>
<td>1.25 W</td>
</tr>
<tr>
<td>ADAM-5017H</td>
<td>8-Channel High speed Analog Input Module (mV, mA or High Voltage)</td>
<td>2.2 W</td>
</tr>
<tr>
<td>ADAM-5018</td>
<td>7-Channel Thermocouple Input Module (mV, mA, Thermocouple)</td>
<td>0.63 W</td>
</tr>
<tr>
<td>ADAM-5024</td>
<td>4-Channel Analog Output Module (V, mA)</td>
<td>2.9 W</td>
</tr>
<tr>
<td>ADAM-5050</td>
<td>16-Channel Universal DIO</td>
<td>1.2 W</td>
</tr>
<tr>
<td>ADAM-5051</td>
<td>16-Channel Digital Input Module</td>
<td>0.53 W</td>
</tr>
<tr>
<td>ADAM-5051D</td>
<td>16-Channel Digital Input W/LED Module</td>
<td>0.84 W</td>
</tr>
<tr>
<td>ADAM-5051S</td>
<td>16-Channel Isolated Digital Input W/LED Module</td>
<td>0.8 W</td>
</tr>
<tr>
<td>ADAM-5052</td>
<td>8-Channel Isolated DI</td>
<td>0.27 W</td>
</tr>
<tr>
<td>ADAM-5055S</td>
<td>16-Channel Isolated Digital I/O W/LED Module</td>
<td>0.68 W</td>
</tr>
<tr>
<td>ADAM-5056</td>
<td>16-Channel Digital Output Module</td>
<td>0.53 W</td>
</tr>
<tr>
<td>ADAM-5056D</td>
<td>16-Channel Digital Output W/LED Module</td>
<td>0.84 W</td>
</tr>
<tr>
<td>ADAM-5056S</td>
<td>16-Channel Isolated Digital Output W/LED Module</td>
<td>0.6 W</td>
</tr>
<tr>
<td>ADAM-5060</td>
<td>6-Channel Relay Output Module (2 of Form A, 4 of Form C)</td>
<td>1.8 W</td>
</tr>
<tr>
<td>ADAM-5068</td>
<td>8-Channel Relay Output Module (8 of Form A)</td>
<td>1.8 W</td>
</tr>
<tr>
<td>ADAM-5069</td>
<td>8-Channel Power Relay Output Module (4 of Form A &amp; 4 of Form C)</td>
<td>2.2 W</td>
</tr>
<tr>
<td>ADAM-5080</td>
<td>4-Channel Counter/Frequency Input Module</td>
<td>1.5 W</td>
</tr>
<tr>
<td>ADAM-5090</td>
<td>4-Port RS-232 Module</td>
<td>0.6 W</td>
</tr>
</tbody>
</table>

Table 2-3: Power Consumption of ADAM-5000 series
Selecting Your Hardware Components

Chapter 2

Calculate the Summary of the whole system’s power consumption. For example, there are 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

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

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

*Table 2-4: Power Supply Specification Table*
2-3 Selecting Link Terminal and Cable

Ethernet Network

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

![Ethernet Terminal and Cable Connection](image)

**Figure 2-2: Ethernet Terminal and Cable Connection**

<table>
<thead>
<tr>
<th>Pin number</th>
<th>Signal</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RD+</td>
<td>Receive (+)</td>
</tr>
<tr>
<td>2</td>
<td>RD-</td>
<td>Receive (-)</td>
</tr>
<tr>
<td>3</td>
<td>TD+</td>
<td>Transmit (+)</td>
</tr>
<tr>
<td>4</td>
<td>(Not Used)</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>(Not Used)</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>TD-</td>
<td>Transmit (-)</td>
</tr>
<tr>
<td>7</td>
<td>(Not Used)</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>(Not Used)</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 2-5: Ethernet RJ-45 port Pin Assignment**
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 the 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. Adopting by Modbus/TCP Protocol, ADAM-5000/TCP exhibits high ability in system integration for various applications.

If you want to configure your ADAM-5000/TCP system, or monitor current status, Advantech offers free charge software:

- ADAM-5000/TCP Windows Utility

If you want to integrate ADAM-5000/TCP 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 driver.

- Advantech Studio
- Wonderware InTouch
- Intellution Fix of i-Fix
- Any other software support Modbus/TCP protocol

Moreover, Advantech also provides OPC Server, the most easy-to-use data exchange tool in worldwide. Any HMI software designed with OPC Client would be able to access ADAM-5000/TCP system.

- Modbus/TCP OPC Server

If you want to develop your own application, the DLL driver and OCX component will be the best tools to build up user’s operator interface.

- ADAM-5000/TCP DLL driver
- ADAM-5000/TCP OCX component

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.
Chapter 3
Hardware Installation Guide

Using this Chapter

<table>
<thead>
<tr>
<th>If you want to read about</th>
<th>Go to page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determining the proper environment</td>
<td>3-2</td>
</tr>
<tr>
<td>Installing your main unit and module</td>
<td>3-3</td>
</tr>
<tr>
<td>System Mounting</td>
<td>3-4</td>
</tr>
<tr>
<td>Wiring and Connection</td>
<td>3-6</td>
</tr>
<tr>
<td>System Network Connection</td>
<td>3-8</td>
</tr>
<tr>
<td>Assigning address for I/O modules</td>
<td>3-10</td>
</tr>
</tbody>
</table>
3-1 Determining the proper environment

Before you start to install the ADAM-5000/TCP system, there are something needed to check.

3-1-1 Check the content of shipping box

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

- ADAM-5000/TCP main unit with two blank slot covers
- ADAM-4000/5000 Products Utility CD

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 Cable with RJ-45 connector
- Power supply for ADAM-5000/TCP (+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
Chapter 3  Hardware Installation Guide

3-3  Mounting

The ADAM-5000/TCP 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](image)

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 each end of the 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 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

Ethnet Network

The ADAM-5000/TCP has an Ethernet communication port allowed you to program, configure, monitor, and integrate into the SCADA system. The 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 provides an RS-485 interface to integrate serial devices for various applications. Adopting by Modbus standard protocol, it solves the communication problem between different networks and different devices. Mean while, users can extend their system scope by integrating up to 32 nodes of ADAM-5511 or other Modbus products, such as meters, card readers, loadcell, and so on.

![Serial Network Connection Diagram]

*Figure 3-9 Serial Network Connection*

Note: The address of ADAM-5000/TCP 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

Basing on Modbus standard, the addresses of the I/O modules you place into the ADAM-5000/TCP system are defined by a simple rule. Please refer the 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 40017~40020.

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.
I/O modules
This manual introduces the detail specifications functions and application wiring of each ADAM-5000 I/O modules. 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. Following table is the I/O modules support list we provided for user’s choice. **More detailed specification and user’s guides, please refer the user’s manual of ADAM-5000 IO Module.** It had integrated and collected this information.

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

*Table 4-1: I/O Module Support List*
**Chapter 5**

**System Hardware Configuration**

Using this Chapter

<table>
<thead>
<tr>
<th>If you want to read about</th>
<th>Go to page</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Hardware Configuration</td>
<td>5-2</td>
</tr>
<tr>
<td>Install Utility Software</td>
<td>5-3</td>
</tr>
<tr>
<td>I/O Module Configuration</td>
<td>5-9</td>
</tr>
<tr>
<td>Ethernet Network Setting</td>
<td>5-5</td>
</tr>
<tr>
<td>I/O Module Calibration</td>
<td>5-15</td>
</tr>
<tr>
<td>Security Setting</td>
<td>5-18</td>
</tr>
<tr>
<td>Technical Emulation</td>
<td>5-19</td>
</tr>
<tr>
<td>UDP Data Stream</td>
<td>5-20</td>
</tr>
<tr>
<td>Modbus Data Gateway</td>
<td>5-22</td>
</tr>
</tbody>
</table>
This chapter explains how to use Windows Utility to configure the ADAM-5000/TCP 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 following items to complete your system hardware configuration.

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 Cable with RJ-45 connector
- Power supply for ADAM-5000/TCP (+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](image-url)
5-2 Install Utility Software on Host PC

ADAM-5000/TCP Systems come packaged with a Utility CD, containing ADAM Product series Utilities as system configuration tool. While you insert the CD into the CD drive (e.g. D:) of the host PC, the Utility software setup menu will start up automatically.

Click the ADAM-5000/TCP 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 ADAM-5000/TCP Windows Utility Overview

The Windows Utility offers a graphical interface that helps you configure the ADAM-5000/TCP 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
- I/O Module Calibration
- Firmware Update
- Security Setting
- Terminal emulation
- Data Stream
- RS-485 Modbus Network Setting

5-3-1 Main Menu

Double Click the icon of ADAM-5000/TCP Windows Utility shortcut, the Operation screen will pop up as Figure 5-2.

![Figure 5-2: operation Screen](image-url)
Chapter 5  System Hardware Configuration

The top of the operation screen consists of a function menu and a tool bar for user’s commonly operating functions.

**Function menu**

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File</strong></td>
<td>contents “Exit” Function, using to exit this Utility program.</td>
</tr>
<tr>
<td><strong>Tool</strong></td>
<td>contents functions as below:</td>
</tr>
<tr>
<td><strong>Add Remote 5000/TCP</strong></td>
<td>Create a new ADAM-5000/TCP located in other Ethernet domination, both available to local LAN and Internet application.</td>
</tr>
<tr>
<td><strong>Search for 5000/TCP</strong></td>
<td>Search all ADAM-5000/TCP units in the specific Ethernet domination. (the same with host PC’s Ethernet domination)</td>
</tr>
<tr>
<td><strong>Refresh 5000/TCP</strong></td>
<td>Refresh the specific ADAM-5000/TCP unit to verify the system status.</td>
</tr>
<tr>
<td><strong>Terminal</strong></td>
<td>Call up the operation screen of Terminal emulation to do the request / response command execution.</td>
</tr>
<tr>
<td><strong>Monitor Data Stream</strong></td>
<td>Call up the monitoring screen of stream data from specific ADAM-5000/TCP.</td>
</tr>
<tr>
<td><strong>Setup</strong></td>
<td>contents Timeout and Scan Rate setting functions. Please be aware of the time setting for other Ethernet domination usually longer than local network.</td>
</tr>
<tr>
<td><strong>About</strong></td>
<td>contents information about software version, released date, and support modules.</td>
</tr>
</tbody>
</table>

5-4  ADAM-5000/TCP User’s Manual
Tool Bar
There are five push buttons in the tool bar.

5-3-2 Ethernet Network Setting
As the moment you start up this Windows Utility, it will search all ADAM-5000/TCP on the host PC’s domination Ethernet network automatically. Then the tree-structure display area will appeal with the searched units and the relative IP address.
See Figure 5-4, there are also Host PC’s information in the status display area, include host name and IP address. Moreover, the Windows Utility provides network connection test tool for user to verify whether the communication is workable. Key-in the specific IP address you want to connect and click the PING button, the testing result will show as Figure 5-5.

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

**Step1:** Choose any one station, all I/O modules plugged in the main unit will be listed on the tree-structure display area. Mean while, the “Device Name” and “Device Description” are editable by operator’s needs.

**Figure 5-6: Define Device Name and Description**
Step 2: Click the Network tip to configure the TCP/IP network setting

**Figure 5-7: TCP/IP Network setting**

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

**Link Speed:** This function will show the current linking speed to be either 10Mbps or 100Mbps. However, the utility will auto-detect the current transmission speed on the network segment and set the transmission speed for the device accordingly without your further efforts.

**Duplex Mode:** The utility will detect the current transmission mode (half-duplex or full-duplex) on the network segment, and set the transmission mode for the device accordingly without your further efforts.

**IP Address, Subnet Mask, Default Gateway:**

The IP address identifies your ADAM-5000/TCP 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 at the same time to avoid the Ethernet collision.

If you want to configure the ADAM-5000/TCP in the host PC’s dominating network, only the IP address and Subnet Mask will need to set (host PC and ADAM-5000/TCP must belong to same subnet Mask).

If you want to configure the ADAM-5000/TCP via Internet or other network domination, you have to ask your network administrator to obtain a specific IP and Gateway addresses then configure each ADAM-5000/TCP with the individual setting.
5-3-3 Add Remote Station

To meet the remote monitoring and maintenance requirements, ADAM-5000/TCP System does not only available to operate in local LAN, but also allowed to access from internet or intranet. Thus users would able to configure an ADAM-5000/TCP easily no matter how far it is.

Select item Tool>Add 5000/TCP in function menu or click the button, the adding station screen will pop up as Figure 5-8. Then key-in the specific IP address and click the Add button. If the communication success, the added ADAM-5000/TCP unit should appeal on the tree-structure display area.

![Adding ADAM-5000/TCP screen](image)

**Figure 5-8: Adding ADAM-5000/TCP screen**

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

1. Be sure the specific IP is existed and available.
2. Be sure to complete the network linkage for both sides.
3. Be sure to adjust the best timing of timeout setting.
4. Even you are not sure whether the communication is workable or not, there is also a “PING” function for testing the network connection.
5-3-4  I/O Module Configuration

Digital Input Output Module

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

![Image of Digital I/O Module Configuration](image)

**Figure 5-9: Digital I/O Module Configuration**

**Location:**  Standard Modbus address. Windows Utility shows the Modbus mapping address of each I/O channel. (Please refer to chapter 3-5 Assigning address for I/O Modules) And the addresses will be the indexes for applying into the database of 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 monitor the current DI/DO status, the Windows Utility offers a graphical operating interface as figure 5-10. You can read the Digital input status through the change of the indicator icons. Oppositely, you can write the digital output status through clicking the indicator icons.
Figure 5-10: Operating and Indicating 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

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

Figure 5-11: Current Analog Input Status

Location: Standard Modbus address. (Refer to chapter 3-5 Assigning address 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 range 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 engineer unit.

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

Analog Output Module

Selecting an ADAM-5024 Analog Output Module, users can certainly read the information about location, type, value, and Description. Actually, ADAM-5024 is designed with four different outputs channel, 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.
Chapter 5  System Hardware Configuration

Figure 5-13: Analog Module Configuration Screen

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

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 5-14: 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 there is a ADAM-5080 plugged in Slot 6 of ADAM-5000/TCP system, the address locations should be:

<table>
<thead>
<tr>
<th>Slot: 0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
</table>

![Figure 5-15: Location of Counter/Frequency Module](image)

**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 tuig ON “1” when counter overflow. Users can write "0" to clear the overflow flag.

4th bit: Non used.
To satisfy the needs of various applications, ADAM-5000/TCP 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 5-16: Alarm Setting for Analog Input and Counter Modules

There are three alarm types in Analog Input Modules:

**Disable:** ADAM-5000/TCP does 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 include “Disable” and “Latch only.”
5-3-6  I/O Module Calibration

Calibration is to adjust the accuracy of ADAM module. There are several modes for module’s calibration: Zero calibration, Span calibration, CJC calibration, and Analog Output calibration. Only analog input and output modules can be calibrated, includes ADAM-5013, 5017, 5017H, 5018 and 5024.

Zero Calibration

1. Apply power to the module and let it warm up for 30 minutes.
2. Make sure the module is correctly installed and properly configured for the input range you want to calibrate.
3. Use a precision voltage source to apply a calibration voltage to the V+ and V- terminals of the ADAM-5013, 5017, 5017H, and 5018 modules.
4. Click the Execute button.

![Figure 5-17: Zero Calibration](image)

Span Calibration

Follow the same procedure of zero calibration and click the Execute button.

![Figure 5-18: Span Calibration](image)
**CJC Calibration**

1. Prepare an accurate voltage source.
2. Run the zero calibration and span calibration function.
3. Use a temperature emulation device (such as Micro-10) to send a temperature signal to the ADAM module and then compare this signal with the value from the ADAM module. If the value is different from the signal, adjust the CJC value to improve it.

![Figure 5-19: CJC Calibration](image)

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

**Analog Output Calibration**

- ADAM 5024: 4 mA and 20 mA

![Figure 5-20: Analog Output Module Calibration](image)
5-3-7 Firmware Update

ADAM-5000/TCP supports all ADAM-5000 series I/O modules and necessary operating function so far. But Advantech always provides better hardware and software functions to improve the perfect DA&C systems. Therefore, users will need to upgrade the firmware of ADAM-5000/TCP sometime.

Select the Firmware Upgrade tab and click **Browsing** to find the specific firmware (*.bin) for upgrade.

![Figure 5-21: Firmware Upgrade](image)

Click the upgrade button, then the new firmware will be downloaded into the ADAM-5000/TCP system.
5-3-8 Security Setting

Though the technology of Ethernet discovered with great benefits in speed and integration, there also exist risk about network invading form anywhere. For the reason, the security protection design has built-in ADAM-5000/TCP system. Once user setting the password into the ADAM-5000/TCP firmware, the important system configurations (Network, Firmware, Password) are only allowed to be changed by password verification.

Note: The default password of ADAM-5000/TCP is “00000000”. Please make sure to keep the correct password by yourself. If you lose it, please contact to Advantech’s technical support center for help.
5-3-9 Terminal Emulation

You can issue commands and receive response by clicking the Terminal button on the tool bar. There are two kinds of command format supported by this emulating function. Users can choose ASCII or Hexadecimal mode as their communication base. If the ASCII mode has been selected, the Windows Utility will translate the request and response string both in Modbus and ASCII format. Please refer Chapter 6-2 to use Modbus Command; and refer Chapter 6-4 to apply ASCII command.

For example, select ASCII mode and key-in the ASCII command “$01M” (read module name), then click Send. The response will show as figure 5-23.

![Figure 5-23: Command Emulation](image_url)
5-3-10 Data Stream

Data Stream Configuration

In addition to TCP/IP communication protocol, ADAM-5000/TCP 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. This UDP Data Stream function broadcasts up to 8 host PCs simultaneously, and the interval is user-defined from 50ms to 7 Days.

Figure 5-24: 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 Figure 5-25.
Select the IP address of the ADAM-5000/TCP you want to read data, then click “Start” button. The Utility software will begin to receive the stream data on this operation display.
5-3-11 Data Gateway Setting

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

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

1. Define the parameter of the network, includes 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 done.
Chapter 6
Planning Your Application Program

Using this Chapter

If you want to read about | Go to page
--- | ---
DLL Driver | 6-2
Programming Flow | 6-4
Programming Flow | 6-32
Command Structure | 6-33
Modbus Function Code Introduction | 6-39
Apply with ASCII Command | 6-40
- System Command Set | 6-48
- Analog Input Command Set | 6-127
- Analog Output Command Set | 6-140
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 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 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 systems to their fullest extent.

In the same path with “ADAM 5000TCP” 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 setup program or monitoring system.

6-2-1 Index

<table>
<thead>
<tr>
<th>Function Libraries</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAM5KTCP_Open</td>
<td>6-9</td>
</tr>
<tr>
<td>ADAM5KTCP_Close</td>
<td>6-10</td>
</tr>
<tr>
<td>ADAM5KTCP_Connect</td>
<td>6-11</td>
</tr>
<tr>
<td>ADAM5KTCP_Disconnection</td>
<td>6-12</td>
</tr>
<tr>
<td>ADAM5KTCP_GetDLLVersion</td>
<td>6-13</td>
</tr>
<tr>
<td>ADAM5KTCP_ReadReg</td>
<td>6-14</td>
</tr>
<tr>
<td>ADAM5KTCP_WriteReg</td>
<td>6-15</td>
</tr>
<tr>
<td>Function Libraries</td>
<td>Pages</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>ADAM5KTCP_ReadCoil</td>
<td>6-16</td>
</tr>
<tr>
<td>ADAM5KTCP_WriteCoil</td>
<td>6-17</td>
</tr>
<tr>
<td>ADAM5KTCP_SendReceive5KTCPCmd</td>
<td>6-18</td>
</tr>
<tr>
<td>ADAM5KTCP_Add5KTCPForStream</td>
<td>6-19</td>
</tr>
<tr>
<td>ADAM5KTCP_ReadStreamData</td>
<td>6-20</td>
</tr>
<tr>
<td>ADAM5KTCP_ReadAlarmInfo</td>
<td>6-21</td>
</tr>
<tr>
<td>ADAM5KTCP_StartStream</td>
<td>6-22</td>
</tr>
<tr>
<td>ADAM5KTCP_StopStream</td>
<td>6-23</td>
</tr>
<tr>
<td>ADAM5KTCP_SetStreamAlarmState</td>
<td>6-24</td>
</tr>
<tr>
<td>ADAM5KTCP_Debug</td>
<td>6-25</td>
</tr>
<tr>
<td>ADAM5KTCP_UDPOpen</td>
<td>6-26</td>
</tr>
<tr>
<td>ADAM5KTCP_UDPClose</td>
<td>6-27</td>
</tr>
<tr>
<td>ADAM5KTCP_SendReceiveUDPCmd</td>
<td>6-28</td>
</tr>
</tbody>
</table>
6-2-2 Programming Flow

* Send a command and receiving response by UDP

```
ADAM5KTCP_Open()
→
ADAM5KTCP_SendReceiveUDPCmd()
→
ADAM5KTCP_UDPOpen()
→
ADAM5KTCP_SendReceiveUDPCmd()
→
ADAM5KTCP_UDPClose()
→
ADAM5KTCP_Close()
```
* Send a command and receiving response by TCP

```
ADAM5KTCP_Open()
ADAM5KTCP_Connect()
ADAM5KTCP_SendReceive5KTCPCmd()
ADAM5KTCP_Disconnect()
ADAM5KTCP_Close()
```
* To receive stream data coming from ADAM-5000/TCP(s)

![Flowchart Diagram]

1. `ADAM5KTCP_Open()`
2. `ADAM5KTCP_Add5KTCPForStream()`
3. `hEvent=CreateEvent()` (A Win32 API)
4. `ADAM5KTCP_StartStream(&hEvent)`
5. `hEvent signaled?` (Branches to either `Y` or `N`)
   - If `hEvent signaled?` is `Y`, proceed to `ADAM5KTCP_ReadStreamData()`
   - If `hEvent signaled?` is `N`, return to the start of the loop
6. `quit?` (Branches to either `Y` or `N`)
   - If `quit?` is `Y`, proceed to `ADAM5KTCP_StopStream()`
   - If `quit?` is `N`, return to the start of the loop
7. `ADAM5KTCP_StopStream()`
8. `ADAM5KTCP_Close()`
*To receive alarm information from ADAM-5000/TCP(s)

1. ADAM5KTCP_Open()
2. ADAM5KTCP_Add5KTCPForStream()
3. hEvent=CreateEvent () (A Win32 API)
4. ADAM5KTCP_SetStreamAlarmState(ADAM5KTCP_ReceiveStreamWhenAlarm)
5. ADAM5KTCP_StartStream(&hEvent)
6. hEvent signaled?
   - N
   - Y
5. ADAM5KTCP_ReadStreamDatat()
7. I
8. ADAM5KTCP_ReadAlarmInfo()
9. Alarm info buffer is empty?
   - N
   - Y
8. quit?
   - N
   - Y
7. ADAM5KTCP_StopStream()
6. ADAM5KTCP_Close()
**To read coil values**

1. ADAM5KTCP_Open()
2. ADAM5KTCP_Connect()
3. ADAM5KTCP_ReadCoil()
4. ADAM5KTCP_Disconnect()
5. ADAM5KTCP_Close()

**To write value to coil**

1. ADAM5KTCP_Open()
2. ADAM5KTCP_Connect()
3. ADAM5KTCP_WriteCoil()
4. ADAM5KTCP_Disconnect()
5. ADAM5KTCP_Close()
* To read holding register value

```
ADAM5KTCP_Open()
ADAM5KTCP_Connect()
ADAM5KTCP_ReadReg()
ADAM5KTCP_Disconnect()
ADAM5KTCP_Close()
```

* To write value to holding register

```
ADAM5KTCP_Open()
ADAM5KTCP_Connect()
ADAM5KTCP_WriteReg()
ADAM5KTCP_Disconnect()
ADAM5KTCP_Close()
```
6-2-3 Function Descriptions

ADAM5KTCP_Open

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

Syntax: int ADAM5KTCP_Open(void);

Parameters: void

Return: Please refer to Chapter6-2-4 “Return Codes” for more detail information
**ADAM5KTCP_Close**

**Description:** Terminates using the “adam5ktcp.dll”.

**Syntax:**
```c
void ADAM5KTCP_Close(void);
```

**Parameters:**
void

**Return:**
void
**ADAM5KTCP_Connect**

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

**Syntax:**
```
int ADAM5KTCP_Connect(char szIP[], unsigned short port, int iConnectionTimeout, int iSendTimeout, int iReceiveTimeout);
```

**Parameter:**
- `szIP[in]`: the IP Address of the ADAM-5000/TCP that to be connected
- `port[in]`: the TCP/IP connection port used with Modbus/TCP, 502 default
- `iConnectionTimeout[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:** Please refer to Chapter 6-2-4 “Return Codes” for more detail information
ADAM5KTCP_Disconnect

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

Syntax: void ADAM5KTCP_Disconnect(void);

Parameter: void

Return: Please refer to Chapter 6-2-4 “Return Codes” for more detail information
DAM5KTCP_GetDLLVersion

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

Syntax: int ADAM5KTCP_GetDLLVersion(void);

Parameter: 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[]);
```

**Parameter:**
- `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 holdings register to be read
- `wData[out]`: a unsigned 16 bits array that stored the read holding register

**Return:**
Please 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 parameters.

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

**Parameter:**
- `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 holdings register to be written
- `wData[out]`: a unsigned 16 bits array that stored the value write to holding value

**Return:** Please refer to Chapter 6-2-4 “Return Codes” for more detail information
ADAM5KTCP_ReadCoil

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

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

Parameter:
- 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]: a 8 bit array that stored the read coil

Return: Please refer to Chapter 6-2-4 “Return Codes” for more detail information
**Chapter 6  Planning Your Application Program**

**ADAM5KTCP_WriteCoil**

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

**Syntax:**

```c
int ADAM5KTCP_WriteCoil(char szIP[], WORD wID, WORD wStartAddress, WORD wCount, BYTE byData[]);
```

**Parameter:**

- **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 coil

**Return:** Please refer to Chapter 6-2-4 “Return Codes” for more detail information
ADAM5KTCP_SendReceive5KTCP_Cmd

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

Syntax: int ADAM5KTCP_SendReceive5KTCP_Cmd(char szIP[], char szSendToTCP[], char szReceiveFromTCP[], char szModbusSend[], char szModbusReceive[]);

Parameter:
- 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: Please refer to Chapter 6-2-4 “Return Codes” for more detail information
ADAM5KTCP_Add5KTCPForStream

Description: Assign 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: Please refer to Chapter 6-2-4 “Return Codes” for more detail information
ADAM5KTCP_ReadStreamData

Description: Receive stream data that comes from the specific ADAM-5000/TCP

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

Parameters:
- szIP[in]: to specify the IP Address for a user to receive the stream data
- *pStreamData[out]: the stream data stored in _StreamData structure
  Please refer to Chapter 6-2-5 “Data Structure” for more detail information about _StreamData structure.

Return: Please 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
Please refer to Chapter 6-2-5 “Data Structure” for more detail information about _AlarmInfo structure.

Return: Please 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. Please refer to ADAM5KTCP_SetStream AlarmState for more detail information.

Return: Please 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 StreamIgnoreAlarm: 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_ReadAlertInfo()” function.

Return: Please 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]: indicating which ADAM-5000/TCP cause signaling the 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]: indicating the working thread status in ADAM5KTCP.DLL

*iTotalStream[out]: reserved

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

Return: Please 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:**
```c
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:**
Please 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: Please 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.

<table>
<thead>
<tr>
<th>Function Library</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAM5KTCP_NoError</td>
<td>0</td>
</tr>
<tr>
<td>ADAM5KTCP_StartupFailure</td>
<td>-1</td>
</tr>
<tr>
<td>ADAM5KTCP_SocketFailure</td>
<td>-2</td>
</tr>
<tr>
<td>ADAM5KTCP_UdpSocketFailure</td>
<td>-3</td>
</tr>
<tr>
<td>ADAM5KTCP_SetTimeoutFailure</td>
<td>-4</td>
</tr>
<tr>
<td>ADAM5KTCP_SendFailure</td>
<td>-5</td>
</tr>
<tr>
<td>ADAM5KTCP_ReceiveFailure</td>
<td>-6</td>
</tr>
<tr>
<td>ADAM5KTCP_ReceiveMaxFailure</td>
<td>-7</td>
</tr>
<tr>
<td>ADAM5KTCP_CreateWsaEventFailure</td>
<td>-8</td>
</tr>
<tr>
<td>ADAM5KTCP_ReadStreamDataFailure</td>
<td>-9</td>
</tr>
<tr>
<td>ADAM5KTCP_InvalidIP</td>
<td>-10</td>
</tr>
<tr>
<td>ADAM5KTCP_ThisIPNotConnected</td>
<td>-11</td>
</tr>
<tr>
<td>ADAM5KTCP_AlarmInfoEmpty</td>
<td>-12</td>
</tr>
</tbody>
</table>
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 example 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:

![Command Structure Diagram]

Figure 6-1: Request Command Structure
And the response should be:

Figure 6-2: Response Comment Structure

6-3-2 Modbus Function Code Introduction

To fulfill the programming requirement, there is a series of function code standard for user’s reference...

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

Table 6-1: Response Comment Structure

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:
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:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
</tbody>
</table>
| Start Address High | Start Address Low | Requested Number of Input High | Requested Number of Input Low | ...

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:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
<tr>
<td>Requested Number of Input Low</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
<tr>
<td>Start Address High</td>
</tr>
</tbody>
</table>
| Requested Number of Input Low | ...

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:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Station Address</strong></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Station Address</strong></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Station Address</strong></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
</tbody>
</table>

**Function Code 06**
Presets integer value into a single register.
Request message format for function code 06:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
</tbody>
</table>
Response message format for function code 08:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>StationAddress</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
</tbody>
</table>

**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 22 21 20 --------------- 29 28

Response message format for function code 15:
The normal responses return the station address, function code, start address, and requested number of coil forced.

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Station Address</th>
<th>Function Code</th>
<th>Start Address High Byte</th>
<th>Start Address Low Byte</th>
<th>Requested Number of Register High Byte</th>
<th>Requested Number of Register Low Byte</th>
<th>Byte Count</th>
<th>Data</th>
</tr>
</thead>
</table>

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.

<table>
<thead>
<tr>
<th>Command Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station Address</td>
</tr>
</tbody>
</table>

Example: 01 10 00 09 00 02
6-4  **Apply with ASCII Command for ADAM-5000/TCP System**

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

<table>
<thead>
<tr>
<th>TCP Format</th>
<th>Modbus Format</th>
<th>ASCII Command</th>
</tr>
</thead>
</table>

![Figure 6-3: ASCII Command Structure in ADAM-5000/TCP](image)

6-4-1  **Syntax 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>
<thead>
<tr>
<th>Command Syntax</th>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%aannccff</td>
<td>Configuration</td>
<td>Set the baudrate and checksum status for a specified ADAM-5000 system</td>
</tr>
<tr>
<td>$aa2</td>
<td>Configuration Status</td>
<td>Returns the configuration status for a specified ADAM-5000 system</td>
</tr>
<tr>
<td>$aaM</td>
<td>Read Module Name</td>
<td>Returns the module name from a specified ADAM-5000/TCP system</td>
</tr>
<tr>
<td>$aaF</td>
<td>Read Firmware Version</td>
<td>Returns the firmware version code from a specified ADAM-5000/TCP system</td>
</tr>
<tr>
<td>$aaT</td>
<td>Read I/O Type</td>
<td>Returns the I/O model number of all slots for a specified ADAM-5000/TCP system</td>
</tr>
</tbody>
</table>

*Table 6-2: CPU Command Set Table*
%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 hexadecimal Modbus network address of the ADAM-5000/TCP system.
mal Modbus network address of an ADAM-5000/TCP system.

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

Example
command: \%01000A40(c)
response: !01(c)
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>
<thead>
<tr>
<th>Baud Rate Code</th>
<th>Baud Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>03h</td>
<td>1200 bps</td>
</tr>
<tr>
<td>04h</td>
<td>2400 bps</td>
</tr>
<tr>
<td>05h</td>
<td>4800 bps</td>
</tr>
<tr>
<td>06h</td>
<td>9600 bps</td>
</tr>
<tr>
<td>07h</td>
<td>19.2 Kbps</td>
</tr>
<tr>
<td>08h</td>
<td>38.4 Kbps</td>
</tr>
<tr>
<td>09h</td>
<td>57.6 Kbps</td>
</tr>
<tr>
<td>0Ah</td>
<td>115.2 Kbps</td>
</tr>
</tbody>
</table>

Table 6-3 Baud rate codes
$aa2

Name  Configuration Status

Description  Returns the configuration status for a specified system module.

Syntax  $aa2(cr)

$ is a delimiter character.

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

2 is the Configuration Status command.

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

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

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

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

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

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

cc represents the baud rate code.

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

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

(See also the %aannccff configuration command)
Example

command: $012(cr)
response: !010600(cr)

The command requests the ADAM-5000/TCP system at address 01h to send its configuration status. The ADAM-5000 system at address 01h responds with a baud rate of 9600 bps and with no checksum function or checksum generation.
$aaM

Name       Read Module Name
Description Returns the module name from a specified ADAM-5000/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)
```

$s$ 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.

ADAM-5013 RTD Input Command Set

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$aaSiArrff</td>
<td>RTD Configuration</td>
<td>Sets slot index, input range, data format and integration time for a specified RTD input module in a specified system</td>
</tr>
<tr>
<td>$aaSiB</td>
<td>RTD Configuration Status</td>
<td>Returns the configuration parameters for a specified RTD input module in a specified system</td>
</tr>
<tr>
<td>$aaSi</td>
<td>All RTD Data In</td>
<td>Returns the input values of all channels of a specified RTD input module of a specified system in engineering units</td>
</tr>
<tr>
<td>$aaSiCj</td>
<td>Specified RTD Data In</td>
<td>Returns the input value of a specified channel for a specified RTD input module of a specified system in engineering units</td>
</tr>
<tr>
<td>Command Syntax</td>
<td>Command Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$aaSiER</td>
<td>Initialize EEPROM Data</td>
<td>Initializes all EEPROM data in a specified RTD input module to their default values</td>
</tr>
<tr>
<td>$aaSi5mm</td>
<td>Enable/Disable Channels for Multiplexing</td>
<td>Enables/disables multiplexing simultaneously for separate channels of the specified input module</td>
</tr>
<tr>
<td>$aaSi6</td>
<td>Read Channels Status</td>
<td>Asks a specified input module to return the status of all channels</td>
</tr>
<tr>
<td>$aaSi0</td>
<td>RTD Span Calibration</td>
<td>Calibrates a specified RTD input module to correct for gain errors</td>
</tr>
<tr>
<td>$aaSi1</td>
<td>RTD Zero Calibration</td>
<td>Calibrates a specified RTD input module to correct for offset errors</td>
</tr>
<tr>
<td>$aaSi2</td>
<td>RTD Self Calibration</td>
<td>Causes a specified RTD input module of a specified system to do a self calibration.</td>
</tr>
</tbody>
</table>

*Table 6-4: ADAM-5013 RTD Input command Set Table*
### $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**

```plaintext
$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 in slot 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)
**Name**
Enable/Disable Channels for multiplexing

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

**Syntax**

\$aaSi5mm\(\text{cr}\)

$ 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.

\(\text{cr}\) is the terminating character, carriage return (0Dh)

**Response**

!aa\(\text{cr}\) if the command is valid.

?aa\(\text{cr}\) 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.

\(\text{cr}\) is the terminating character, carriage return (0Dh)

**Example**

command: $01S1501\(\text{cr}\)

response: !01\(\text{cr}\)

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 span calibration command, 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 Table

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

**$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**

- **Date Format**  
  - 00: Engineering units  
- **Integration Time**  
  - 0.50 ms (Operation under 60 Hz power)  
  - 1.80 ms (Operation under 60 Hz power)
**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: **S01S3A0000**(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.
Chapter 6  Planning Your Application Program

$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).
**Planning Your Application Program**

**$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.
?aacr) 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.
mn 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:

- ?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

<table>
<thead>
<tr>
<th>Name</th>
<th>Span Calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Calibrates a specified analog input module to correct for gain errors</td>
</tr>
<tr>
<td>Syntax</td>
<td>$aaSi0(cr)</td>
</tr>
</tbody>
</table>

$ 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)

<table>
<thead>
<tr>
<th>Response</th>
<th>!aa(cr) if the command is valid.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>?aa(cr) if an invalid operation was entered.</td>
</tr>
</tbody>
</table>

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 ad-

dress does not exist.

! delimiter character indicating a valid command was
received.

? delimiter character indicating the command was in-
valid.

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

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

Note:  In order to successfully calibrate an analog input module’s in-
put range, a proper calibration input signal should be connected
to the analog input module before and during the calibration
process.
**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.

>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 analog 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.
ADAM-5017H Analog Input Command Set

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$aaSiCjArrFF</td>
<td>Set Input Range</td>
<td>Sets input range for a specified channel of an analog input module in a specified system</td>
</tr>
<tr>
<td>$aaSiCjB</td>
<td>Read Input Range</td>
<td>Returns the input range for a specified channel of a specified analog input module in a specified system</td>
</tr>
<tr>
<td>$aaSiAFFff</td>
<td>Set Data Format</td>
<td>Sets data format in engineering units or two's complement for a specified analog input module in a specified system</td>
</tr>
<tr>
<td>$aaSiB</td>
<td>Read Data Format</td>
<td>Returns the data format for a specified analog input module in a specified system</td>
</tr>
<tr>
<td>$aaSi5mm</td>
<td>Enable/Disable Channels for Multiplexing</td>
<td>Enables/Disables multiplexing for separate channels of the specified input module</td>
</tr>
<tr>
<td>$aaSi6</td>
<td>Read Channels Status</td>
<td>Asks the specified input module to return the status of all channels</td>
</tr>
</tbody>
</table>
### Table 6-6: ADAM-5017H Analog Input command Set Table

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#aaSi</td>
<td>All Analog Data In</td>
<td>Returns the input value of all channels for a specified analog input module of a specified system in currently configured data format</td>
</tr>
<tr>
<td>#aaSiCj</td>
<td>Specified Analog Data In</td>
<td>Returns the input value of a specified channel of a specified analog input module of a specified system in currently configured data format</td>
</tr>
<tr>
<td>$aaSiER</td>
<td>Initialize EEPROM Data</td>
<td>Initializes all EEPROM data in a specified analog input module to their default values.</td>
</tr>
<tr>
<td>$aaSi0</td>
<td>Span Calibration</td>
<td>Calibrates a specified analog input module to correct for gain errors</td>
</tr>
<tr>
<td>$aaSi1</td>
<td>Zero Calibration</td>
<td>Calibrates a specified analog input module to correct for offset errors</td>
</tr>
</tbody>
</table>

**Note:** The command sets “$aasi5mm, $aasi6, $aasi0, $aasi1” for ADAM-5017H are the same with ADAM-5017. Please refer the preceding pages to learn the detail.
### Planning Your Application Program

**Chapter 6**

#### $aaSiCjArrFF$

<table>
<thead>
<tr>
<th>Name</th>
<th>Set Input Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Sets the input range for a specified channel of a specified analog input module in a specified system.</td>
</tr>
</tbody>
</table>

**Syntax**

$\$aSiCjArrFF$

- $\$$ 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 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 an ADAM-5017H 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 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 format. The response indicates that the command has been received as a valid command.
$aaSiCjB

Name                  Read Input Range
Description           Returns the input range in engineering units for a
                       specified channel of a specified analog input module
                       in a specified system.
Syntax                $aaSiCjB
                       $ is a delimiter character.
                       aa (range 00-FF) represents the 2-character hexadecimal
                       address of the ADAM-5000/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 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 ad-
                       dress does not exist.
                       ! delimiter character indicating a valid command was
                       received.
                       ? delimiter character indicating the command was in-
                       valid.
                       aa (range 00-FF) represents the 2-character hexadecimal
                       address of an ADAM-5000/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: S01S3C1B(cr)
                       response: !010b00(cr)
                       Channel 1 of the ADAM-5017H 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Data Format</td>
<td>Sets the data format in engineering units or in two’s complement format for a specified analog input module in a specified system.</td>
<td>$aaSiAFFff</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>!aa(cr)</td>
<td>if the command is valid.</td>
</tr>
<tr>
<td>?aa(cr)</td>
<td>if an invalid operation was entered.</td>
</tr>
<tr>
<td>!</td>
<td>delimiter character indicating a valid command was received.</td>
</tr>
<tr>
<td>?</td>
<td>delimiter character indicating the command was invalid.</td>
</tr>
</tbody>
</table>

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

**Example**

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

**response:** !01(cr)

The data format of the ADAM-5017H 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.
### Planning Your Application Program

#### $aaSiB

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th>Read Data Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Returns the data format for a specified analog input module in a specified system.</td>
</tr>
<tr>
<td><strong>Syntax</strong></td>
<td>$aaSiB</td>
</tr>
<tr>
<td></td>
<td>$ is a delimiter character.</td>
</tr>
<tr>
<td></td>
<td>aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000/TCP system you want to interrogate.</td>
</tr>
<tr>
<td></td>
<td>Si identifies the I/O slot of the ADAM-5000/TCP system containing the ADAM-5017H module you want to interrogate.</td>
</tr>
<tr>
<td></td>
<td>B represents the read data format command.</td>
</tr>
<tr>
<td></td>
<td>(cr) is the terminating character, carriage return (0Dh).</td>
</tr>
</tbody>
</table>

#### Response

<table>
<thead>
<tr>
<th>Response</th>
<th>!aaFFff(cr) if the command is valid.</th>
</tr>
</thead>
<tbody>
<tr>
<td>?aa(cr)</td>
<td>if an invalid operation was entered.</td>
</tr>
<tr>
<td>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</td>
<td></td>
</tr>
<tr>
<td>! delimiter character indicating a valid command was received.</td>
<td></td>
</tr>
<tr>
<td>? delimiter character indicating the command was invalid.</td>
<td></td>
</tr>
<tr>
<td>aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000/TCP system.</td>
<td></td>
</tr>
<tr>
<td>ff represents the 2-character hexadecimal code of the data format. 00 is for engineering unit format. 02 is for two’s complement format.</td>
<td></td>
</tr>
<tr>
<td>(cr) is the terminating character, carriage return (0Dh).</td>
<td></td>
</tr>
</tbody>
</table>

#### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>$01S3B(cr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>!01FF00(cr)</td>
</tr>
<tr>
<td>The ADAM-5017H module in slot 3 of the ADAM-5000/TCP system at address 01h responds that it is configured for engineering unit data format.</td>
<td></td>
</tr>
</tbody>
</table>
#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  

# 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  

if the command is valid. (Engineering Unit Data Format)

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 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 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.
## Analog Input Alarm Command Set

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

**Table 6-7 Analog Input alarm command set table**

Note: This command set applies to the ADAM-5013, ADAM-5017, ADAM-5017H 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**  
$aaSiCjAh(cr)$

- $s$ 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**  
$aaSiCjAhEs(cr)$

- **S**: 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).
- **AhEs**: is the Set Alarm Mode command.
- **h**: indicates alarm type and can have the value H = High alarm, L = Low alarm
- **s**: indicates alarm enable/disable and can have the value E = Enable, D = Disable
- **(cr)**: represents terminating character, carriage return (0Dh)

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

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

- **! delimiter character indicating a valid command was received.**
- **aa**: represents the 2-character hexadecimal address of the corresponding ADAM-5000/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)

S 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 connects to the digital output at point 0 of slot 1 in the same ADAM-5000/TCP system.
Chapter 6  Planning Your Application Program

$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.
## Analog Input Alarm Command Set

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$aaSiCjAh</td>
<td>Read Alarm Mode</td>
<td>Returns the alarm mode for the specified channel.</td>
</tr>
<tr>
<td>$aaSiCjAhEs</td>
<td>Enable/Disable Alarm</td>
<td>Enables or Disables the High/Low alarm of the specified channel.</td>
</tr>
<tr>
<td>$aaSiCjCh</td>
<td>Clear Latch Alarm</td>
<td>Resets a latched alarm</td>
</tr>
<tr>
<td>$aaSiCjAhCSk-Cn</td>
<td>Set Alarm Connection</td>
<td>Connects the High/Low alarm of a specified input channel to a specified digital output channel.</td>
</tr>
<tr>
<td>$aaSiCjRhC</td>
<td>Read Alarm Connection</td>
<td>Returns the alarm limit output connection of a specified input channel.</td>
</tr>
<tr>
<td>$aaSiCjAhU(data)</td>
<td>Set Alarm Limit</td>
<td>Sets the High/Low alarm limit value for the specified input channel.</td>
</tr>
<tr>
<td>$aaSiCjRhU</td>
<td>Read Alarm Limit</td>
<td>Returns the High/Low alarm limit value for the specified input channel.</td>
</tr>
<tr>
<td>$aaSiCjS</td>
<td>Read Alarm Status</td>
<td>Reads whether an alarm occurred for a specified input channel.</td>
</tr>
</tbody>
</table>

### Table 6-8 Analog Input alarm command set table

Note: This command set applies to the ADAM-5013, ADAM-5017, ADAM-5017H 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 $H = \text{High alarm}, \ L = \text{Low alarm}$

$s$ indicates alarm mode and can have the value $M = \text{Momentary mode}, \ L = \text{Latching mode}$

$(cr)$ 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.
Planning Your Application Program

$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 ad-
dress 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).

AhEs is the Set Alarm Mode command.

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

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

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

Response
!aa(cr) if the command was valid

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

! delimiter character indicating a valid command was received.

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

$aaSiCjAhCSkCn (cr)

$s$ 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**  
$aaSiCjRhC$\text{(cr)}$

$S$ 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**  
$!aaSkCn$\text{(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(crl)
response: !0101(crl)

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

*Table 6-9: Analog Output command Set Table*
### Planning Your Application Program

**$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. Bits 2,3,4,5 represent slew rate. The layout of the 8-bit parameter is shown in Figure 6-4. The other bits are not used and are set to 0.
- (cr) is the terminating character, carriage return (0Dh)

**Figure 6-5: 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 addressed to perform any other actions.
### Planning Your Application Program

#### $aaSiCjB$

<table>
<thead>
<tr>
<th>Name</th>
<th>Configuration Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Returns the configuration parameters of a specified channel in a specified analog output module of a specified system.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
<th>$aaSiCjB(\text{cr})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$ is a delimiter character.</td>
<td></td>
</tr>
<tr>
<td>$aa$ (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000/TCP system you want to interrogate.</td>
<td></td>
</tr>
<tr>
<td>$\text{SiCj}$ identifies the I/O slot $i$ ($i: 0$ to $7$) and the channel $j$ ($j: 0$ to $3$) you want to read.</td>
<td></td>
</tr>
<tr>
<td>$B$ is configuration status command.</td>
<td></td>
</tr>
<tr>
<td>(\text{cr}) is the terminating character, carriage return (0Dh)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>!aarrff(\text{cr}) if the command is valid.</th>
</tr>
</thead>
<tbody>
<tr>
<td>?aa(\text{cr}) if an invalid operation was entered.</td>
<td></td>
</tr>
<tr>
<td>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</td>
<td></td>
</tr>
<tr>
<td>! delimiter character indicating a valid command was received.</td>
<td></td>
</tr>
<tr>
<td>? delimiter character indicating the command was invalid.</td>
<td></td>
</tr>
<tr>
<td>$aa$ (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000/TCP system.</td>
<td></td>
</tr>
<tr>
<td>$rr$ represents the 2-character hexadecimal code of the output range.</td>
<td></td>
</tr>
<tr>
<td>$ff$ 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 $\text{$aSiCjArrff}$)</td>
<td></td>
</tr>
<tr>
<td>(\text{cr}) is the terminating character, carriage return (0Dh)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>command: $01S1C1B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>response: !013210</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
#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: #01S1C115.000(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 15 mA from it’s channel 1. The module should be an analog output module with it’s channel 1 configured for a range of 0-20 mA or 4-20 mA. If it is an analog output module configured for the range 0-10 V, it’s output value will be 10 V and the response will be ?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.
Chapter 6  Planning Your Application Program

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)
$s 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.
### Planning Your Application Program

<table>
<thead>
<tr>
<th>Command</th>
<th>Name</th>
<th>Description</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>$aaSiCj1</td>
<td>20 mA Calibration</td>
<td>Directs the specified channel to store parameters following a calibration for 20 mA output</td>
<td>$aaSiCj1(cr)</td>
</tr>
</tbody>
</table>

$ 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)

<table>
<thead>
<tr>
<th>Response</th>
<th>!aa(cr) if the command is valid.</th>
<th>?aa(cr) if an invalid operation was entered.</th>
<th>! delimiter character indicating a valid command was received.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</td>
<td>? delimiter character indicating the command was invalid.</td>
<td>! delimiter character indicating a valid command was received.</td>
</tr>
<tr>
<td></td>
<td>$aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000/TCP system.</td>
<td>(cr) is the terminating character, carriage return (0Dh)</td>
<td>! delimiter character indicating a valid command was received.</td>
</tr>
</tbody>
</table>

**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 hexadeci-
            mal address of the ADAM-5000/TCP system.
            SiCj identifies the I/O slot i (i : 0 to 7) and the chan-
            nel 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 in-
            crease 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 ad-

dress does not exist.

! delimiter character indicating a valid command was

received.

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

mal 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.
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).
## Digital Input/Output Command Set

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$aaSi6</td>
<td>Digital Data In</td>
<td>&quot;Returns the values of digital I/O channels for a specified module&quot;</td>
</tr>
<tr>
<td>#aaSiBB(data)</td>
<td>Digital Data Out</td>
<td>&quot;Sets output values of a single digital output channel or of all digital output channels simultaneously for a specified module.&quot;</td>
</tr>
<tr>
<td>$aaSiM</td>
<td>&quot;Read Channel Masking Status&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;Asks the specified module to return the masking status of all digital output channels.&quot;</td>
<td></td>
</tr>
</tbody>
</table>

### $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)
Chapter 6  Planning Your Application Program

!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.
Chapter 6  Planning Your Application Program

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. Channels 0, 1, 3, 4 and 5 will be set to ON while channel 2 is set to OFF.

Bits 6 and 7 are not used and always default to 0.

---

Note: If any channel of the digital output module is configured as the output for an analog input alarm, it cannot be reconfigured via digital output commands. Channels used for analog input alarms always have a higher priority.
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 sta-
tus of digital output channels
Syntax $aaSiM(cr)
$ is a delimiter character.
aa (range 00-FF) represents the 2-character hexadecli-
mal 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 ad-
dress does not exist.
! delimiter character indicating a valid command was
received.
? delimiter character indicating the command was in-
valid.
aa (range 00-FF) represents the 2-character hexadeci-
mal address of an ADAM-5000/TCP system that is
responding.
(data) is the hexadecimal value representing the sta-
tus 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.
Example

(cr) is the terminating character, carriage return (0Dh)
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.
# ADAM-5080 Counter/Frequency Command Set

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$aaT</td>
<td>Read Module Name</td>
<td>Returns the module name from a specified ADAM-5000 system.</td>
</tr>
<tr>
<td>$aaF</td>
<td>Read Firmware Version</td>
<td>Returns the firmware version code from a specified ADAM-5000 system.</td>
</tr>
<tr>
<td>$aaSiArrff</td>
<td>Set Configuration</td>
<td>Set slot index and counter mode</td>
</tr>
<tr>
<td>$aaSiB</td>
<td>Read Configuration</td>
<td>The command requests the Configuration of slot</td>
</tr>
<tr>
<td>#aaSi</td>
<td>Read All Channel Counter (Frequency) Data</td>
<td>Returns the input value of all channels for the specified input module for a specified system in engineering unit only.</td>
</tr>
<tr>
<td>#aaSiCj</td>
<td>Read One Channel Counter (Frequency) Data</td>
<td>The command will return the input value from one of the four channels of a specified module.</td>
</tr>
<tr>
<td>$aaSiØ(data)</td>
<td>Set Digital filter Scale</td>
<td>Set the filter seconds to start to measure the input signal.</td>
</tr>
<tr>
<td>$aaSiØ</td>
<td>Read Digital filter scale</td>
<td>Read the filter seconds to start to measure the input signal.</td>
</tr>
<tr>
<td>$aaSiCj5s</td>
<td>Set Counter Start/Stop</td>
<td>Request the addressed counter/frequency module to start or stop the counting.</td>
</tr>
</tbody>
</table>
### Table 6-10: Counter/Frequency Command Set Table

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$aaSiCj6</td>
<td>Clear Counter</td>
<td>Clear the counters of the specified counter/frequency module</td>
</tr>
<tr>
<td>$aaSi7</td>
<td>Read Overflow Flag</td>
<td>The command requests the addressed module to return the status of the overflow flag of counter.</td>
</tr>
<tr>
<td>@aaSiCjP(data)</td>
<td>Set Initial Counter Value</td>
<td>Set initial counter value for counter of the specified counter module.</td>
</tr>
<tr>
<td>@aaSiCjG</td>
<td>Read Counter Initial Value</td>
<td>Read initial of the specified counter module.</td>
</tr>
<tr>
<td>$aaSiCjAhEs</td>
<td>Set Alarm Disable/Latch</td>
<td>The addressed counter module is instructed to set alarm disable or latch.</td>
</tr>
<tr>
<td>$aaSiCjAh</td>
<td>Read Alarm Disable/Latch</td>
<td>Returns the alarm mode for the specified channel.</td>
</tr>
<tr>
<td>$aaSiCjCh</td>
<td>Clear Alarm Status</td>
<td>Returns the alarm status to normal.</td>
</tr>
<tr>
<td>$aaSiCjAhCSkCn</td>
<td>Set Alarm Connection</td>
<td>Connects the High/Low alarm of the specified input channel to the specified digital output in the addressed ADAM-5000 system</td>
</tr>
<tr>
<td>$aaSiCjRhC</td>
<td>Read Alarm Connection</td>
<td>Returns the High/Low alarm limit output connection of a specified input channel in the addressed ADAM-5000 system</td>
</tr>
<tr>
<td>$aaSiCjAhU(data)</td>
<td>Set Alarm Limit</td>
<td>Sets the High/Low alarm limit value for the specified input channel of a specified ADAM-5000 system.</td>
</tr>
<tr>
<td>$aaSiCjRhU</td>
<td>Read Alarm Limit</td>
<td>Returns the High/Low alarm limit value for the specified input channel in the addressed ADAM-5000 system.</td>
</tr>
<tr>
<td>$aaSiCjS</td>
<td>Read Alarm Status</td>
<td>Reads whether an alarm occurred for the specified input channel in the specified ADAM-5000 system.</td>
</tr>
</tbody>
</table>
$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.
**Chapter 6  Planning Your Application Program**

### $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

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th>Read Configuration.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The command requests the Configuration of slot</td>
</tr>
<tr>
<td><strong>Syntax</strong></td>
<td>$aaSiB(cr)</td>
</tr>
<tr>
<td>$ is a delimiter character.</td>
<td></td>
</tr>
<tr>
<td>aa (range 00-FF) represents the 2-character hexadecimal address of the ADAM-5000/TCP system you want to interrogate.</td>
<td></td>
</tr>
<tr>
<td>Si identifies the desired slot i</td>
<td></td>
</tr>
<tr>
<td>B represents the configuration status command</td>
<td></td>
</tr>
<tr>
<td>(cr) is the terminating character, carriage return (0Dh).</td>
<td></td>
</tr>
<tr>
<td><strong>Response</strong></td>
<td>!aarrff(cr) if the command is valid.</td>
</tr>
<tr>
<td>?aa(cr) if an invalid operation was entered.</td>
<td></td>
</tr>
<tr>
<td>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</td>
<td></td>
</tr>
<tr>
<td>! delimiter character indicating a valid command is received.</td>
<td></td>
</tr>
<tr>
<td>? delimiter character indicating the command is invalid.</td>
<td></td>
</tr>
<tr>
<td>aa (range 00-FF) represents the 2-character hexadecimal address of an ADAM-5000/TCP system.</td>
<td></td>
</tr>
<tr>
<td>rr=00 represents Bi-direction counter mode.</td>
<td></td>
</tr>
<tr>
<td>rr=01 represents UP/DOWN counter mode.</td>
<td></td>
</tr>
<tr>
<td>rr=02 represents Frequency mode.</td>
<td></td>
</tr>
<tr>
<td>ff indicates which format is</td>
<td></td>
</tr>
<tr>
<td>ff=00 represents the engineer format.</td>
<td></td>
</tr>
<tr>
<td>ff=02 represents the hexadecimal format.</td>
<td></td>
</tr>
<tr>
<td>(cr) is the terminating character, carriage return (0Dh).</td>
<td></td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>command: $01S3B(cr)</td>
</tr>
<tr>
<td>response: !010100(cr)</td>
<td></td>
</tr>
<tr>
<td>The ADAM-5080 in Slot 3 of ADAM-5000/TCP system at address 01h responds that it is configured in UP/DOWN counter mode and for engineering unit data format.</td>
<td></td>
</tr>
</tbody>
</table>
#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:

>00000987000000649000000762000000011600(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 example of the value above, the actual frequency should be:

\[
\text{actual frequency} = \frac{98700}{100} = 987
\]

If the response is:

\[
>0000F1000002000000031000000DD400\text{(cr)}
\]

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:

\[
\text{F100 (hexdecimal)} = 24100 \text{ (decimal)}
\]

\[
\text{actual frequency} = \frac{24100}{100} = 241
\]
#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 calibrated.

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.

?aaa(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.

?aaa(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 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.
- **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.  
ffffffffff is the number of overflow for a specified channel.  
(ffffffffff 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.

AhEs is the command for setting Alarm Disable/Latch
Mode command.

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

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

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

Response       !aa(cr) if the command was valid

There is no response if the system detects a syntax
error or communication error or if the specified ad-
dress 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 that it is latched.
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 done 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.
### Planning Your Application Program

<table>
<thead>
<tr>
<th>Name</th>
<th>Set Alarm Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Set the High/Low alarm limit value for the specified input channel of a specified ADAM-5000/TCP system.</td>
</tr>
<tr>
<td>Syntax</td>
<td>$aaSiCjAhU(data)(cr)$</td>
</tr>
</tbody>
</table>

$ 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)

<table>
<thead>
<tr>
<th>Response</th>
<th>!aa(cr) if the command was valid</th>
</tr>
</thead>
</table>

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: $01S0C1AHU0000000020(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: $01$0C1S
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-
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

<table>
<thead>
<tr>
<th>Function Area</th>
<th>Equipment</th>
<th>Input or Output</th>
<th>I/O Module Type</th>
<th>I/O Module Product No.</th>
<th>Voltage of Range</th>
<th>Current of Range</th>
<th>Special Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.
### Design Worksheets  Appendix A

#### Table A-2: Summary Required Modules

<table>
<thead>
<tr>
<th>&lt;A&gt; ADAM-5000/TCP IP Address</th>
<th>&lt;B&gt; I/O Module Product No.</th>
<th>&lt;C&gt; I/O Points per Module</th>
<th>&lt;D&gt; Total I/O Points Required</th>
<th>&lt;E&gt; I/O Module Required</th>
<th>&lt;F&gt; Spare I/O Modules</th>
<th>&lt;G&gt; Total I/O Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 4: Implement the Modbus address into the I/O table.

<table>
<thead>
<tr>
<th>ADAM-5000/TCP IP Address</th>
<th>I/O Module Slot No.</th>
<th>I/O Type</th>
<th>Channel Number</th>
<th>I/O Address</th>
<th>Tag Name</th>
<th>Equipment &amp; Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 A  Design Worksheets
Appendix B

Data Formats and I/O Ranges
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:

<table>
<thead>
<tr>
<th>Input Range</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>±15 mV, ±50 mV</td>
<td>1 µV (three decimal places)</td>
</tr>
<tr>
<td>100 mV, 150 mV, 500 mV</td>
<td>10 µV (two decimal places)</td>
</tr>
<tr>
<td>±1 V, ±2.5 V, ±5 V</td>
<td>100 µV (four decimal places)</td>
</tr>
<tr>
<td>±10 V</td>
<td>1 mV (three decimal places)</td>
</tr>
<tr>
<td>±20 mA</td>
<td>1 µA (three decimal places)</td>
</tr>
<tr>
<td>Type J and T thermocouple</td>
<td>0.01º C (two decimal places)</td>
</tr>
<tr>
<td>Type K, E, R, S, and B</td>
<td>0.1º C (one decimal places)</td>
</tr>
<tr>
<td>thermocouple</td>
<td></td>
</tr>
</tbody>
</table>
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\(\text{cr}\)

Example 2
The input value is 305.5°C. The analog input module is configured for a Type J thermocouple whose range is 0°C to 760°C. The response to the Analog Data In command is:

+305.50\(\text{cr}\)

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\(\text{cr}\)
### B.2 Analog Input Ranges - ADAM-5017

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Range</th>
<th>Engineering Units</th>
<th>% of FS</th>
<th>0.00 ± 0.00</th>
<th>-FS</th>
<th>Displayed Resolution</th>
<th>Actual Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>08h</td>
<td>±10 V</td>
<td>Engineering Units</td>
<td>+10.000</td>
<td>±0.0000</td>
<td>-10.00</td>
<td>1 mV</td>
<td>Reading 1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FS</td>
<td>+100.00</td>
<td>±0.0000</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FFF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>±5 V</td>
<td>Engineering Units</td>
<td>+5.0000</td>
<td>±0.0000</td>
<td>-5.0000</td>
<td>1000 µV</td>
<td>Reading 1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FS</td>
<td>+100.00</td>
<td>±0.0000</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FFF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>0Ah</td>
<td>±1 V</td>
<td>Engineering Units</td>
<td>+1.0000</td>
<td>±0.0000</td>
<td>-1.0000</td>
<td>1000 µV</td>
<td>Reading 1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FS</td>
<td>+100.00</td>
<td>±0.0000</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FFF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>0Bh</td>
<td>±500 mV</td>
<td>Engineering Units</td>
<td>±500.00</td>
<td>±0.0000</td>
<td>-500.00</td>
<td>10 µV</td>
<td>Reading 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FS</td>
<td>±100.00</td>
<td>±0.0000</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FFF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>0Ch</td>
<td>±150 mV</td>
<td>Engineering Units</td>
<td>±150.00</td>
<td>±0.0000</td>
<td>-150.00</td>
<td>10 µV</td>
<td>Reading 100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FS</td>
<td>±100.00</td>
<td>±0.0000</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FFF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>0Dh</td>
<td>±20 mA</td>
<td>Engineering Units</td>
<td>±20.00</td>
<td>±0.0000</td>
<td>-20.00</td>
<td>1 µV</td>
<td>Reading 1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FS</td>
<td>±100.00</td>
<td>±0.0000</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FFF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
</tbody>
</table>
## Appendix B

### B.3 Analog Input Ranges - ADAM-5018

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Range</th>
<th>Data Formats</th>
<th>+FS</th>
<th>Zero</th>
<th>-FS</th>
<th>Displayed Resolution</th>
<th>Actual Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>±15 mV</td>
<td>Engineering Units</td>
<td>+15000</td>
<td>±00000</td>
<td>-15000</td>
<td>1 µV</td>
<td>Reading/1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>±000.00</td>
<td>-100.00</td>
<td>0.001%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>FF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>00h</td>
<td>±50 mV</td>
<td>Engineering Units</td>
<td>+50000</td>
<td>±00000</td>
<td>-50000</td>
<td>1 µV</td>
<td>Reading/100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>±000.00</td>
<td>-100.00</td>
<td>0.001%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>FF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>02h</td>
<td>±100 mV</td>
<td>Engineering Units</td>
<td>+100000</td>
<td>±00000</td>
<td>-100000</td>
<td>10 µV</td>
<td>Reading/100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>±000.00</td>
<td>-100.00</td>
<td>0.001%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>FF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>03h</td>
<td>±500 mV</td>
<td>Engineering Units</td>
<td>+500000</td>
<td>±00000</td>
<td>-500000</td>
<td>100 µV</td>
<td>Reading/1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>±000.00</td>
<td>-100.00</td>
<td>0.001%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>FF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>04h</td>
<td>±1 V</td>
<td>Engineering Units</td>
<td>+1000000</td>
<td>±00000</td>
<td>-1000000</td>
<td>1000 µV</td>
<td>Reading/10000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>±000.00</td>
<td>-100.00</td>
<td>0.001%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>FF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>05h</td>
<td>±2.5 V</td>
<td>Engineering Units</td>
<td>+2500000</td>
<td>±00000</td>
<td>-2500000</td>
<td>1000 µV</td>
<td>Reading/10000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>±000.00</td>
<td>-100.00</td>
<td>0.001%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>FF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>06h</td>
<td>±20 mA</td>
<td>Engineering Units</td>
<td>±20000</td>
<td>±0000</td>
<td>±20000</td>
<td>1 µA</td>
<td>Reading/1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>±100.00</td>
<td>±000.00</td>
<td>±100.00</td>
<td>0.001%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>FF</td>
<td>0000</td>
<td>8000</td>
<td>1 LSB</td>
<td></td>
</tr>
<tr>
<td>07h</td>
<td>Not Used</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Appendix B  Data Formats and I/O Ranges

<table>
<thead>
<tr>
<th>Module</th>
<th>Range Code</th>
<th>Input Range Description</th>
<th>Data Formats</th>
<th>Minimum Specified Signal</th>
<th>Maximum Specified Signal</th>
<th>Displayed Resolution</th>
<th>Actual Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAM-5018</td>
<td>00h</td>
<td>Type J Thermocouple 0°C to 760°C</td>
<td>Engineering Units</td>
<td>+760.00</td>
<td>-760.00</td>
<td>0.1°C</td>
<td>Reading/10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FF</td>
<td>0000</td>
<td>1 LSB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>01h</td>
<td>Type K Thermocouple 0°C to 1370°C</td>
<td>Engineering Units</td>
<td>+1370.00</td>
<td>-1370.00</td>
<td>0.1°C</td>
<td>Reading/10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FF</td>
<td>0000</td>
<td>1 LSB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10h</td>
<td>Type T Thermocouple -100°C to 400°C</td>
<td>Engineering Units</td>
<td>+400.00</td>
<td>-400.00</td>
<td>0.1°C</td>
<td>Reading/10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FF</td>
<td>0000</td>
<td>1 LSB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11h</td>
<td>Type E Thermocouple 0°C to 1000°C</td>
<td>Engineering Units</td>
<td>+1000.00</td>
<td>-1000.00</td>
<td>0.1°C</td>
<td>Reading/10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FF</td>
<td>0000</td>
<td>1 LSB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12h</td>
<td>Type R Thermocouple 500°C to 1750°C</td>
<td>Engineering Units</td>
<td>+1750.00</td>
<td>-1750.00</td>
<td>0.1°C</td>
<td>Reading/10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FF</td>
<td>2482</td>
<td>1 LSB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13h</td>
<td>Type S Thermocouple 500°C to 1750°C</td>
<td>Engineering Units</td>
<td>+1750.00</td>
<td>-1750.00</td>
<td>0.1°C</td>
<td>Reading/10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FF</td>
<td>2482</td>
<td>1 LSB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14h</td>
<td>Type B Thermocouple 500°C to 1800°C</td>
<td>Engineering Units</td>
<td>+1800.00</td>
<td>-1800.00</td>
<td>0.1°C</td>
<td>Reading/10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of FSR</td>
<td>+100.00</td>
<td>-100.00</td>
<td>0.01%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two's Complement</td>
<td>7FF</td>
<td>2381</td>
<td>1 LSB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### B.4 Analog Input Ranges - ADAM-5017H

<table>
<thead>
<tr>
<th>Range Code</th>
<th>Input Range</th>
<th>Data Formats</th>
<th>+Full Scale</th>
<th>Zero</th>
<th>-Full Scale</th>
<th>Displayed Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>±10 V</td>
<td>Engineering</td>
<td>11</td>
<td>0</td>
<td>-11</td>
<td>2.7 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>EFFF</td>
<td>1</td>
</tr>
<tr>
<td>01h</td>
<td>0 ~ 10 V</td>
<td>Engineering</td>
<td>11</td>
<td>0</td>
<td>Don’t care</td>
<td>2.7 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>Don’t care</td>
<td>1</td>
</tr>
<tr>
<td>02h</td>
<td>±5 V</td>
<td>Engineering</td>
<td>5.5</td>
<td>0</td>
<td>-5.5</td>
<td>1.3 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>EFFF</td>
<td>1</td>
</tr>
<tr>
<td>03h</td>
<td>0 ~ 5 V</td>
<td>Engineering</td>
<td>5.5</td>
<td>0</td>
<td>Don’t care</td>
<td>1.3 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>Don’t care</td>
<td>1</td>
</tr>
<tr>
<td>04h</td>
<td>±2.5 V</td>
<td>Engineering</td>
<td>2.75</td>
<td>0</td>
<td>-2.75</td>
<td>0.67 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>EFFF</td>
<td>1</td>
</tr>
<tr>
<td>05h</td>
<td>0 ~ 2.5 V</td>
<td>Engineering</td>
<td>2.75</td>
<td>0</td>
<td>Don’t care</td>
<td>0.67 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>Don’t care</td>
<td>1</td>
</tr>
<tr>
<td>06h</td>
<td>±1 V</td>
<td>Engineering</td>
<td>1.375</td>
<td>0</td>
<td>-1.375</td>
<td>0.34 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>EFFF</td>
<td>1</td>
</tr>
<tr>
<td>07h</td>
<td>0 ~ 1 V</td>
<td>Engineering</td>
<td>1.375</td>
<td>0</td>
<td>Don’t care</td>
<td>0.34 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>Don’t care</td>
<td>1</td>
</tr>
<tr>
<td>08h</td>
<td>±500 mV</td>
<td>Engineering</td>
<td>687.5</td>
<td>0</td>
<td>-687.5</td>
<td>0.16 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>EFFF</td>
<td>1</td>
</tr>
<tr>
<td>09h</td>
<td>0 ~ 500 mV</td>
<td>Engineering</td>
<td>687.5</td>
<td>0</td>
<td>Don’t care</td>
<td>0.16 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>Don’t care</td>
<td>1</td>
</tr>
<tr>
<td>0ah</td>
<td>4 ~ 20 mA</td>
<td>Engineering</td>
<td>22</td>
<td>4.0</td>
<td>Don’t care</td>
<td>5.3 µA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>02E9</td>
<td>Don’t care 1</td>
</tr>
<tr>
<td>0bh</td>
<td>0 ~ 20 mA</td>
<td>Engineering</td>
<td>22</td>
<td>0</td>
<td>Don’t care</td>
<td>5.3 µA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two’s Comp</td>
<td>0FF</td>
<td>0</td>
<td>Don’t care</td>
<td>1</td>
</tr>
</tbody>
</table>

**Note:** The full scale values in this table are theoretical values for your reference; actual values will vary.
B.5 Analog Output Formats

You can configure ADAM analog output modules to receive data from the host in Engineering Units.

**Engineering Units**

Data can be represented in engineering units by setting bits 0 and 1 of the data format/checksum/integration time parameter to 0.

This format presents data in natural units, such as milliamps. The Engineering Units format is readily parsed by the majority of computer languages as the total data string length is fixed at six characters: two decimal digits, a decimal point and three decimal digits. The resolution is 5 µA.

**Example:**

An analog output module on channel 1 of slot 0 in an ADAM-5000 system at address 01h is configured for a 0 to 20 mA range. If the output value is +4.762 mA, the format of the Analog Data Out command would be #01S0C14.762<cr>

B.6 Analog Output Ranges

<table>
<thead>
<tr>
<th>Range Code</th>
<th>Output Range Description</th>
<th>Data Formats</th>
<th>Maximum Specified Signal</th>
<th>Minimum Specified Signal</th>
<th>Displayed Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0 to 20 mA</td>
<td>Engineering Units</td>
<td>20.000</td>
<td>00.000</td>
<td>5 µA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of Span</td>
<td>+100.00</td>
<td>+000.00</td>
<td>5 µA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hexadecimal Binary</td>
<td>FFF</td>
<td>000</td>
<td>5 µA</td>
</tr>
<tr>
<td>31</td>
<td>4 to 20 mA</td>
<td>Engineering Units</td>
<td>20.000</td>
<td>04.000</td>
<td>5 µA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of Span</td>
<td>+100.00</td>
<td>+000.00</td>
<td>5 µA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hexadecimal Binary</td>
<td>FFF</td>
<td>000</td>
<td>5 µA</td>
</tr>
<tr>
<td>32</td>
<td>0 to 10 V</td>
<td>Engineering Units</td>
<td>10.000</td>
<td>00.000</td>
<td>2.442 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>% of Span</td>
<td>+100.00</td>
<td>+000.00</td>
<td>2.442 mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hexadecimal Binary</td>
<td>FFF</td>
<td>000</td>
<td>2.442 mV</td>
</tr>
</tbody>
</table>
### B.7 ADAM-5013 RTD Input Format and Ranges

<table>
<thead>
<tr>
<th>Range Code (hex)</th>
<th>Input Range Description</th>
<th>Data Formats</th>
<th>Maximum Specified Signal</th>
<th>Minimum Specified Signal</th>
<th>Displayed Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>100 Ohms Platinum RTD -100 to 100° C (a=0.00385)</td>
<td>Engineering Units</td>
<td>+100.00</td>
<td>-100.00</td>
<td>±0.1° C</td>
</tr>
<tr>
<td>21</td>
<td>100 Ohms Platinum RTD 0 to 100° C (a=0.00385)</td>
<td>Engineering Units</td>
<td>+100.00</td>
<td>+000.00</td>
<td>±0.1° C</td>
</tr>
<tr>
<td>22</td>
<td>100 Ohms Platinum RTD 0 to 200° C (a=0.00385)</td>
<td>Engineering Units</td>
<td>+200.00</td>
<td>+000.00</td>
<td>±0.2° C</td>
</tr>
<tr>
<td>23</td>
<td>100 Ohms Platinum RTD 0 to 600° C (a=0.00385)</td>
<td>Engineering Units</td>
<td>+600.00</td>
<td>+000.00</td>
<td>±0.6° C</td>
</tr>
<tr>
<td>24</td>
<td>100 Ohms Platinum RTD -100 to 100° C (a=0.00392)</td>
<td>Engineering Units</td>
<td>+100.00</td>
<td>-100.00</td>
<td>±0.1° C</td>
</tr>
<tr>
<td>25</td>
<td>100 Ohms Platinum RTD 0 to 100° C (a=0.00392)</td>
<td>Engineering Units</td>
<td>+100.00</td>
<td>+000.00</td>
<td>±0.1° C</td>
</tr>
<tr>
<td>26</td>
<td>100 Ohms Platinum RTD 0 to 200° C (a=0.00392)</td>
<td>Engineering Units</td>
<td>+200.00</td>
<td>+000.00</td>
<td>±0.2° C</td>
</tr>
</tbody>
</table>

**Note:** See next page for table continuation.
### Appendix B  Data Formats and I/O Ranges

**Note:** This table continued from previous page.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Engineering Units</th>
<th>+000.00</th>
<th>+000.00</th>
<th>±0.6°C C</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>100 Ohms Platinum RTD 0 to 600°C a=0.00392</td>
<td></td>
<td></td>
<td>Engineering Units</td>
<td>+600.00</td>
<td>+000.00</td>
<td>±0.6°C C</td>
</tr>
<tr>
<td>28</td>
<td>120 Ohms Nickel RTD -80 to 100°C</td>
<td></td>
<td></td>
<td>Engineering Units</td>
<td>+100.00</td>
<td>-80.00</td>
<td>±0.1°C C</td>
</tr>
<tr>
<td>29</td>
<td>120 Ohms Nickel RTD 0 to 100°C</td>
<td></td>
<td></td>
<td>Engineering Units</td>
<td>+100.00</td>
<td>+000.00</td>
<td>±0.1°C C</td>
</tr>
</tbody>
</table>
## ADAM 5000 AI/AO Scaling

<table>
<thead>
<tr>
<th>Module</th>
<th>Type</th>
<th>Range Low</th>
<th>Range High</th>
<th>Scale Low</th>
<th>Scale High</th>
<th>Data Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>385(IEC)</td>
<td>0-100</td>
<td>100</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>600</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5013RTD</td>
<td>0-100</td>
<td>100</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>600</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ni</td>
<td>-80</td>
<td>100</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5017AI</td>
<td>mV</td>
<td>-150</td>
<td>150</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
</tr>
<tr>
<td>mV</td>
<td>-500</td>
<td>500</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>-5</td>
<td>5</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>-10</td>
<td>10</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>mA</td>
<td>-20</td>
<td>20</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>-2.5</td>
<td>2.5</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>2.5</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>mA</td>
<td>4</td>
<td>20</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>mA</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>5017H AI</td>
<td>mV</td>
<td>-15</td>
<td>15</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
</tr>
<tr>
<td>mV</td>
<td>-50</td>
<td>50</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>-10</td>
<td>10</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>-5</td>
<td>5</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>-2.5</td>
<td>2.5</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>2.5</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>mA</td>
<td>4</td>
<td>20</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>mA</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>5018 AI</td>
<td>mV</td>
<td>-15</td>
<td>15</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
</tr>
<tr>
<td>mV</td>
<td>-50</td>
<td>50</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>mV</td>
<td>-100</td>
<td>100</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>mV</td>
<td>-500</td>
<td>500</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>T/C(J)</td>
<td>0</td>
<td>760</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>T/C(K)</td>
<td>0</td>
<td>1370</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>T/C(T)</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>T/C(E)</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>T/C(R)</td>
<td>500</td>
<td>1750</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>T/C(S)</td>
<td>500</td>
<td>1750</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>T/C(B)</td>
<td>500</td>
<td>1800</td>
<td>0</td>
<td>65535</td>
<td>U16B</td>
<td></td>
</tr>
<tr>
<td>5024 AO</td>
<td>V</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>4095</td>
<td>U12B</td>
</tr>
<tr>
<td>mA</td>
<td>4</td>
<td>20</td>
<td>0</td>
<td>4095</td>
<td>U12B</td>
<td></td>
</tr>
<tr>
<td>mA</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>4095</td>
<td>U12B</td>
<td></td>
</tr>
</tbody>
</table>
Appendix C
Grounding Reference
Field Grounding and Shielding Application

Overview

Unfortunately, it’s impossible to finish a system integration task at one time. We always meet some trouble in the field. A communication network 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.

1. Grounding
   1.1 The ‘Earth’ for reference
   1.2 The ‘Frame Ground’ and ‘Grounding Bar’
   1.3 Normal Mode and Common Mode
   1.4 Wire impedance
   1.5 Single Point Grounding

2. Shielding
   2.1 Cable Shield
   2.2 System Shielding

3. Noise Reduction Techniques

4. Check Point List
C.1 Grounding

1-1 The ‘Earth’ for reference

As you know that the EARTH cannot be conductive. 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.

*Figure C-1: Think the EARTH as GROUND.*

As you know, the EARTH cannot be conductive. 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.
1-2 The ‘Frame Ground’ and ‘Grounding Bar’

Neutral is the physical cable from Generator. Ground is the local physical cable that connected to Ground Bar.

**Figure 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, telecommunication networks, etc. Those individual grounding bars not only provide the individual reference point, but also make the earth a our ground!
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 nonsense 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 equipments.

**Normal Mode**: refers to defects occurring between the live and neutral conductors. Normal mode is sometimes abbreviated as NM, or L-N for live-to-neutral.

**Common Mode**: refers to defects occurring between either conductor and ground. It is sometimes abbreviated as CM, or N-G for neutral-to-ground.

*Figure C-3: Normal mode and Common modC*
Normal Mode & Common Mode

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

**Figure C-4: Normal mode and Common mode**

- Ground-pin is longer than others, for first contact to power system and noise bypass.
- Neutral-pin is broader than Live-pin, for reducing contact impedance.
1-4 Wire impedance

What's the purpose of High Voltage Transmission?

Referring to Ohm rule, above diagram shows that how to reduce the power loss on cable.

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 voltage. What is the purpose of high voltage transmission wires? According to the energy formula, \( P = V \times I \), 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, \( V = I \times R \) this decreased current means lower power losses in the wires. So, high voltage lines are for reducing the cost of moving electrical power from one place to another.
Wire Impedance

The wire impedance will consume the power.

Figure C-6: wire impedance
1-5 Single Point Grounding

Single Point Grounding

Those devices will influence each other with swiftly load change.

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.
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.

Figure C-8: Single point grounding (2)
C.2 Shielding

2-1 Cable Shield

Single Isolated Cable

Use Aluminum foil to cover those wires, for isolating the external noise.

Figure C-9: Single isolated cable

- Single isolated cable
  
The diagram shows the structure of an isolated cable. 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 noise.
Figure C-10: Double isolated cable

- Double isolated cable

Figure 10 is an example of a double isolated cable. 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 noise.

Additionally, follow these tips just for your reference.

- 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 device. (for EMI consideration)
- Don’t strip off too long of plastic cover for soldering.
2-2 System Shielding

Figure C-11: System Shielding

- Never stripping 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. Please refer to following page 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)
Characteristic of Cable

This will destroy the twist rule.

Don’t strip off too long of plastic cover for soldering, or will influence the characteristic of twisted-pair cable.

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.
System Shielding

A difficult way for signal.

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.
System Shielding

A more easy way for signal.

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 noise.

![Noise Reduction Techniques Diagram]

*Figure C-15: Noise Reduction Techniques*

Separate Load and Device power.
Cascade amplify/isolation circuit before I/O channel.
Appendix C  Grounding Reference

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?