

ADAM 4000 Series

**Data Acquisition Modules
User's Manual**

ADAM 4000 Series

Data Acquisition Modules

User's Manual

Copyright Notice

This document is copyrighted, 1997, 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.

CE Notification

The ADAM-4000 series developed by Advantech Co., Ltd. has passed the CE test for environmental specifications when operated within an industrial enclosure (ADAM-4950-ENC). Therefore, in order to protect the ADAM modules from being damaged by ESD (Electric Static Discharge), we strongly recommend that the use of CE-compliant industrial enclosure products when using any ADAM module.

Acknowledgments

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

Table of Contents

| | | |
|------------------|--|------------|
| Chapter 1 | Introduction | 1-1 |
| 1.1 | Overview | 1-2 |
| 1.2 | Applications | 1-4 |
| Chapter 2 | Installation Guideline | 2-1 |
| 2.1 | System Requirements to set up an ADAM network | 2-3 |
| 2.2 | Basic configuration and hook-up | 2-6 |
| 2.3 | Baud rate and Checksum | 2-9 |
| 2.4 | Multiple Module Hookup | 2-11 |
| 2.5 | Application Example | 2-12 |
| Chapter 3 | I/O Modules | 3-1 |
| 3.1 | ADAM-4011/4011D/4012/4013/4015 Analog Input Modules | 3-2 |
| 3.2 | ADAM-4016 Analog Input/Output Module | 3-15 |
| 3.3 | ADAM-4017/4017+/4018/4018M/4018+ 8-channel Analog Input Modules | 3-20 |
| 3.4 | ADAM-4019 8-channel Universal Analog Input Module | 3-32 |
| 3.5 | ADAM-4021 Analog Output Module | 3-35 |
| 3.6 | ADAM-4050/4051/4052/4053/4055 Digital I/O Modules | 3-38 |
| 3.7 | ADAM-4060/4068 Relay Output Module | 3-50 |
| 3.8 | ADAM-4080/4080D Counter/Frequency Input Modules | 3-54 |
| Chapter 4 | Command Set | 4-1 |
| 4.1 | Introduction | 4-2 |
| 4.2 | Syntax | 4-2 |
| 4.3 | I/O Module Commands Search Table | 4-4 |
| 4.4 | Analog Input Module Command | 4-41 |
| 4.4.1 | Analog Input Command Set | 4-41 |
| 4.4.2 | Data Conversion and Display Command Set | 4-77 |
| 4.4.3 | Analog Input Data Logger Command Set | 4-89 |
| 4.4.4 | Digital I/O, Alarm and Event Command Set | 4-103 |
| 4.4.5 | Excitation Voltage Output Command Set | 4-119 |
| 4.5 | Analog Output Module Command | 4-127 |

| | | |
|-------------------|--|------------|
| 4.6 | Digital I/O and Relay Output Module Command | 4-151 |
| 4.7 | Counter/Frequency Module Command | 4-167 |
| 4.7.1 | Configuration, Counter Input and Display Command Set | 4-167 |
| 4.7.2 | Counter Setup Command Set | 4-179 |
| 4.7.3 | Digital Filter and Programmable Threshold Command Set .. | 4-189 |
| 4.7.4 | Digital Output and Alarm Command Set | 4-201 |
| Chapter 5 | Calibration | 5-1 |
| 5.1 | Analog Input Module Calibration | 5-2 |
| 5.2 | Analog Input Resistance Calibration | 5-7 |
| 5.3 | Analog Output Calibration | 5-9 |
| Appendix A | Technical Specifications | A-1 |
| A.1 | ADAM-4011 Thermocouple Input Module | A-2 |
| A.2 | ADAM-4011D Thermocouple Input Module with LED Display | A-5 |
| A.3 | ADAM-4012 Analog Input Module | A-8 |
| A.4 | ADAM-4013 RTD Input Module | A-10 |
| A.5 | ADAM-4014D Analog Input Module with LED Display | A-12 |
| A.6 | ADAM-4016 Strain Gauge Input Module | A-14 |
| A.7 | ADAM-4017, 4017+ 8-Channel Analog Input Module | A-16 |
| A.8 | ADAM-4018, 4018+ 8-channel Analog Input Module | A-18 |
| A.9 | ADAM-4018M 8-channel Analog Input Data Logger | A-20 |
| A.10 | ADAM-4021 Analog Output Module | A-22 |
| A.11 | ADAM-4050 Digital I/O Module | A-26 |
| A.12 | ADAM-4052 Isolated Digital Input Module | A-28 |
| A.13 | ADAM-4053 16-channel Digital Input Module | A-30 |
| A.14 | ADAM-4060 Relay Output Module | A-32 |
| A.15 | ADAM-4080 Counter/Frequency Input Module | A-34 |
| A.16 | ADAM-4080D Counter/Frequency Input Module with LED Display ... | A-36 |
| Appendix B | Data Formats and I/O Ranges | B-1 |
| B.1 | Analog Input Formats | B-2 |
| B.1.1 | Engineering Units | B-2 |
| | B-3 | |
| B.1.2 | Percent of FSR | B-3 |
| B.1.3 | Twos complement hexadecimal | B-5 |
| B.1.4 | Ohms | B-7 |

| | | |
|---|-----------------------------------|------------|
| B.2 | Analog Input Ranges | B-8 |
| B.3 | Analog Output Formats | B-13 |
| B.3.1 | Engineering Units | B-13 |
| B.3.2 | Percent of Span | B-14 |
| B.3.3 | Hexadecimal | B-14 |
| B.4 | Analog Output Ranges | B-15 |
| Appendix C Technical Diagrams | | C-1 |
| C.1 | ADAM Dimensions | C-2 |
| C.2 | Installation | C-3 |
| C.2.1 | DIN-Rail Mounting | C-3 |
| C.2.2 | Panel Mounting | C-5 |
| C.2.3 | Piggyback Stack | C-7 |
| Appendix D Utility Software | | D-1 |
| D.1 | ADAM-4000 Utility Software | D-1 |
| D.2 | ADAM-4018M Utility Software | D-7 |
| Appendix E RS-485 Network | | E-1 |
| E.1 | Basic Network Layout | E-2 |
| E.2 | Line Termination | E-5 |
| E.3 | RS-485 Data Flow Control | E-8 |
| Appendix F How to use the Checksum feature | | F-1 |
| F.1 | Checksum Enable/Disable | F-2 |

Introduction

1

1.1 Overview

The ADAM Series is a set of intelligent sensor-to-computer interface modules containing built-in microprocessor. They are remotely controlled through a simple set of commands issued in ASCII format and transmitted in RS-485 protocol. They provide signal conditioning, isolation, ranging, A/D and D/A conversion, data comparison, and digital communication functions. Some modules provide digital I/O lines for controlling relays and TTL devices.

Software Configuration and Calibration

ADAM modules contain no pots or switches to set. By merely issuing a command from the host computer, you can change an analog input module to accept several ranges of voltage input, thermocouple input or RTD input. All the module's configuration parameters including I/O address, speed, parity, HI and LO alarm, calibration parameters settings may be set remotely. Remote configuration can be done by using either the provided menu-based software or the command set's configuration and calibration commands.

By storing configuration and calibration parameters in a nonvolatile EEPROM, modules are able to retain these parameters in case of power failure.

Watchdog Timer

A watchdog timer supervisory function will automatically reset the ADAM modules in the event of system failure. Maintenance is thus simplified.

Power Requirements

Although the modules are designed for standard industrial unregulated 24 V_{DC} power supply, they accept any power unit that supplies power within the range of +10 to +30 V_{DC} . The power supply ripple must be limited to 5 V peak-to-peak, and the immediate ripple voltage should be maintained between +10 and +30 V_{DC} .

Connectivity and Programming

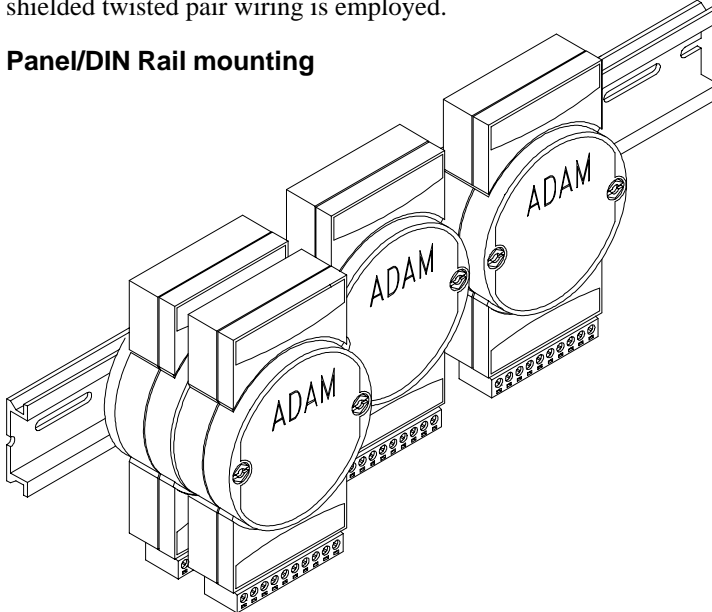
ADAM modules can connect to and communicate with all computers and terminals. They use RS-485 transmission standards, and communicate with ASCII format commands. The command set for every module type consists of approximately ten different commands. The command set for input modules is larger because it incorporates alarm functions. All communications to and from the module are performed in ASCII, which means that ADAM modules can be programmed in virtually any high-level language.

RS-485 Network

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

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

Panel/DIN Rail mounting



ADAM modules mount on any panel, on provided brackets, on DIN rails or may be stacked together.

The RS-485 network, together with screw-terminal plug connectors, allows for system expansion, reconfiguration and repair without disturbing field wiring.

Protection against the environment

Hardened plastic packing forms the outer shell of every module. Since all configuration is controlled by software, the module is not designed to be opened. This greatly enhances resistance against corrosive materials, moisture and vibration. ADAM modules' low power requirements help them to operate in temperatures from 0 to 70°C and in humidities from 0 to 95% (non-condensing). They're built compactly using automated SMT technology so you can pack them into water-tight and explosion-proof industrial enclosures.

1.2 Applications

- Remote data acquisition
- Process monitoring
- Industrial process control
- Energy management
- Supervisory control
- Security systems
- Laboratory automation
- Building automation
- Product testing
- Direct digital control

Installation Guideline

2

This chapter provides guidelines to what is needed to set up and install an ADAM network. A quick hookup scheme is provided that lets you configure modules before they are installed in a network.

To help you to connect ADAM modules with sensor inputs, several wiring examples are provided. Finally, you will find at the end of this chapter a programming example using the ADAM command set.

Be sure to carefully plan the layout and configuration of your network before you start. Guidelines regarding layout are given in Appendix E: RS-485 Network.

NOTICE: *Except for the communication modules, which have on-board switches for their baud rate setting, ADAM modules should not be opened. There is no need to open the ADAM modules: all configuration is done remotely and there are no user serviceable parts are inside. Opening the cover will therefore void the warranty.*

2.1 System Requirements to set up an ADAM network

The following list gives an overview of what is needed to setup, install and configure an ADAM environment.

- ADAM modules
- A host computer, such as an IBM PC/AT compatible, that can output ASCII characters with an RS-232C or RS-485 port.
- Power supply for the ADAM modules (+10 to +30 V_{DC})
- ADAM Series Utility software
- ADAM Isolated RS-232/RS-485 Converter (optional)
- ADAM Repeater (optional)

Host computer

Any computer or terminal that can output in ASCII format over either RS-232 or RS-485 can be connected as the host computer. When only RS-232 is available, an ADAM RS-232/RS-485 Converter is required to transform the host signals to the correct RS-485 protocol. The converter also provides opto-isolation and transformer-based isolation to protect your equipment.

Power supply

For the ease of use in industrial environments the ADAM modules are designed to accept industry standard +24 V_{DC} unregulated power. Operation is guaranteed when using any power supply between +10 and +30 V_{DC}. Power ripples must be limited to 5 V peak to peak while the voltage in all cases must be maintained between +10 and +30 V_{DC}. All power supply specifications are referenced at module connector. When modules are powered remotely, the effects of line voltage drops must be considered.

All modules use on-board switching regulators to sustain good efficiency over the 10-30 V input range, therefore we can assume that the actual current draw is inversely proportional to the line voltage. The following example shows how to calculate the required current that a power supply should be able to provide.

Installation Guideline

Assume that a $+24 V_{DC}$ will be used to power five ADAM-4011 Analog Input Modules. The distance from power supply to modules is not so big that significant line voltage drop will occur. One ADAM-4011 module consumes a maximum of 1.2 Watts. The total required power will equal $5 \times 1.2 = 6$ Watts. A power supply of $+24 V_{DC}$ should therefore be able to supply a minimal current of $6 / 24 = 0.25$ Amps.

Small systems may be powered by using wall-mounted modular power supplies. Also when modules operate on long communication lines (>500 feet) it is often more reliable to power the modules locally with modular power supplies. These inexpensive units can easily be obtained from any electronics retail store.

The power cables should be selected according to the number of modules connected and the length of the power lines. When using a network with long cables, we advise the use of thicker wire to limit the line voltage drop. In addition to serious voltage drops, long voltage lines can also cause interference with communication wires.

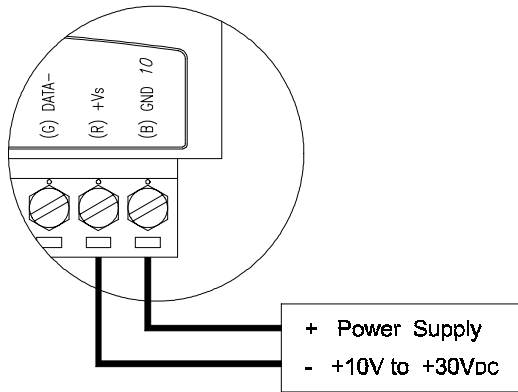


Figure 2-1 *Power Supply Connections*

We advise that the following standard colors (as indicated on the modules) be used for power lines:

| | | |
|-----|-----|-------|
| +Vs | (R) | Red |
| GND | (B) | Black |

Communication Wiring

We recommend that shielded-twisted-pair cables that comply with the EIA RS-485 standard be used with the ADAM network to reduce interference. Only one set of twisted-pair cables is required to transmit both Data and RTS signals. We advise that the following standard colors (as indicated on the modules) be used for the communication lines:

| | |
|-----------|--------|
| DATA+ (Y) | Yellow |
| DATA- (G) | Green |

ADAM Utility Software

A menu-driven utility program is provided for ADAM module configuration, monitoring and calibration. It also includes a terminal emulation program that lets you easily communicate through the ADAM command set. (See Appendix D, Utility Software)

ADAM Communication Speed

In ADAM series, the baudrate can be configured from 1200 bps to 38.4 Kbps. And the baudrate of all modules in an RS-485 network must be the same.

ADAM Isolated RS-232/RS485 Converter (optional)

When the host computer or terminal has only a RS-232 port, an ADAM Isolated RS-232/RS-485 Converter, connected to the host's RS-232 port, is required. Since this module is not addressable by the host, the baud rate must be set using a switch inside the module. The factory default setting is 9600 baud.

ADAM Repeater (optional)

When communication lines exceed 4000 ft (1200 meter) or the number of ADAM modules connected is more than 32, a repeater should be connected to expand the first segment. Up to 8 Repeater modules can be connected allowing connection of up to 256 ADAM modules. As with the Converter module, the Repeater module is not addressable by the host and the baud rate must be set by changing the switch inside the module. The factory default setting is 9600 baud.

2.2 Basic configuration and hook-up

Before placing a module in an existing network, the module should be configured. Though all modules are initially configured at the factory, it is recommended to check that the baud rate is set correctly.

Default Factory Settings

Baud rate: 9600 Bit/sec.
Address: 01 (hexadecimal)

The basic hook-up for module configuration is shown below.

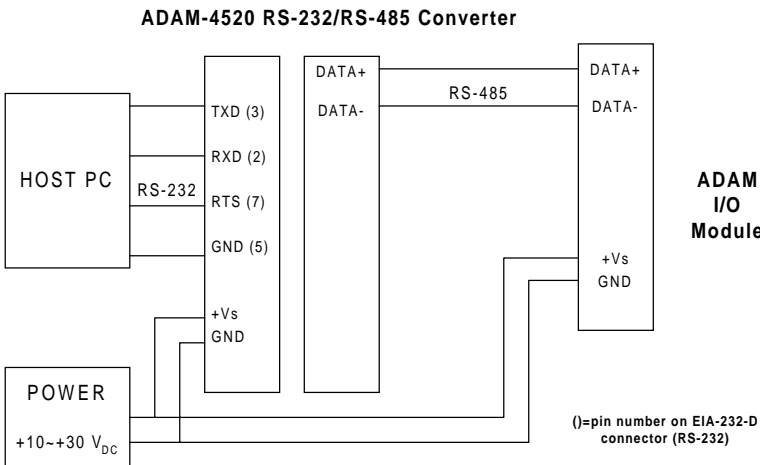


Figure 2-2 Basic Hook-up of ADAM Module to Host Switches

The following items are required to configure a module: an ADAM converter module, a personal computer with RS-232 port (baud rate set to 9600) and the ADAM utility software.

Configuration with the ADAM Utility Software

The easiest way to configure the ADAM module is by using the ADAM utility software: an easy-to-use menu-structured program will guide you through every step of the configuration. (See Appendix D, Utility Software)

Configuration with the ADAM command set

ADAM modules can also be configured by issuing direct commands from within a terminal emulation program that is part of the ADAM utility software.

The following example guides you through the setup of an analog input module. Assume that an ADAM-4011 Analog Input module still has its default settings (baud rate 9600 and address 01h). Before the module is reconfigured, it is first requested to send its default settings.

NOTICE: *An analog input module requires a maximum of 7 seconds to perform auto calibration and ranging after it is rebooted or powered on. During this time span, the module can not be addressed to perform any other actions.*

Example:

Make sure that the module is properly connected as shown in figure 2-5. Power up all the connected devices, start the terminal emulation program, and issue the following command:

```
$012(cr)
```

requests that module with address 01 send its configuration status

```
!01050600
```

Module at address 01 responds that it is configured for an input range of +/-2.5 V, baud rate 9600, integration time of 50 ms (60 Hz), engineering units and no checksum checking or generation.

Installation Guideline

To change the configuration setting of the analog input module, the following command is issued:

```
%01070F0600(cr)
```

% = change configuration

01 = target module at address 00 to:

07 = change address to 07 hexadecimal

0F = set input range to Type K thermocouple

06 = set baud rate to 9600

00 = set integration time to 50 ms (60 Hz)

disable checksum

set data format to engineering units

(See Chapter 4, Command Set for a full description of the syntax of the configuration command for an analog input module)

When the module received the configuration command it will respond with its new address:

```
!07(cr)
```

Wait 7 seconds to let the new configuration settings take effect before issuing a new command to the module.

NOTICE: *All reconfiguration except changing of baud rate and checksum values can be done dynamically, i.e. the modules need not to be reset.*

When changing the baud rate or checksum, these changes should be made for all connected devices. After reconfiguration, all modules should be powered down and powered up to force a reboot and let the changes take effect. See the next page for a strategy for changing baud rate and or checksum for an entire network.

2.3 Baud rate and Checksum

Adam modules contain EEPROMs to store configuration information and calibration constants. The EEPROM replaces the usual array of switches and pots required to specify baud rate, input/output range etc. All of the ADAM modules can be configured remotely through their communication ports, without having to physically alter pot or switch settings.

Since there is no visual indication of a module's configuration status, it is impossible just by looking at it what the baud rate, address and other settings are. It might not be possible to establish communications with a module whose baud rate and address are unknown. To overcome this problem, every module has an input terminal labeled INIT*. By booting the module while connecting the INIT* terminal with the module's GND terminal, the modules configuration is forced into a known state. This state is called the INIT* state.

INIT* state defaults:

Baud rate: 9600

Address: 00h

Checksum: disabled

Forcing the module in the INIT* state does not change any parameters in the module's EEPROM. When the module is in the INIT* state with its INIT* and GND terminals shorted, all configuration settings can be changed and the module will respond to all other commands normally.

Changing Baud rate and Checksum

Baud rate and checksum settings have several things in common:

- They should be the same for all modules and host computer.
- Their setting can only be changed by putting a module in the INIT* state.
- Changed settings can only take effect after a module is rebooted

To alter baud rate or checksum settings you must perform the following steps:

- Power on all components except the ADAM Module.
- Power the ADAM module on while shorting the INIT* and GND terminals (See Figure 2-3).

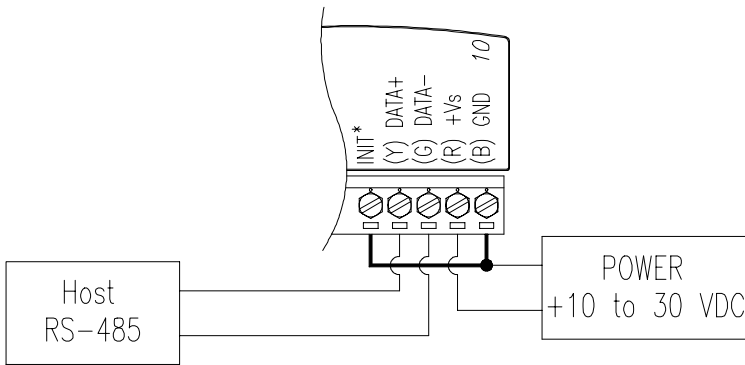


Figure 2-3 *Grounding the INIT* Terminal*

- Wait at least 7 seconds to let self calibration and ranging take effect.
- Configure the checksum status and/or the baud rate.
- Switch the power to the ADAM Module OFF.
- Remove the grounding of the INIT* terminal and power the module on.
- Wait at least 7 seconds to let self calibration and ranging take effect.
- Check the settings (If the baud rate has changed, the settings on the host computer should be changed accordingly).

2.4 Multiple Module Hookup

The Figure below shows how ADAM modules are connected in a multiple module example:

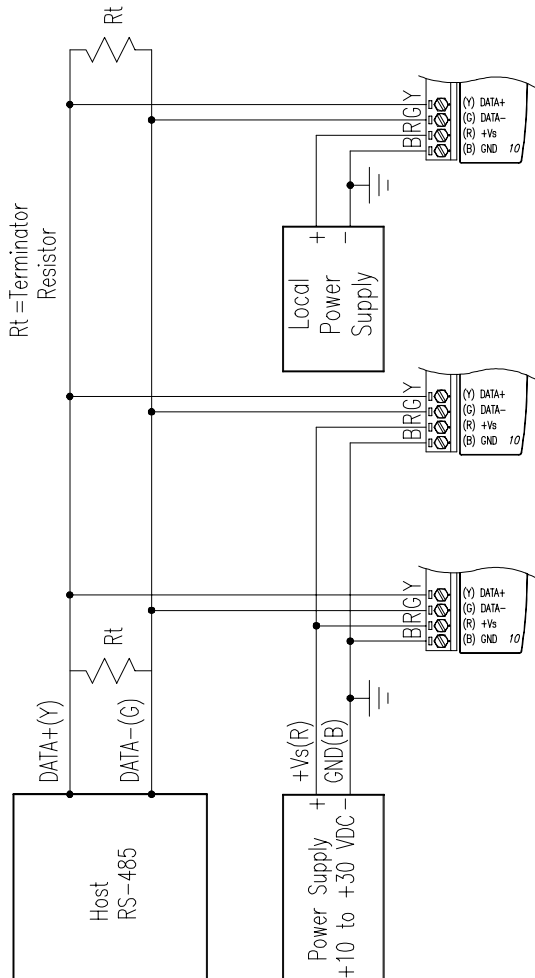


Figure 2-4 Multi-module Connection

2.5 Application Example

ADAM-4011 alarm functions may be used to build a simple ON-OFF controller application that will operate without host intervention.

When the proper alarm settings have been stored in ADAM's EEPROM it would be able to function as a stand alone device where no communication lines are required.

A simple controller application would use the momentary alarm output to control the process. Lets assume we are controlling a heating process. The input of the Analog Input will be the process' temperature and its output determine whether the heater is turned on or turned off. (See Figure 2-5)

In order to maintain a steady temperature set the LO limit of the alarm function to desired setpoint and configure the alarm mode as Momentary. Utilize the LO alarm output (DO0/LO) to control the SSR relay that controls the heater.

If the module measures a temperature that undergoes the LO alarm setting it will turn the LO alarm high which causes the heater to be switched on. When a temperature is measured that exceeds the LO alarm setting the LO alarm is set to low and the heater is turned off. In this application the HI alarm output is still available to activate an alarm or generate an emergency shut-down if the temperature gets out of control.

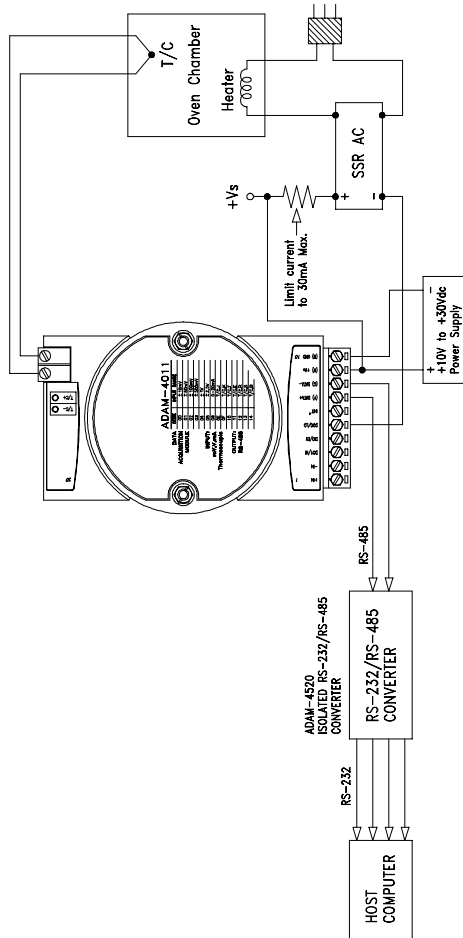


Figure 2-5 Simple ON/OFF Controller Function

Programming Example

The following program is a simple program written in BASIC that resembles our application example. The program first configures the ADAM-4011 module to act as an ON/OFF controller and then monitors and displays the process temperature.

I/O Modules

3

3.1 ADAM-4011/4011D/4012/4013/4015 Analog Input Modules

Analog input modules use a microprocessor-controlled integrating A/D converter to convert sensor voltage, current, thermocouple, or RTD signals into digital data. The digital data is then translated into either engineering units, two's complement hexadecimal format or percentage of full-scale range (FSR) according to the module's configuration. When prompted by the host computer, the data is sent through a standard RS-485 interface.

The Analog Input Modules offer signal conditioning, A/D conversion, ranging, and RS-485 digital communication functions. They protect your equipment from ground loops and power surges by providing optoisolation of the A/D input and transformer based isolation up to 3000 V_{DC}. (ADAM-4011 has transformer-based isolation up to 500 V_{DC}.)

Open Thermocouple Detection and Input Surge Protection (ADAM-4011D only)

The ADAM-4011D provides an open thermocouple detection function. Users can use a simple command to detect whether the thermocouple is open or closed. The module also provides surge protection on its input channel. Internal high speed transient suppressor on its input channel protects the module from dangerous spikes and voltages.

Front Panel LED Indicator (ADAM-4011D only)

The 4½ digit LED display on the back of the ADAM-4011D lets you monitor process readings right at their source. The module displays readings in a wide variety of formats as well as high-low alarm messages. The ADAM-4011D offers flexibility, ease of installation and direct availability of process data. For critical process monitoring, this module is the ideal choice.

Digital Inputs/Outputs (Except ADAM-4013)

Analog input modules also contain two digital outputs and one digital input. Outputs are open-collector transistor switches that may be controlled by the host computer. They can control solid-state relays which in turn may control heaters, pumps, and other electrical powered equipment. The digital inputs may be read by the host computer and used to sense the state of a remote digital signal.

Event counting (Except ADAM-4013)

The event counter is connected to the Digital Input channel and can be used to keep track of the total amount of external low-speed pulses. Its accumulated maximal count is 65535. The number 65535 is held, even if the actual number of events exceeds 65535. The counter can be read or reset to 0 by the host computer.

Since the Event counter's data is not stored in EEPROM, the event counter is cleared and set to zero after every reset or power up of the analog input module.

Alarm signalling (Except ADAM-4013)

Analog input modules include High and Low alarm functions. High and Low alarm limits may be downloaded into the module's EEPROM by the host computer.

The alarm functions can be enabled or disabled remotely. When the alarm function is enabled, both Digital Output channels are used to indicate the High and Low alarm state. Digital Output channel 1 (DO1) equals High alarm state and Digital Output channel 0 (DO0) equals Low alarm state. The High and Low alarm states can be read at any time by the host computer.

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

There are two alarm mode options: Momentary and Latching.

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

When the module receives a value that is lower than the Low alarm limit, it will clear the High alarm and turn the Low alarm ON.

When the alarm is in Momentary mode, the alarm will be turned OFF as soon as the input value returns to within limits.

The arrangement of coupling High and Low alarm states with Digital Output lines may be utilized to build ON/OFF controllers that can operate without host computer involvement.

Function Description for the ADAM-4011 analog input module

To provide a better understanding of the functioning of the ADAM modules, the following is a description of the module with the most extensive set of functions, the ADAM-4011.

All analog input data first flows through the PGA (programmable gain amplifier). The amplifier can vary its gain from 1 to 128. The PGA automatically adjusts the signal to a range of -2.5 V to +2.5 V. This ensures optimal input voltage and resolution for the A/D converter.

The A/D conversion is supervised by the microprocessor that holds the calibration software. Two kinds of calibration take place automatically on startup or reset: Auto Zero calibration and Auto Span calibration. Normal calibration is used to adjust the signal according to calibration parameters defined by the user.

The digital 10 Hz filter provides a steady state output by using the $D\Sigma$ function.

Before the data enters the microprocessor it passes through an optical isolation device. The opto isolation prevents ground loops and limits the chance of damage from power surges.

The microprocessor has six basic functions:

- Linearization of T/C (Thermocouple)
- Communication software and command set
- Calibration software
- Alarm monitoring
- Event counting
- Management of the EEPROM device that holds the system parameters
- Data transformation

After data has been transformed to the right data format it is passed on the RS-485 output port.

If an input value exceeds the High alarm setting or falls below the Low alarm setting, a flag is set in one of the Digital Output channels.

Finally, the on-board switching regulator accepts voltage between +10 and +30 V_{DC}. This power circuit has an isolation value of 500 V_{DC} to protect your equipment from damage from power surges.

ADAM-4011

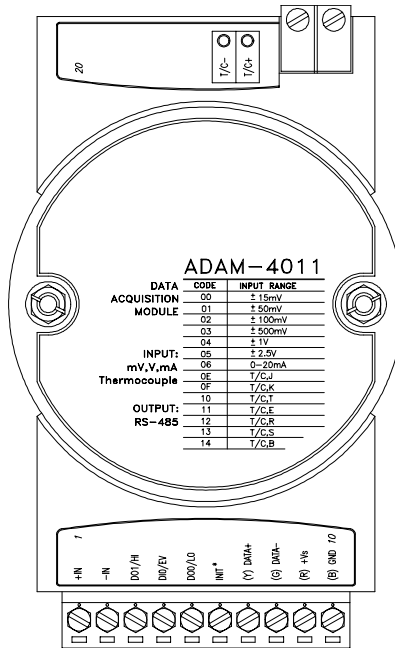


Figure 3-1 ADAM-4011 Thermocouple Input Module

Accepts:

- J, K, T, E, R, S and B thermocouples
- millivolt inputs: ± 15 mV, ± 50 mV, ± 100 mV and ± 500 mV
- Volt inputs: ± 1 V and ± 2.5 V
- Current input: ± 20 mA (Requires a 125 Ω resistor)

Two digital output channels and one digital input channel are provided.

Depending on the module's configuration setting, it can forward the data to the host computer in one of the following formats:

- engineering units ($^{\circ}$ C, mV, V, or mA)
- percent of full-scale range (FSR)
- twos complement hexadecimal

ADAM-4011D

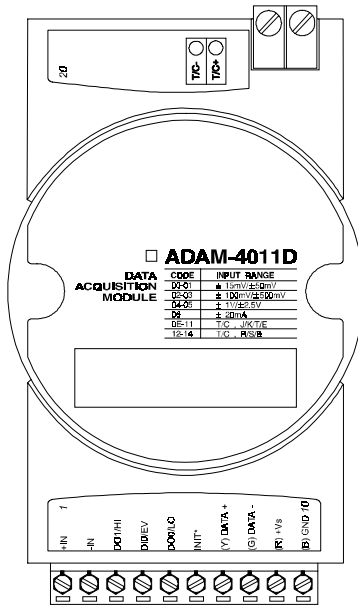


Figure 3-2 ADAM-4011D Thermocouple Input Module with LED Display
Accepts:

- J, K, T, E, R, S and B thermocouples
- millivolt inputs: ±15 mV, ±50 mV, ±100 mV and ±500 mV
- Volt inputs: ±1 V and ±2.5 V
- Current input: ±20 mA (Requires a 125 Ω resistor)

Two digital output channels and one digital input channel are provided.

Depending on the module's configuration setting, it can forward the data to the host computer in one of the following formats:

- engineering units (°C, mV, V, or mA)
- percent of full-scale range (FSR)
- twos complement hexadecimal

ADAM-4012

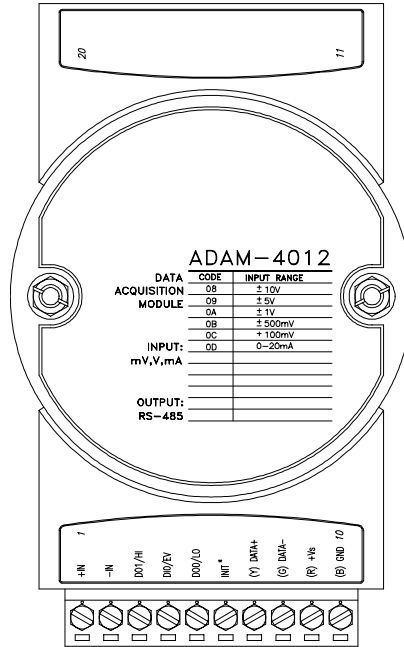


Figure 3-3 ADAM-4012 Analog Input Module

Accepts:

- millivolt inputs ± 150 mV and ± 500 mV
- volt inputs: ± 1 V, ± 5 V and ± 10 V
- current input: ± 20 mA (requires a 125Ω resistor)

Two digital output channels and one digital input channel are provided.

Depending on the module's configuration setting, it can forward the data to the host computer in one of the following formats:

- engineering units (mV, V, or mA)
- percent of full-scale range (FSR)
- twos complement hexadecimal

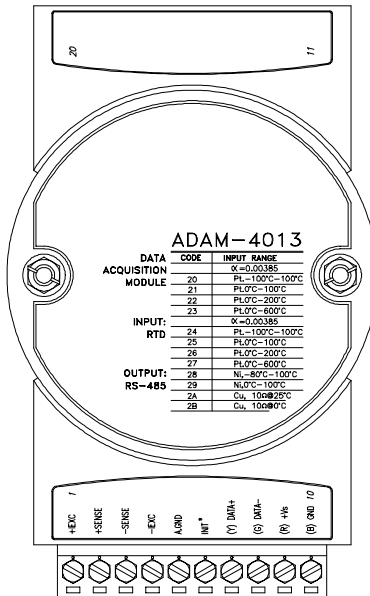


Figure 3-4 ADAM-4013 RTD Input Module

Accepts:

- input from platinum and nickel RTDs

Depending on the module's configuration setting, it can forward the data to the host computer in one of the following formats:

- engineering units (°C)
- percent of full-scale range (FSR)
- twos complement hexadecimal

Application Wiring

The following gives you examples how to connect various types of analog input and high-low alarm applications to your ADAM modules.

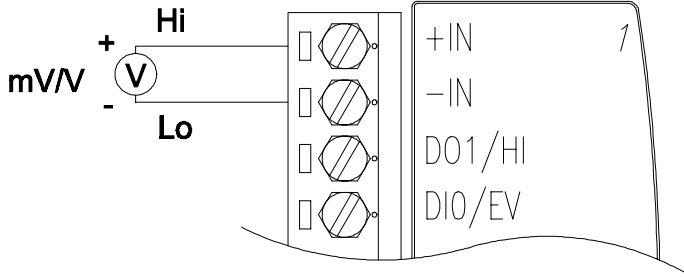


Figure 3-5 Millivolt and Volt Input

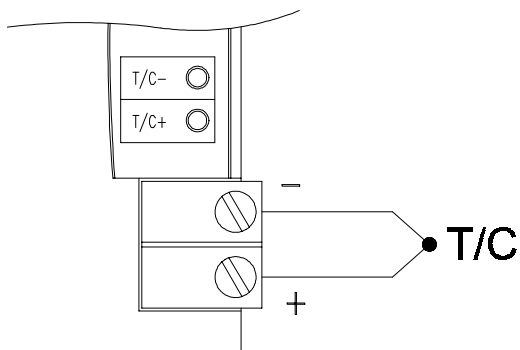


Figure 3-6 Thermocouple Input

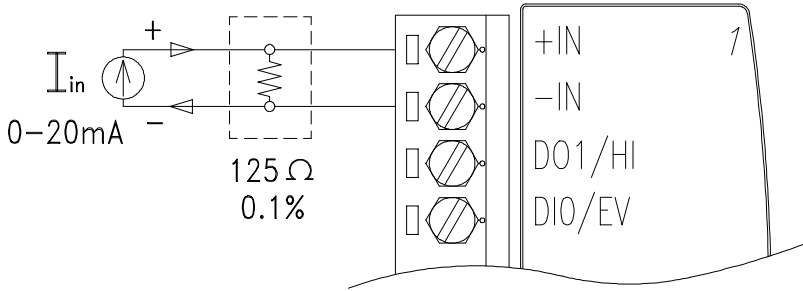


Figure 3-7 Process Current Input

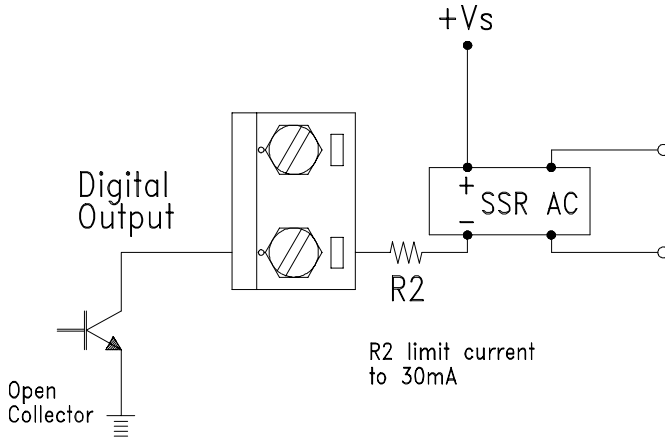


Figure 3-8 Digital Output used with SSR (HI-LO alarm)

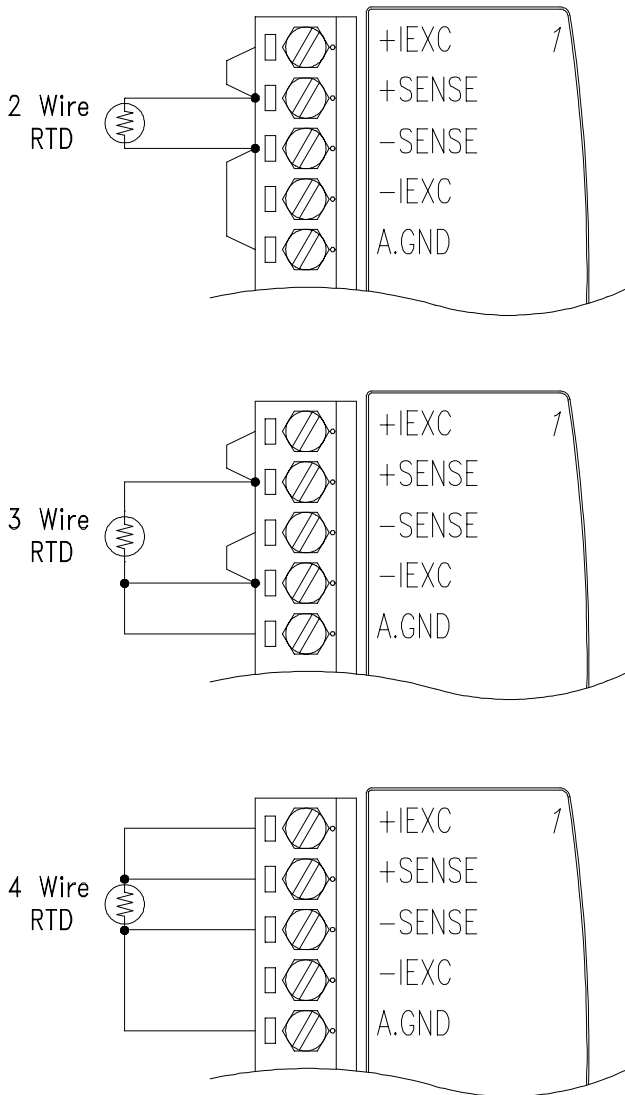


Figure 3-9 RTD Inputs

ADAM-4015 6-channel RTD Input Module

A RTD module is popular for temperature measurement. Unlike the traditional design, the ADAM-4015 provides six RTD input channels for different types of RTD signal as an effective solution in industrial & building automation. Usually, broken external wiring will lead to inaccurate current value. The ADAM-4015 provides a broken wiring detecting function so users can easily troubleshoot broken wiring problems.

ADAM-4015

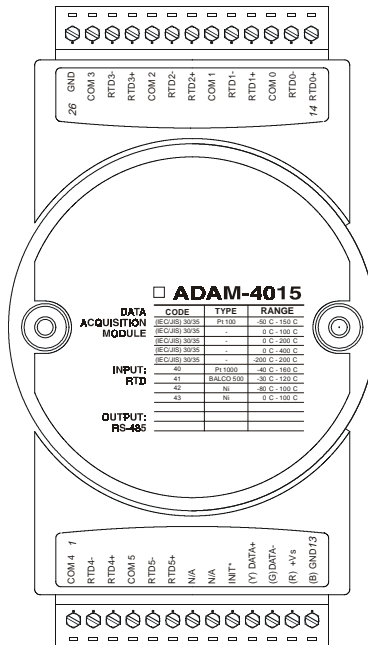


Figure 3-10: ADAM-4015 6-channel RTD Input Module

Application Wiring

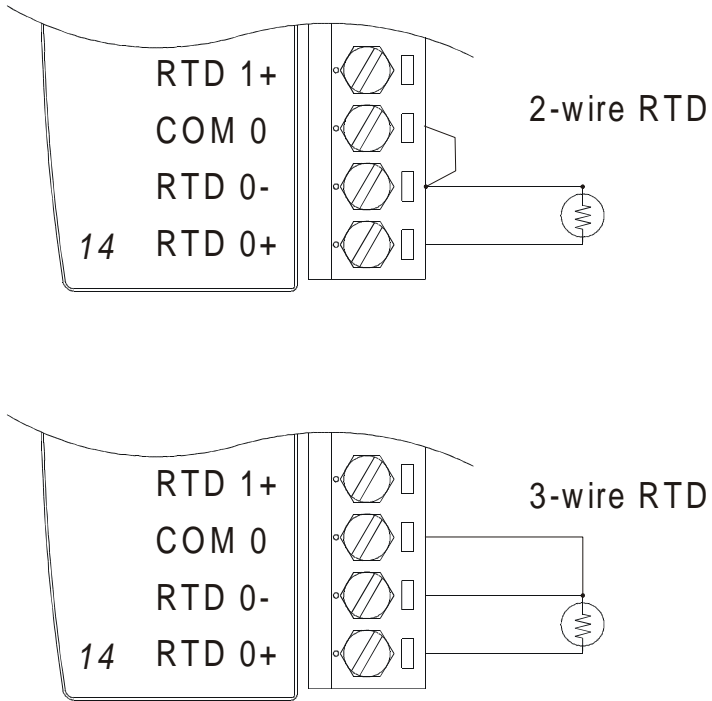


Figure 3-11: ADAM-4015 RTD Input Module Wiring Diagram

Technical specification of ADAM-4015

| | |
|---|--|
| Channel | 6 |
| Input Type | Pt100, Pt1000, BALCO500, Ni |
| Input type and temperature range | Pt100: -50 to 150 C 0 to 100 C 0 to 200 C 0 to 400 C -200 to 200 C Pt1000: -40 to 160 C BALCO500: -30 to 120 C Ni: -80 to 100 C 0 to 100 C |
| Isolation Voltage | 3000 V _{DC} |
| Sampling Rate | 12 sample/sec (total) |
| Input Impedance | 10 M Ω |
| Accuracy | +/- 0.1% or better |
| Power Consumption | 1 W |
| I/O Connector Type | 13-pin plug-terminal |

Table 3-1: Technical specification of ADAM-4015

3.2 ADAM-4016 Analog Input/Output Module

A strain gauge input module uses a microprocessor-controlled integrating A/D converter to convert sensor voltage or current signals into digital data for load cell and stress measurement. The digital data is then translated into either engineering units, two's complement hexadecimal format or percentage of full-scale range (FSR) according to the module's configuration. When prompted by the host computer, the data is sent through a standard RS-485 interface.

The strain gauge input module offers signal conditioning, A/D conversion, ranging, and RS-485 digital communication functions. They protect your equipment from ground loops and power surges by providing opto-isolation of the A/D input and transformer based isolation up to $3000 V_{DC}$.

Excitation Voltage Output

A strain gauge input module can supply single channel voltage output for excitation. The module receives digital input from the host computer. The format of the data is engineering units. It then uses its microprocessor-controlled D/A converter to convert the digital data into output signals.

Strain gauge input modules protect your equipment from ground loops and power surges by providing opto-isolation of the D/A output and transformer-based isolation up to $3000 V_{DC}$.

Digital Outputs

A strain gauge input module also contains 4 digital outputs. Outputs are open-collector transistor switches that may be controlled by the host computer. They can control solid-state relays which in turn may control heaters, pumps, and other electrical equipment.

Alarm signalling

Strain Gauge input modules include High and Low alarm functions. High and Low alarm limits may be downloaded into the module's EEPROM by the host computer.

The alarm functions can be enabled or disabled remotely. When the alarm function is enabled, both Digital Output channels are used to indicate the High and Low alarm state. Digital Output channel 1 (DO1) equals High alarm state and Digital Output channel 0 (DO0) equals Low alarm state. The High and Low alarm states can be read at any time by the host computer.

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

There are two alarm mode options: Momentary and Latching.

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

When the module receives a value that is lower than the Low alarm limit, it will clear the High alarm and turn the Low alarm ON.

When the alarm is in Momentary mode, the alarm will be turned OFF as soon as the input value returns to within limits.

The arrangement of coupling High and Low alarm states with Digital Output lines may be utilized to build ON/OFF controllers that can operate without host computer involvement. .

ADAM-4016

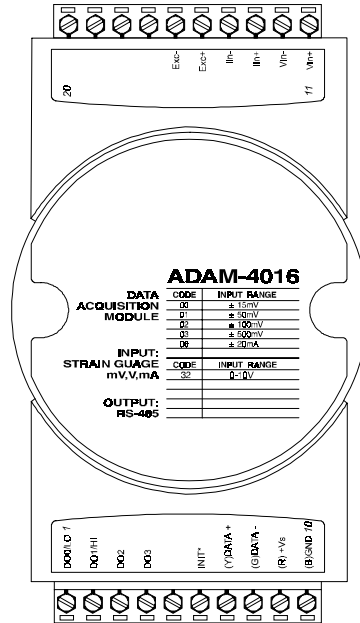


Figure 3-12 ADAM-4016 Analog Input/Output Module

Accepts:

- millivolt inputs: ± 15 mV, ± 50 mV, ± 100 mV, ± 500 mV Strain Gauge
- Current input: ± 20 mA
- Excitation voltage output: 0 ~ 10 V

Four digital output channels are provided.

Depending on the module's configuration setting, it can forward the data to the host computer in one of the following formats:

- engineering units (mV or mA)
- percent of full-scale range (FSR)
- twos complement hexadecimal

Application Wiring

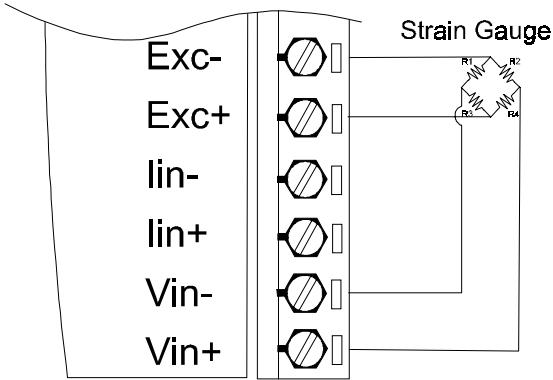


Figure 3-13 Strain Gauge Voltage Input

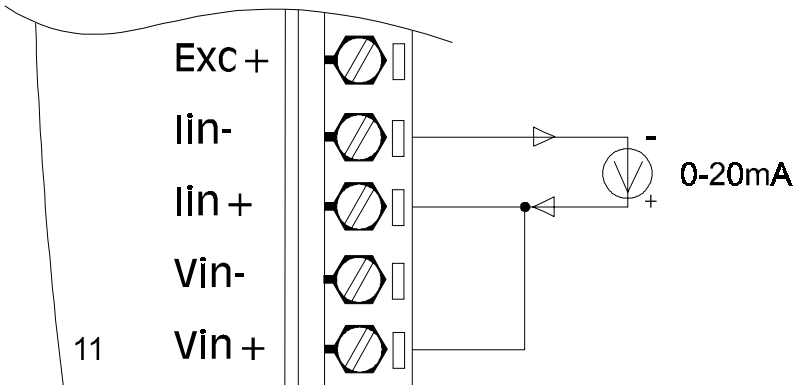


Figure 3-14 Strain Gauge Current Input

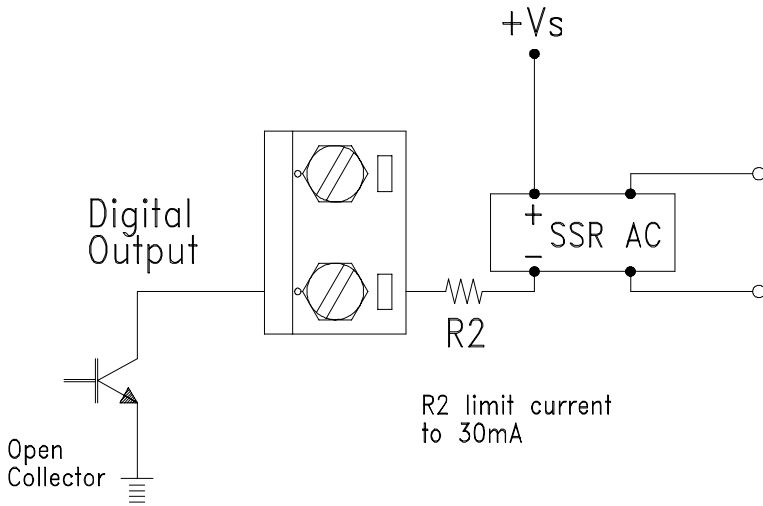


Figure 3-15 Digital Output used with SSR

3.3 ADAM-4017/4017+/4018/4018M/4018+ 8-channel Analog Input Modules

ADAM-4017/4018 8-channel Analog Input Module

The ADAM-4017/4017+/4018/4018+ is a 16-bit, 8-channel analog input module that provides programmable input ranges on all channels. This module is an extremely cost-effective solution for industrial measurement and monitoring applications. Its opto-isolated inputs provide 3000 V_{DC} of isolation between the analog input and the module, protecting the module and peripherals from damage due to high input-line voltages.

ADAM-4017/4017+/4018/4018+ offers signal conditioning, A/D conversion, ranging and RS-485 digital communication functions. The module protects your equipment from ground loops and power surges by providing opto-isolation of A/D input and transformer based isolation up to 3000 V_{DC}.

The ADAM-4017/4017+/4018/4018+ uses a 16-bit microprocessor-controlled sigma-delta A/D converter to convert sensor voltage or current into digital data. The digital data is then translated into engineering units. When prompted by the host computer, the module sends the data to the host through a standard RS-485 interface.

ADAM-4018M 8-channel Analog Input Data logger

The ADAM-4018M is a 16-bit, 8-channel analog input data logger featuring programmable input ranges on all channels. This reliable and easy to use analog input logger can store up to 38,000 measurements for a maximum duration of 20 years. The ADAM-4018M can accept various analog inputs, such as thermocouple, mV, V and mA, and offers three configurable logging modes: standard log, event log, and mixed log. Optically isolated inputs provide 500 V_{DC} of isolation between the module and the analog input, protecting the module and peripherals from damage due to high voltages on the input lines. The ADAM-4018M is an extremely cost-effective solution for industrial measurement and monitoring applications.

ADAM-4017+ 8-channel Differential Analog Input Module

Here comes a solution to the demand for more channels of analog input. Similar to its counterpart, the ADAM-4017+ enables eight differential channels with multiple input ranges. This multi-ch/multi-range structure allows channels with different input ranges at the same time, say channel 1 with the range +/- 5 V meanwhile the others with +/- 10 V and +/- 20 mA.

Instead of leaving two single-ended channels in ADAM-4017 module due to the limit number of pins, the ADAM-4017+ uses a jumper to switch AGND and INIT* to Vin6- and Vin7-, respectively, to allow 8-channel input. Additionally, the ADAM-4017+ has been expanded to accept 4 ~ 20 mA so the user can employ it in various applications

ADAM-4018+ 8-ch. Thermocouple Input Module

Here comes a solution to the demand for more channels of thermocouple input. Similar to its counterpart, the ADAM-4018+ enables eight differential channels with multiple input types. This multi-ch/multi-type structure allows channels with different input types at the same time, say channel 1 with K input type meanwhile the others with R and S types.

ADAM-4018+ is an 8-channel T/C input module. Compared with the universal analog input module ADAM-4019, it is rather dedicated to T/C and 4 ~ 20 mA inputs for those with special request. It improves ADAM-4018, the traditional design six differential and two single-ended channels, to enhance the steadiness and reliability of wiring. Usually, broken external will lead to inaccurate current value. ADAM-4018+ provides burned-out detection so that users can easily troubleshoot broken wiring problems.

ADAM-4017

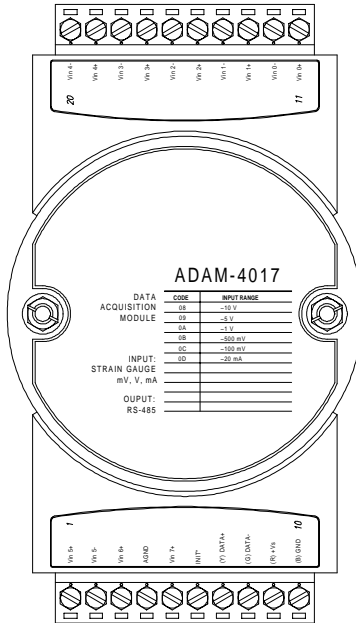


Figure 3-16 ADAM-4017 8-channel Analog Input Module

Channels:

- six differential, two single-ended

Accepts:

- Millivolt inputs: ± 150 mV and ± 500 mV
- Volt inputs: ± 1 V, ± 5 V, and ± 10 V
- Current input: ± 20 mA (requires a 125Ω resistor)

The module forwards the data to the host computer in engineering units (mV, V, or mA)

ADAM-4017+ 8-channel Differential Analog Input Module

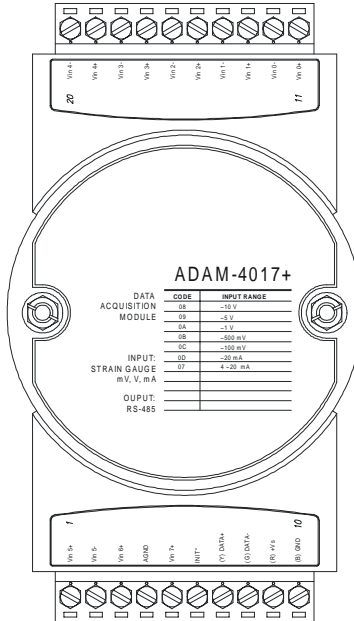


Figure 3-17 ADAM-4017+ 8-ch. differential analog input module

Jumper setting

1. JP9 and JP10 (default setting is six differential and two single-ended)

| | | |
|-----------|--|---------------------------------------|
| JP9, Jp10 | <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | Six differential and two single-ended |
| | <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> | Eight differential |

2. JP1~JP8

| | | | | | | | | |
|--------------------|--|---------------------|------|------|------|------|------|-------------|
| JP1~JP8 | <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | 20mA Input Range | | | | | | |
| | <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> | Voltage Input range | | | | | | |
| Mapping to Channel | Ch.1 | Ch.2 | Ch.3 | Ch.4 | Ch.5 | Ch.6 | Ch.7 | Ch.8 |
| | JP 1 | JP 3 | JP 5 | JP 7 | JP 2 | JP 4 | JP 6 | JP 8 |

3. JP13

| | | |
|------|--|------------------------|
| Jp13 | <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | System is normal mode |
| | <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> | System is initial mode |

Technical specification of ADAM-4017+

| | |
|---|--|
| Channel | 8 |
| Input Type | mV, V, mA |
| Input range | +/- 150 mV, +/- 500 mV, +/- 1 V, +/- 5 V, +/- 10 V, +/- 20 mV, 4 ~ 20 mA |
| Isolation Voltage | 3000 V _{DC} |
| Fault and overvoltage protection | With stands overvoltage up to +/-35 V |
| Sampling Rate | 10 sample/sec (total) |
| Input Impedance | 20 MΩ |
| Accuracy | +/- 0.1% or better |
| Power Consumption | 1.2 W |
| I/O Connector Type | 10-pin plug-terminal |

ADAM-4018

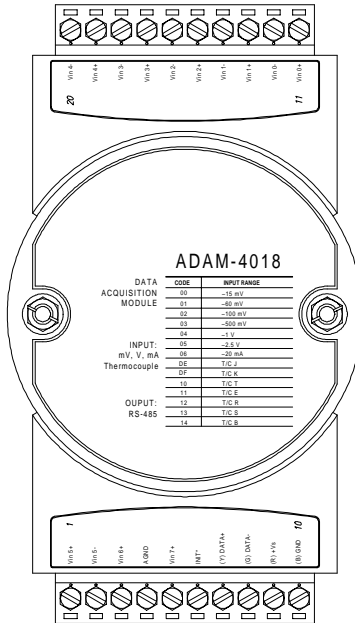


Figure 3-18 ADAM-4018 8-channel Thermocouple Input Module

Channels:

- six differential, two single-ended

Accepts:

- J, K, T, E, R, S and B thermocouples
- Millivolt inputs: ± 15 mV, ± 50 mV, ± 100 mV and ± 500 mV
- Volt inputs: ± 1 V and ± 2.5 V
- Current input: ± 20 mA (requires a 125Ω resistor)

The module forwards the data to the host computer in engineering units ($^{\circ}\text{C}$, mV, V, or mA)

ADAM-4018M

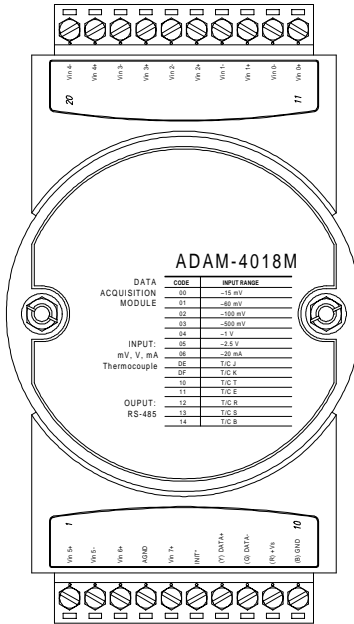


Figure 3-19 ADAM-4018M 8-channel Analog Input Data Logger Channels:

- six differential, two single-ended

Accepts:

- J, K, T, E, R, S and B thermocouples
- Millivolt inputs: ± 15 mV, ± 50 mV, ± 100 mV, ± 500 mV
- Volt inputs: ± 1 V and ± 2.5 V
- Current input: ± 20 mA (requires a 125Ω resistor)

The module forwards the data to the host computer in engineering units ($^{\circ}\text{C}$, mV, V, or mA)

Storage Capacity:

- 128 KB flash memory

Technical specification of ADAM-4018+

| | |
|---|---|
| Channel | 8 |
| Input Type | Thermocouple |
| T/C type and Temperature Range | J 0 ~ 760 C K 0 ~ 1000 C T -100 ~ 400 C E 0 ~ 1000 C R 500 ~ 1750 C S 500 ~ 1750 C B 500 ~ 1800 C |
| Isolation Voltage | 3000 V _{DC} |
| Fault and overvoltage protection | Withstands over voltage up to +/- 35 V |
| Sampling Rate | 10 sample/sec (total) |
| Input Impedance | 20 MΩ |
| Accuracy | +/- 0.1% or better |
| Power Consumption | 0.5 W |
| I/O Connector Type | 10-pin plug-terminal |

<http://www.advantech.com/> ➔ "service & support"
➔ search "ADAM-4018+"

Application Wiring

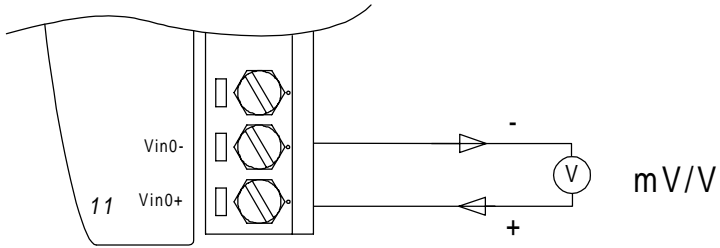


Figure 3-21 Differential Input (CH0 to CH5)

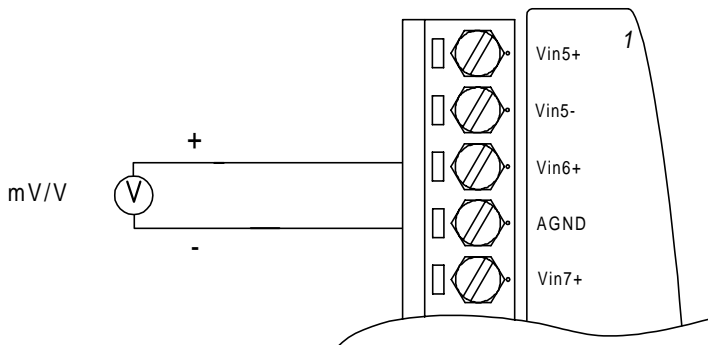
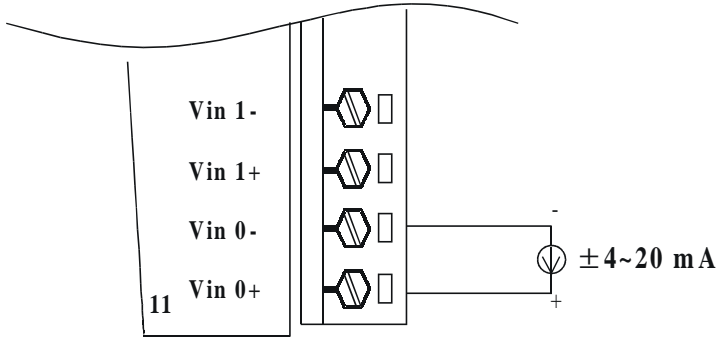
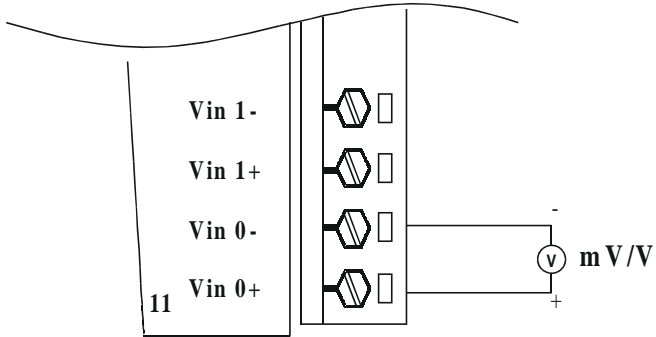


Figure 3-22 Single-ended Input (CH6 to CH7)

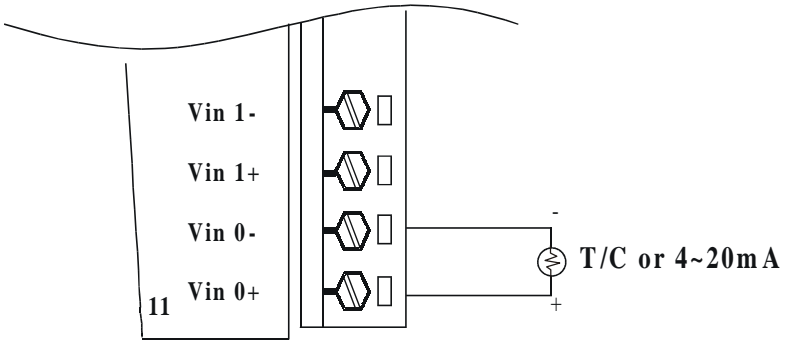
Application Wiring



internal Built 1200hm Resister

ADAM-4017+ 8-ch. differential analog input module wiring diagram

- 1) Link to <http://www.advantech.com>
- 2) Click Support to get in eService Knowledge Center
- 3) Search for download and key-in "ADAM-4000" to get the latest ADAM-4000 User's Manual



ADAM-4018+ 8-ch. thermocouple input module wiring diagram

3.4 ADAM-4019 8-channel Universal Analog Input Module

Here comes good news for users needing various analog input signals. If there are different types of analog input, such as V, mV, mA, or thermocouple signals, users have to prepare individual modules for data acquisition. Now Advantech announces the ADAM-4019 universal analog input module to integrate the various AI modules as one. It not only reduces hardware cost, but also simplifies wiring engineering.

ADAM-4019

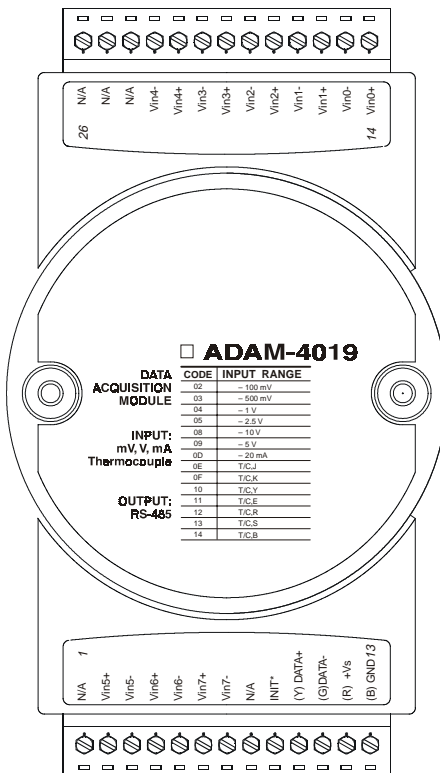


Figure 3-25: ADAM-4019 8-channel Universal AI

Application Wiring

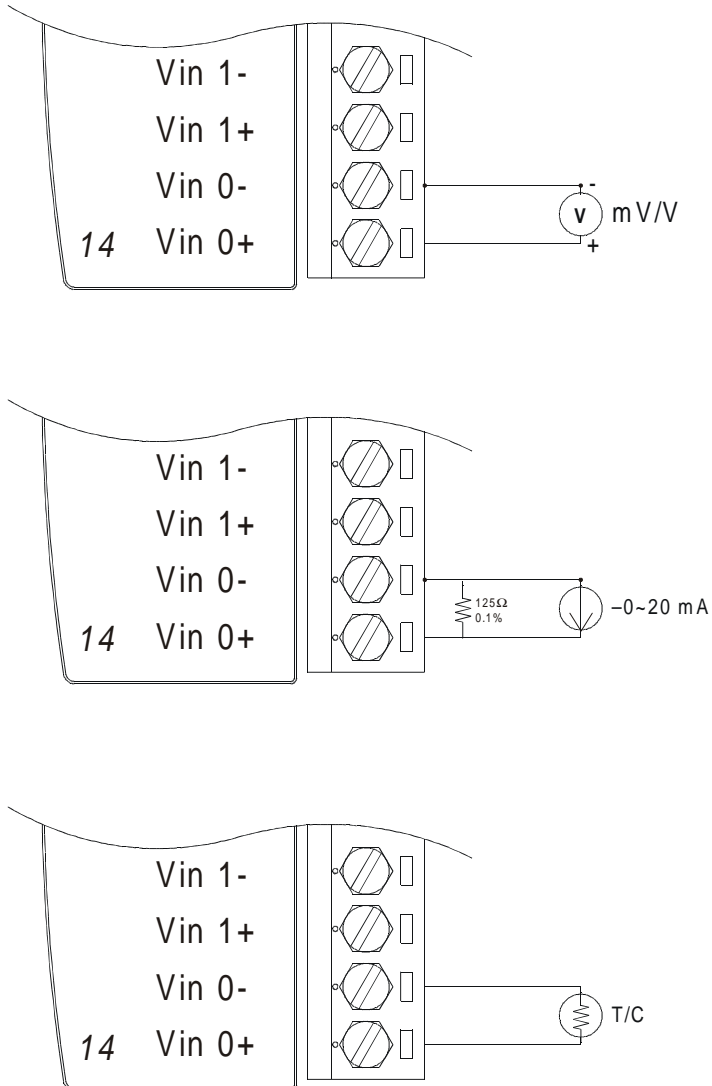


Figure 3-26: ADAM-4019 Universal AI wiring diagram

Technical specification of ADAM-4019

| | |
|---|---|
| Channel | 8 |
| Input Type | V, mV, mA, T/C |
| Input type and temperature range | <p>V: +/-1V , +/-2.5V, +/-5V , +/-10V mV: +/- 100mV , +/-500mV mA: +/-20mA (w/125Ω resistor) Thermocouple: J 0 to 760 C K 0 to 1370 C T -100 to 400 C E 0 to 1400 C R 500 to 1750 C S 500 to 1750 C B 500 to 1800 C</p> |
| Isolation Voltage | 3000 V _{DC} |
| Sampling Rate | 6 sample/sec (total) |
| Input Impedance | 20 MW |
| Accuracy | +/- 0.1% or better |
| Power Consumption | 1 W |
| I/O Connector Type | 13-pin plug-terminal |

Table 3-2: Technical specification of ADAM-4019

3.5 ADAM-4021 Analog Output Module

Analog output module receives their digital input through an RS-485 interface from the host computer. The format of the data is either engineering units, two's complement hexadecimal format or percentage of full-scale range (FSR), depending on the module's configuration. It then uses its microprocessor-controlled D/A converter to convert the digital data into output signals.

You get a true readback of the analog output signal from the unit's ADC, which independently monitors the output. You can specify slew rates and start up currents through the configuration software. The Analog Output Module can supply single-channel analog output in a range of voltages or currents.

They protect your equipment from ground loops and power surges by providing opto-isolation of the D/A output and transformer based isolation up to $3000 V_{DC}$.

Slew Rate

The slew rate is defined as the discrepancy between the number of milli-amps (or Volts) per second of the present and the required output currents (or voltages). An ADAM analog output module may be configured for a specific slew rate.

ADAM-4021

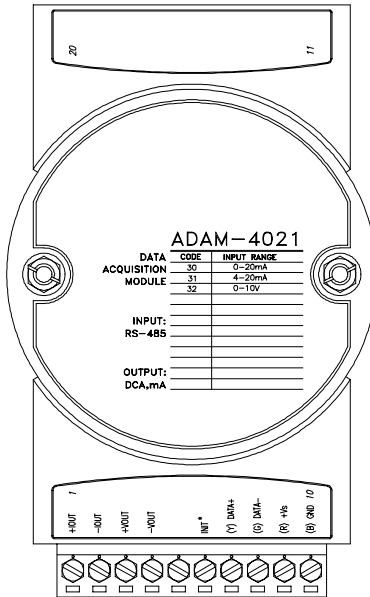


Figure 3-27 ADAM -4021 Analog Output Module

Depending on its configuration settings the module accepts the following formats from the host computer:

- Engineering units
- Percent of full-scale range (FSR)
- Twos complement hexadecimal format,

Output types:

- Voltage: 0 to 10 V
(Slew rate: 0.0625 to 64 V/sec)
- Currents: 0 to 20 mA, or 4 to 20 mA.
(Slew rate: 0.125 to 128 mA/sec)

Application Wiring

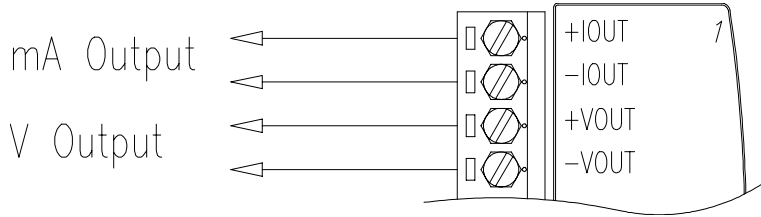


Figure 3-28 Analog Output

3.6 ADAM-4050/4051/4052/4053/4055 Digital I/O Modules

ADAM-4050 Digital I/O Module

The ADAM-4050 features seven digital input channels and eight digital output channels. The outputs are open-collector transistor switches that you can control from the host computer. You can also use the switches to control solid-state relays, which in turn can control heaters, pumps and power equipment. The host computer can use the module's digital inputs to determine the state of limit or safety switches or remote digital signals.

ADAM-4051 16-channel Isolated Digital Input Module

The ADAM-4051 is a 16-ch. Digital Input Module built with $2500V_{DC}$ optical isolation, it is suitable to critical applications. Different from other modules, the ADAM-4051 accept 10 ~ 50V input voltage to fit various digital signals, such as 12 V, 24 V, 48 V. Moreover, users can read the current status from the LED indicators on the front panel.

ADAM-4052 Isolated Digital Input Module

The ADAM-4052 provides eight digital input channels: six fully independent isolated channels and two isolated channels with a common ground. All have $5000 V_{RMS}$ isolation to prevent ground loop effects and prevent damage from power surges on the input lines.

ADAM-4053 16-channel Digital Input Module

The ADAM-4053 provides 16 digital input channels for dry contact or wet contact signals. For dry contact, effective distance from DI to contact point is up to 500 m.

ADAM-4055 16-channel Isolated Digital I/O Module

The ADAM-4055 offers 8-ch. isolated digital input and 8-ch. isolated digital output for critical applications. The inputs accept 10~50V voltage, and the outputs supply $5\sim40V_{DC}$ open collector. Considered to user friendly, the ADAM-4055 also built with LED indicator for status reading easily.

ADAM-4051

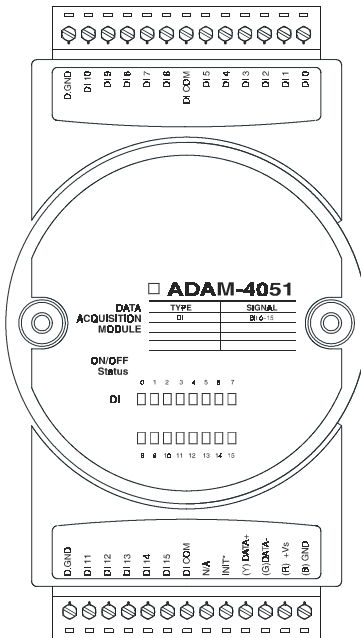


Figure 3-32 ADAM-4051 16-channel Digital Input Module

Technical Specification of ADAM-4051

- **Channel:** 16 (4-channel/group)
- **Optical Isolation:** 2500 V_{DC}
- **Opto-isolator response time:** 25 Ωs
- **Over-voltage Protect:** 70 V_{DC}
- **ESD (Electro Static Discharge) :** 2000 V_{DC}
- **LED Indicator :** On: Active; Off: Non-active
- **Input Voltage:** Logic level 1: 10 ~ 50 V
Logic level 0: 3 V
- **Poer consumption:** 1W
- **I/O Connector Type:** 13-pin plug-terminal*2

ADAM-4053

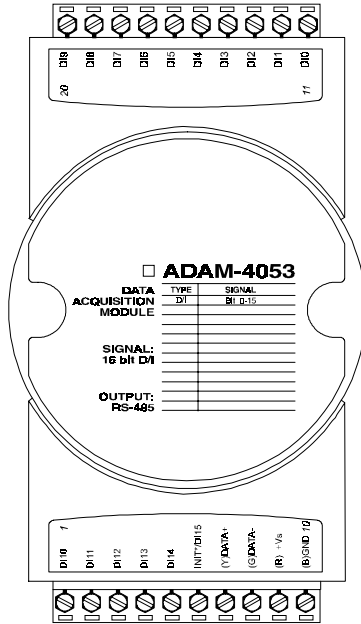


Figure 3-34 ADAM-4053 16-channel Digital Input Module

Channels: 16

Digital Input:

-Dry contact

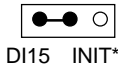
logic level 0: Close to GND

logic level 1: OPEN

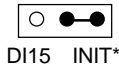
-Wet contact

logic level 0: +2 V max.

logic level 1: +4 V to +30 V



DI15



INIT*

ADAM-4055

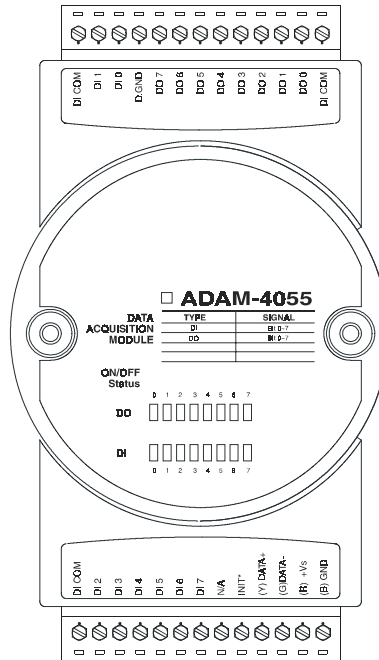


Figure 3-34 ADAM-4055 16-channel Digital I/O Module

- **Number of Input Channel** : 8 (4-channel/group)
- **Optical Isolation** : 2500 V_{DC}
- **Opto-isolator response time** : 25 μs
- **Over-voltage Protect**: 70V_{DC}
- **ESD (Electro Static Discharge)**: 2000 V_{DC}
- **Input Voltage:**
 - Dry Contact Logic level 0: Close to GND
 - Logic level 1: Open
 - Wet Contact Logic level 1: 10 ~ 50 V
 - Logic level 0: 3 V
 - Dry Contact & Wet contact Selectable

I/O Modules

- **Number of Output Channel:** 8 (8-channel/group)
- **Optical Isolation:** 2500 V_{DC}
- **Opto-isolator response time:** 25 μs
- **Supply Voltage:** Open Collector 5 ~ 40 V_{DC}
- **Sink Current:** 200 mA max/channel
- **LED Indicator:** On: Active Off: Non-active
- **Power Consumption:** 1 W
- **I/O Connector Type:** 13-pin plug-terminal * 2

Application Wiring

The following give you examples of how to connect various types of digital I/O applications to your ADAM modules.

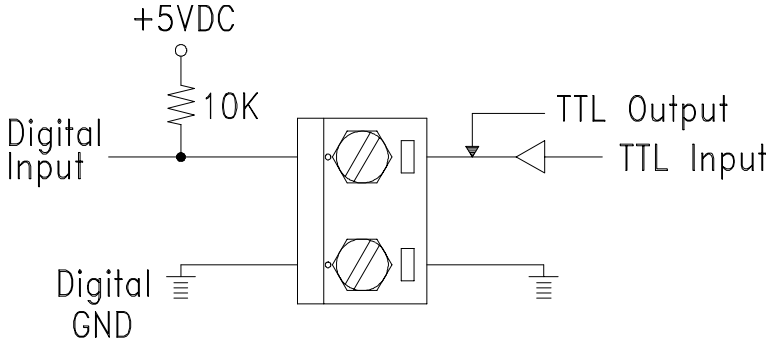


Figure 3-35 TTL Input (ADAM-4050)

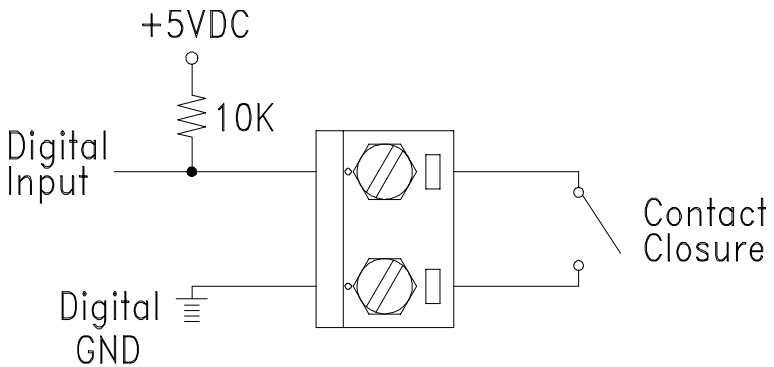


Figure 3-36 Contact Closure Input (ADAM-4050)

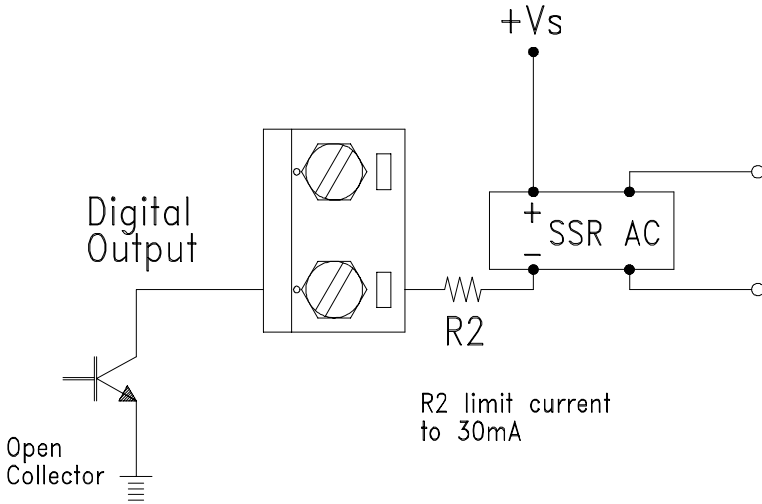


Figure 3-37 Digital Output used with SSR (ADAM-4050)

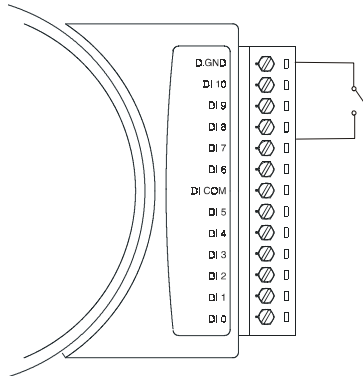


Figure 3-38 Dry Contact Wiring (ADAM-4051)

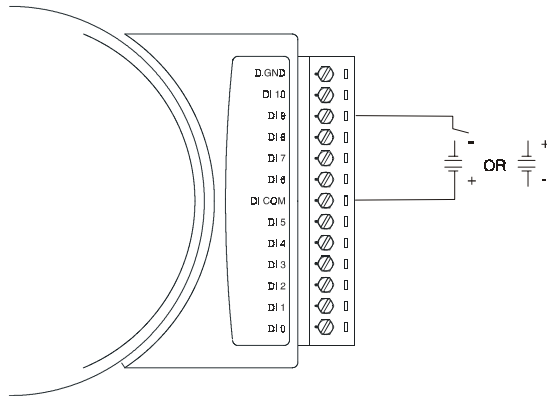


Figure 3-39 Wet Contact Wiring (ADAM-4051)

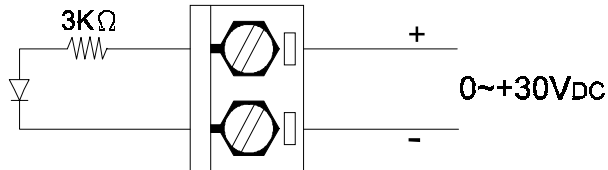


Figure 3-40 Isolation Digital Input (ADAM-4052)

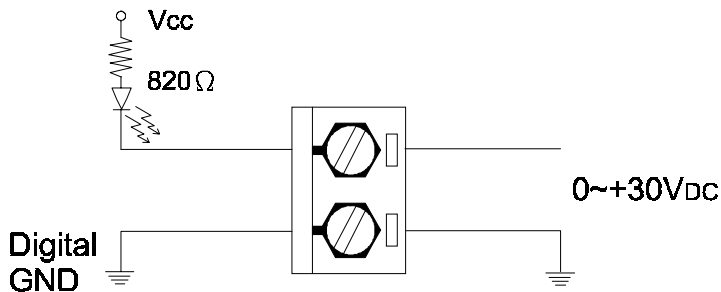


Figure 3-41 Wet Contact Input (ADAM-4053)

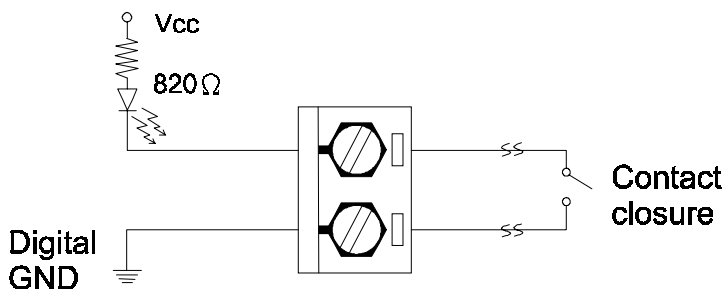


Figure 3-42 Contact Closure Input (ADAM-4053)

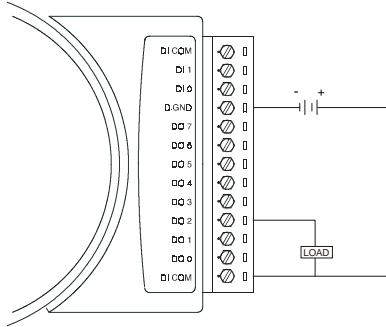


Figure 3-43 Digital Output wiring (ADAM-4055)

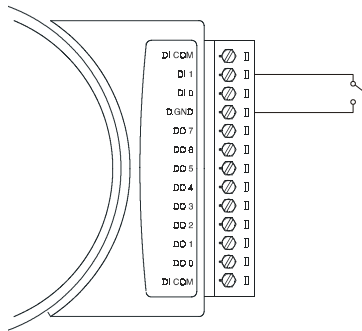


Figure 3-44 Digital Input Dry Contact Wiring (ADAM-4055)

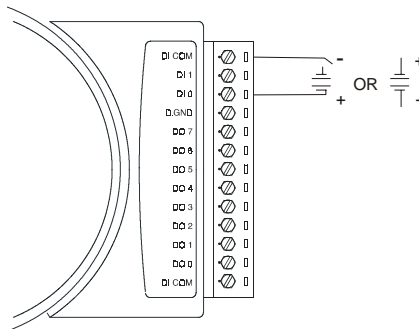


Figure 3-45 Digital Input Wet Contact Wiring (ADAM-4055)

3.7 ADAM-4060/4068 Relay Output Module

The ADAM Relay Output Module is a low-cost alternative to SSR modules. The ADAM-4060 Relay Output Module provides four relay channels, two of Form A and two of Form C. The ADAM-4068 Relay Output Module provides eight channels, four of Form A and four of Form C. These two modules are excellent for ON/OFF control or low-power switching applications.

ADAM-4060

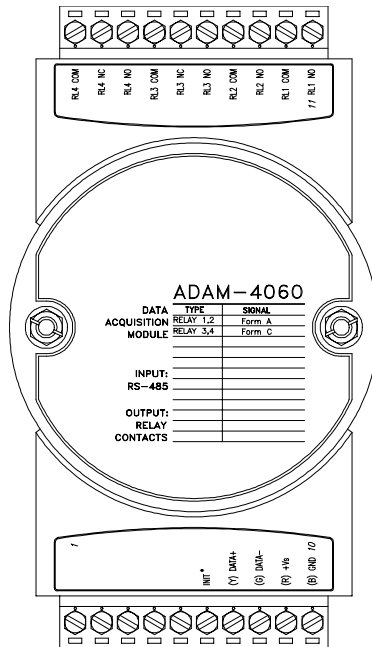


Figure 3-46 ADAM-4060 Relay Output Module

Contact rating for Form A and Form C:

AC: 0.5 A / 120 V_{AC}

DC: 1 A / 24 V_{DC}

ADAM-4068

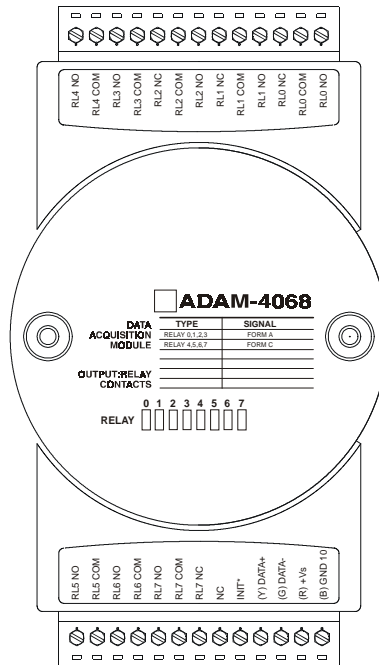


Figure 3-47 ADAM-4068 8-channel Relay Output Module

Contact Rating :

AC 125 V @ 0.6 A; 250 V @ 0.3 A

DC 30 V @ 2 A; 110 V @ 0.6 A

Application Wiring

The following gives you examples on how to connect form A and form C relay output applications to your ADAM modules.

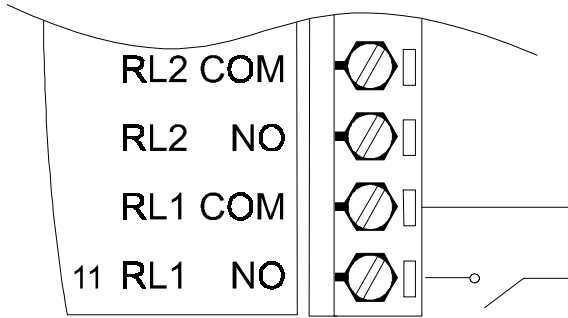


Figure 3-48 Form A relay output (ADAM-4060)

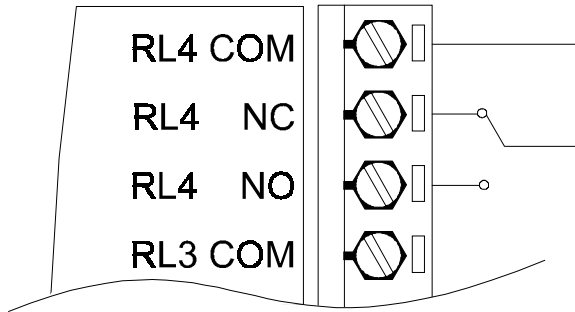


Figure 3-49 Form C relay output (ADAM-4060)

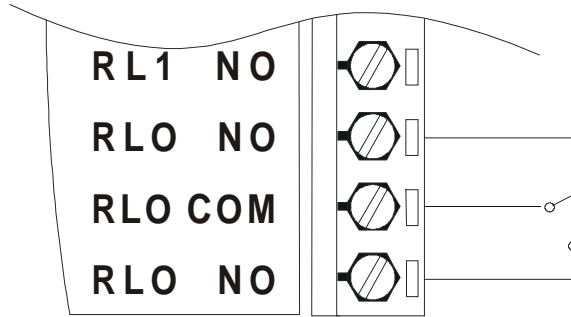


Figure 3-50 Form C relay output (ADAM-4068)

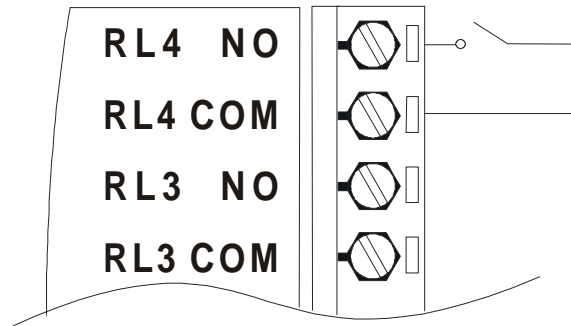


Figure 3-51 Form A relay output (ADAM-4068)

3.8 ADAM-4080/4080D Counter/Frequency Input Modules

ADAM-4080/4080D Counter/frequency input module has two 32-bit counter input channels (counter 0 and counter 1) with built-in programmable timer for frequency measurement. These cost-effective modules let you monitor counter/frequency data as measurements are taken.

Front Panel LED Indicator (ADAM-4080D only)

The 5-digit LED display of the ADAM-4080D lets you monitor its counter data right at the source. The module's LED display shows counter values as well as high-low alarm messages as a counter exceeds its programmed limits (direct display).

Another option is to display data send by the host computer. The module first sends counter data to the host computer for conversion or calculation. The host computer sends the data back to the ADAM-4080D and the module shows it on the module's LED display (remote display).

Signal Input Mode

The ADAM-4080/4080D provides separate terminals for photo isolated input and non-isolated input to simplify wiring. Opto-isolated input provides 2500 V_{RMS} isolation to protect your equipment from ground loops. After you make the physical connections, program the module to identify which of its two sets of input terminals it should activate (isolated or non-isolated terminals).

Programmable Digital Filter

The ADAM-4080/4080D module includes a unique programmable digital filter to reject noise on the input lines. You can specify separate time constants, such as minimum signal width at high level and minimum signal width at low level, to provide stable output readings.

Programmable Threshold

When the ADAM-4080/4080D is programmed for non-isolated input you can set a high and low trigger level. Like the programmable digital filter, the programmable threshold rejects noise on the input lines and provides stable input readings

External Control (Gate mode)

Besides the GND and counter terminal each channel has a gate terminal to connect an external gate signal. The gate signal (high or low) can trigger the counter to start or stop counting. The gate mode can be either low, high or disabled (low means that counting starts when the gate signal is low and stops when the gate signal becomes high)

Programmable Alarm Output

The ADAM-4080 module provides a configurable alarm for each counter. The ADAM-4080D provides high and low alarm functions for counter 0. When the counter reaches an alarm limit, it will trigger the built-in digital output for machine ON/OFF control. The alarm limits may be downloaded into the module's EEPROM by the host computer. The initial count value of ADAM-4080 module's counter can be configured as any values.

The alarm functions can be enabled or disabled remotely. When the alarm functions are enabled, digital output channels are used to indicate the alarm states. For ADAM-4080, digital output channel 0 equals the alarm state of counter 0, and digital output channel 1 equals the alarm state of counter 1. For ADAM-4080D, digital output channel 0 equals the low alarm state of the counter 0 and digital output channel 1 equals the high alarm state of the counter 0.

Every A/D conversion will be followed by a comparison with the alarm limits. If the input value exceeds one of these limits, the corresponding alarm state is set to ON.

There are two alarm mode options for the ADAM-4080D: momentary and latching. If the alarm is in latching mode, the alarm will stay on even when the input value returns within limits. An alarm in latching mode can be turned off by issuing a Clear Alarm command from the host computer. A latching alarm is cleared by the module when the opposite alarm is set. For example: the alarm is in latching mode and the high alarm is turned ON. When the module receives a value that is lower than the low alarm limit, it will clear the high alarm and turn the low alarm ON.

When the alarm is in Momentary mode, the alarm will be turned OFF as soon as the input value returns to within limits.

The arrangement of coupling high and low alarm states with digital output lines may be utilized to build ON/OFF controllers that can operate without host computer involvement.

ADAM-4080

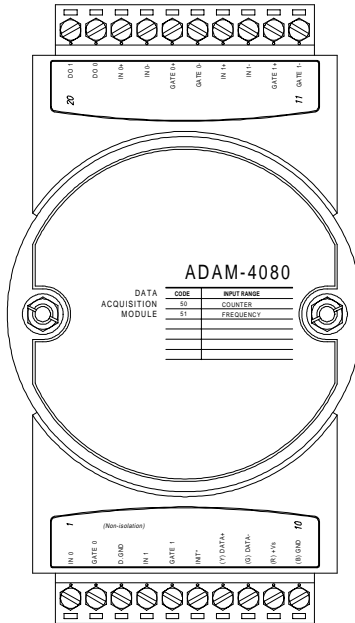


Figure 3-52 ADAM-4080 Counter/Frequency Input Module

Channels: Two independent 32-bit counters (counter 0 and counter 1)

Input frequency: 50 kHz max.

Input mode: Isolated or non-isolated

Isolation input level:

- Logic level 0: +1 V max
- Logic level 1: +3.5 V to +30 V

Non-isolation input level (programmable threshold):

- Logic level 0: 0 to +5V (default=0.8 V)
- Logic level 1: 0 to +5V (default = 2.4 V)

ADAM-4080D

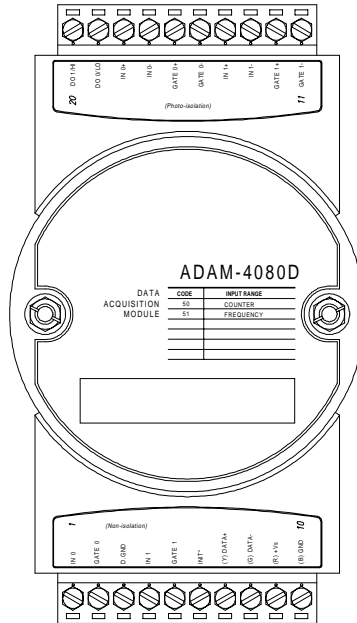


Figure 3-53 ADAM-4080D Counter/Frequency Input Module with LED Display

Channels: Two independent 32-bit counters (counter 0 and counter 1)

Input frequency: 50 kHz max.

Input mode: Isolated or non-isolated

Isolation input level:

- Logic level 0: +1 V max

- Logic level 1: +3.5 V to +30 V

Non-isolation input level (programmable threshold):

- Logic level 0: 0 to +5V (default=0.8 V)

- Logic level 1: 0 to +5V (default = 2.4 V)

Application Wiring

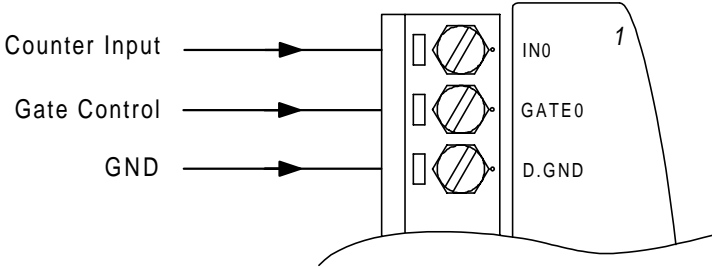


Figure 3-54 *Non-isolated Input*

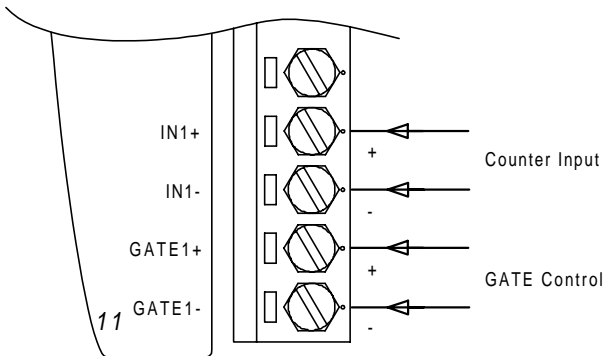


Figure 3-55 *Photo-isolated Input*

Command Set

4

4.1 Introduction

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

When modules are not transmitting they are in listen mode. The host issues a command to a module with a specified address and waits a certain amount of time for the module to respond. If no response arrives, a timeout aborts the sequence and returns control to the host.

Changing ADAM's configuration might require the module to perform auto calibration before changes can take effect. Especially when changing the range, the module has to perform all stages of auto calibration that it also performs when booted. When this process is under way, the module does not respond to any other commands. The command set includes the exact delays that might occur when modules are reconfigured.

4.2 Syntax

[delimiter character][address][command][data][checksum] [carriage return]

Every command begins with a delimiter character. There are four valid characters: a dollar sign \$, a pound sign #, a percentage sign % and an at sign @.

The delimiter character is followed by a two-character address (hexadecimal) that specifies the target module. The actual two character command follows the address. Depending on the command, an optional data segment follows the command string. An optional two character checksum may be appended to the total string. Every commands is terminated by a carriage return (cr).

ALL COMMANDS SHOULD BE ISSUED IN UPPERCASE CHARACTERS!

Before the command set, we provide the I/O module commands search table to help you find the commands you wish to use. The command set is divided into the following four subsections:

- Analog Input Module commands
- Analog Output Module commands
- Digital I/O and Relay Output Module commands
- Counter/Frequency Module commands

Every subsection 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 sometimes share the same format, the effect they have on a certain module can be completely different than they have on another. For example, the configuration command: %AANNTCCFF affects analog input modules and analog output modules differently. Therefore, the full command set for every module is listed.

4.3 I/O Module Commands Search Table

ADAM-4011 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|---|----------|
| %AANNTCCFF | Configuration | Sets the address, input range, baud rate, data format, checksum status, and/or integration time for a specified analog input module | 4 44 |
| #AA | Analog Data In | Returns the input value from a specified analog input module in the currently configured data format | 4 52 |
| \$AA0 | Span Calibration | Calibrates an analog input module to correct for gain errors | 4 57 |
| \$AA1 | Offset Calibration | Calibrates an analog input module to correct for offset errors | 4 58 |
| *** | Synchronized Sampling | Orders all analog input modules to sample their input values and store them in special registers | 4 59 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified module's register after the *** command | 4 60 |
| \$AA2 | Configuration Status | Returns the configuration parameters for the specified analog input module | 4 48 |
| \$AA3 | CJC Status | Returns the value of the CJC sensor for a specified analog input module | 4 63 |
| \$AA9 | CJC Offset Calibration | Calibrates the CJC sensor for offset errors | 4 64 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified analog input module | 4 50 |
| \$AAM | Read Module Name | Return the module name from the specified analog input module | 4 51 |

(continued on following page)

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|-----------------------------------|--|----------|
| @AAD1 | Read Digital I/O and Alarm Status | The addressed module returns the state of its digital input channel, its two digital output channels and the status of its alarm | 4 104 |
| @AAD0(data) | Set Digital Output | Set the values of the module's two digital outputs (ON or OFF) | 4 106 |
| @AAEAT | Enable Alarm | Enables the alarm in either Momentary or Latching mode | 4 108 |
| @AAHI(data) | Set High Alarm | Downloads the High alarm limit value | 4 110 |
| @AALO(data) | Set Low Alarm | Downloads the Low alarm limit value | 4 111 |
| @AADA | Disable Alarm | Disables all alarm functions | 4 112 |
| @AACA | Clear Latch Alarm | The latch alarm is reset | 4 113 |
| @AARH | Read High Alarm | The addressed analog input module is asked to return its high alarm value | 4 114 |
| @AARL | Read Low Alarm | The addressed analog input module is asked to return its low alarm value | 4 115 |
| @AARE | Read Event Counter | The addressed module is asked to return its event counter value | 4 116 |
| @AACE | Clear Event Counter | The event counter is set to 0 | 4 117 |

ADAM-4011D Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|-----------------------------|---|----------|
| %AANNTCCFF | Configuration | Sets the address, input range, baud rate, data format, checksum status, and/or integration time for a specified analog input module | 4 44 |
| #AA | Analog Data In | Returns the input value from a specified analog input module in the currently configured data format | 4 52 |
| \$AA0 | Span Calibration | Calibrates an analog input module to correct for gain errors | 4 57 |
| \$AA1 | Offset Calibration | Calibrates an analog input module to correct for offset errors | 4 58 |
| #** | Synchronized Sampling | Orders all analog input modules to sample their input values and store them in special registers | 4 59 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified module's register after the #** command | 4 60 |
| \$AA2 | Configuration Status | Returns the configuration parameters for the specified analog input module | 4 48 |
| \$AA3 | CJC Status | Returns the value of the CJC sensor for a specified analog input module | 4 63 |
| \$AA9 | CJC Offset Calibration | Calibrates the CJC sensor for offset errors | 4 64 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified analog input module | 4 50 |
| \$AAM | Read Module Name | Return the module name from the specified analog input module | 4 51 |
| \$AAB | Open Thermocouple Detection | Ask the module to respond whether the thermocouple is open or not | 4 62 |

(continued on following page)

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|-----------------------------------|--|----------|
| @AADI | Read Digital I/O and Alarm Status | The addressed module returns the state of its digital input channel, its two digital output channels and the status of its alarm | 4 104 |
| @AADO(data) | Set Digital Output | Set the values of the module's two digital outputs (ON or OFF) | 4 106 |
| @AAEAT | Enable Alarm | Enables the alarm in either Momentary or Latching mode | 4 108 |
| @AAHI(data) | Set High Alarm | Downloads the High alarm limit value | 4 110 |
| @AALO(data) | Set Low Alarm | Downloads the Low alarm limit value | 4 111 |
| @AADA | Disable Alarm | Disables all alarm functions | 4 112 |
| @AACA | Clear Latch Alarm | The latch alarm is reset | 4 113 |
| @AARH | Read High Alarm | The addressed analog input module is asked to return its high alarm value | 4 114 |
| @AARL | Read Low Alarm | The addressed analog input module is asked to return its low alarm value | 4 115 |
| @AARE | Read Event Counter | The addressed module is asked to return its event counter value | 4 116 |
| @AACE | Clear Event Counter | The event counter is set to 0 | 4 117 |

ADAM-4012 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|---|----------|
| %AANN TTCCFF | Configuration | Sets the address, input range, baud rate, data format, checksum status, and/or integration time for a specified analog input module | 4 44 |
| #AA | Analog Data In | Returns the input value from a specified analog input module in the currently configured data format | 4 52 |
| \$AA0 | Span Calibration | Calibrates an analog input module to correct for gain errors | 4 57 |
| \$AA1 | Offset Calibration | Calibrates an analog input module to correct for offset errors | 4 58 |
| *** | Synchronized Sampling | Orders all analog input modules to sample their input values and store them in special registers | 4 59 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified module's register after the *** command | 4 60 |
| \$AA2 | Configuration Status | Returns the configuration parameters for the specified analog input module | 4 48 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified analog input module | 4 50 |
| \$AAM | Read Module Name | Return the module name from the specified analog input module | 4 51 |

(continued on following page)

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|-----------------------------------|--|----------|
| @AADI | Read Digital I/O and Alarm Status | The addressed module returns the state of its digital input channel, its two digital output channels and the status of its alarm | 4 104 |
| @AADO(data) | Set Digital Output | Set the values of the module's two digital outputs (ON or OFF) | 4 106 |
| @AAEAT | Enable Alarm | Enables the alarm in either Momentary or Latching mode | 4 108 |
| @AAHI(data) | Set High Alarm | Downloads the High alarm limit value | 4 110 |
| @AALO(data) | Set Low Alarm | Downloads the Low alarm limit value | 4 111 |
| @AADA | Disable Alarm | Disables all alarm functions | 4 112 |
| @AACA | Clear Latch Alarm | The latch alarm is reset | 4 113 |
| @AARH | Read High Alarm | The addressed analog input module is asked to return its high alarm value | 4 114 |
| @AARL | Read Low Alarm | The addressed analog input module is asked to return its low alarm value | 4 115 |
| @AARE | Read Event Counter | The addressed module is asked to return its event counter value | 4 116 |
| @AACE | Clear Event Counter | The event counter is set to 0 | 4 117 |

ADAM-4013 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|--|----------|
| %AANNTCCFF | Configuration | Sets the address, baud rate, data format, checksum status, and/or integration time for a specified analog input module | 4-44 |
| #AA | Analog Data In | Returns the input value from a specified analog input module in the currently configured data format | 4-52 |
| \$AA0 | Span Calibration | Calibrates an analog input module to correct for gain errors | 4-57 |
| \$AA1 | Offset Calibration | Calibrates an analog input module to correct for offset errors | 4-58 |
| *** | Synchronized Sampling | Orders all analog input modules to sample their input values and store them in special registers | 4-59 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified module's register after the *** command | 4-60 |
| \$AA2 | Configuration Status | Returns the configuration parameters for the specified analog input module | 4-48 |
| \$AAF | Read Firmware Version | Returns the firmware version code from the specified analog input module | 4-50 |
| \$AAM | Read Module Name | Returns the module name from the specified analog input module | 4-51 |

ADAM-4015 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|--|--|----------|
| %AANNTTCFF | Configuration | Sets the address, baud rate, data format, checksum status, and/or integration time for a specified analog input module | 4-44 |
| #AAN | Read Analog Input from Channel N | Returns the input value from a specified channel of analog input module in the currently configured data format | 4-54 |
| #AA | Analog Data In | Returns the input value from a specified analog input module in the currently configured data format | 4-52 |
| \$AA0Ci | Single Channel Span Calibration | Calibrates a specified channel to correct for gain errors | 4-65 |
| \$AA1Ci | Single Channel Offset Calibration | Calibrates a specified channel to correct for offset errors | 4-66 |
| \$AA2 | Configuration Status | Returns the configuration parameters for the specified analog input module | 4-58 |
| ##** | Synchronized Sampling | Orders all analog input modules to sample their input values and store them in special registers | 4-59 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified module's register after the ##** command | 4-60 |
| \$AA5VV | Enable/Disable Channels for Multiplexing | Enable or disable the individual channels in an analog module | 4-55 |
| \$AA6 | Read Channel Status | Get the enable/disable status of all channels in an analog module | 4-56 |
| \$AAB | Channel Diagnose | Diagnose channel status in over range, under range, and wire opening | 4-62 |
| \$AA7CiRrr | Single Channel Range Configuration | Configure the input type and range of the specified channel in an analog input module | 4-67 |
| \$AA8Ci | Read Single Channel Range Configuration | Get the input type and range of the specified channel in an analog input module | 4-70 |
| \$AAXnnnn | Watchdog Timer Setting | Set WDT communication cycle | 4-71 |
| \$AAY | Read Watchdog Timer Setting | Read the setting of WDT communication cycle | 4-72 |
| \$AAS0 | Internal Calibration | Internal self-calibration for offset and gain errors | 4-73 |
| \$AAS1 | Reload default calibrating parameter | Reload factory default calibrating parameter to overwrite current calibrating parameter | 4-74 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified analog input module | 4-50 |
| \$AAM | Read Module Name | Return the module name from the specified analog input module | 4-51 |

ADAM-4016 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|---|----------|
| %AANNNTCCFF | Configuration | Set the address, input range, baud rate, data format, checksum status and/or integration time for the specified analog input module | 4 44 |
| #AA | Read Analog Input | Return the input value from the specified analog input module in the currently configured data format | 4 52 |
| \$AA0 | Span Calibration | Calibrate an analog input module to correct for gain errors | 4 57 |
| \$AA1 | Offset Calibration | Calibrate an analog input module to correct for offset errors | 4 58 |
| \$AA2 | Configuration Status | Return the configuration parameters for the specified analog input module | 4 48 |
| ##* | Synchronized Sampling | Orders all analog input modules to sample their input values and store them in special registers | 4 59 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified module's register after the ##* command | 4 60 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified analog input module | 4 50 |
| \$AAM | Read Module Name | Return the module name from the specified analog input module | 4 51 |

(continued on following page)

| Command Syntax | Command Name | Command Description | Page No. |
|-----------------------|-----------------------------------|--|-----------------|
| @AAID | Read Digital I/O And Alarm Status | Ask the addressed module to return the state of its four digital output channels and the status of its alarm | 4 104 |
| @AADO(data) | Set Digital Output Values | Set the values of the module's four digital outputs (ON or OFF) | 4 106 |
| @AAEAT | Enable Alarm | Enable the alarm in either momentary or latching mode | 4 108 |
| @AAHI(data) | Set High Alarm Value | Download the high alarm limit value | 4 110 |
| @AALO(data) | Set Low Alarm Value | Download the low alarm limit value | 4 111 |
| @AADA | Disable Alarm | Disable all alarm functions | 4 112 |
| @AACA | Clear Latch Alarm | Reset the module's latch alarm to zero | 4 113 |
| @AARH | Read High Alarm Value | Ask the addressed module to return its high alarm value | 4 114 |
| @AARL | Read Low Alarm Value | Ask the addressed module to return its low alarm value | 4 115 |

(continued on following page)

Command Set

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|---------------------------------------|---|----------|
| \$AA6 | Get Excitation Voltage Output Value | Returns either last value sent to specified module by \$AA7 command, or start-up output voltage | 4 120 |
| \$AA7 | Excitation Voltage Output | Direct output excitation voltage data to a specified module | 4 121 |
| \$AAS | Start-up Voltage Output Configuration | Stores a default value in a specified module. The output value will take effect upon startup. | 4 122 |
| \$AAE | Trim Calibration | Trims specified module a number of units up or down | 4 124 |
| \$AAA | Zero Calibration | Tells the module to store parameters for zero calibration | 4 125 |
| \$AAB | Span Calibration | Tells the module to store parameters for span calibration | 4 126 |

ADAM-4017 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|--|---|----------|
| %AANNTTCCFF | Configuration | Set the address, input range, baud rate, data format, checksum status and/or integration time for the specified analog input module | 4 44 |
| #AAN | Read Analog Input from Channel N | Return the input value from channels number N of the specified analog input module | 4 54 |
| #AA | Read Analog Input from all Channels | Return the input values from all channels of the specified analog input module | 4-52 |
| \$AA0 | Span Calibration | Calibrate the analog input module to correct for gain errors | 4 57 |
| \$AA1 | Offset Calibration | Calibrate the analog input module to correct for offset errors | 4 58 |
| \$AA2 | Configuration Status | Return the configuration parameters for the specified analog input module | 4 48 |
| \$AA5VV | Enable/disable Channels for Multiplexing | Enables/disables multiplexing simultaneously for separate channels of the specified input module | 4 55 |
| \$AA6 | Read Channel Status | Ask the specified input module to return the status of all eight channels | 4 56 |
| \$AAF | Read Version | Return the firmware version code from the specified analog input module | 4 50 |
| \$AAM | Read Module Name | Return the module name from the specified analog input module | 4 51 |

ADAM-4017+ Command Set

| Command Syntax | Command Response Syntax | Command Description | Command Example | Command response Example |
|---|--|--|------------------|---|
| %AANNTTCFF(cr) TT : All channel input range code. CC : Baud rate code, 03 : 1200 04 : 2400 05 : 4800 06 : 9600 07 : 19200 08 : 38400 09 : 57600 0A : 115200 FF : reference command description | !AA (cr) | TT : Initial Value FF (not use) If change FF to XX , All channel input range be seted to XX XX : +/-10V : 0x08 +/-5V : 0x09 +/-1V : 0x0a +/-500mV : 0x0b +/-150mV : 0x0c +/-0~20mA: 0x0d +/-4~20mA: 0x07 FF : setting 7 6 X X X X 1 0 bit 7 : Integration time 0 : 50ms(60HZ) 1 : 60ms(50HZ) bit 6 : checksum 0 : disable 1 : enable bit 1,0 : Data Format 00 : Engineer unit | %0105FF06-00(cr) | !05(cr) |
| #AAN(cr)N: 0 ~ 7 | >(data)(cr)data : snnnnnn s : + or -n : digit or dot | Read Analog Input from Channel N (N from 0 - 7) When over range the (data) will show>+999999(cr) When under range the (data) will show>-000000(cr) | #035(cr) | >+09.789(cr) |
| #AA(cr) | >(data0)(data1)(data2)(data3)(data4)(data5)(cr)data 0 ~ 5: snnnnnn s : + or -n : digit or dot | Read Analog Input (for all 8 ch) When over range the (data n) will show>+999999(cr) When under range the (data n) will show>-000000(cr) | #03(cr) | >+05.000-04.9-00 +100.23-089.32+09.123+123.45-09.13-4(cr) (channel 5 be disable) |
| \$AA0Ci(cr) i: 0 ~ 7 channel number | !AA(cr) | Gain error calibrate of assignment channel, User have to input the external standard source for calibrating process | \$050C7(cr) | !05(cr) |

| Command Syntax | Command Response Syntax | Command Description | Command Example | Command response Example |
|--|--------------------------------|---|---|--------------------------|
| \$AA1Ci(cr) i: 0 ~ 7 channel number | !AA(cr) | Offset error calibrate of assignment channel, User have to input the external standard source for calibrating process | \$031C5(cr) | !03(cr) |
| \$AA2(cr) | !AATTCFF(cr) | The command requests the return of the configuration data from the analog input module at address AA | \$062(cr) | !06FF0600(cr) |
| \$AA5VV(cr) VV: 00 ~ FF each bit for each channel | !AA(cr) | Channel enable/disable setting | \$03507(cr) ex. 07 mean channel 0,1 and 2 is enable and others will be disable | !03(cr) |
| \$AA6(cr) | !AAVV(cr) VV: 00 ~ FF | Get the channel enable/disable status | \$036(cr) | !0307(cr) |
| \$AAF(cr) | !AA(version)(c-r) version:A1.0 | The command requests the module at address AA to return the version code of module firmware | \$03F(cr) | !03A1.0(cr) |
| \$AAM(cr) | !AA(name)(cr)-name : 4017P | The command request the module at address AA to return the module name | \$03M(cr) | !034017P(cr) |
| \$AA7CiRrr(cr) C : channel i : 0 ~ 7 R : Ranger r : Range code | !AA(cr) | R : Range rr : +/-10V : 0x08 +/-5V : 0x09 +/-1V : 0x0a +/-500mV : 0x0b +/-150mV : 0x0c +/-0~20mA: 0x0d +/-4~20mA: 0x07 | \$057R0EW-5C6(cr) | !05(cr) |
| \$AA8Ci(cr)i= 0 ~ 7 (channel number) | !AACiRrr(cr) | Get the input type of the assignment channel | \$058C6(cr) | !05C6R08(cr) |

Command Set

Input range and external calibrating input source for each input type

| Range code(Hex) | Input Range | Max. Signal | Min Signal | Displayed Resolution |
|-----------------|-------------|-------------|------------|----------------------|
| 07 | +/-4-20 mA | +20.000 | -20.000 | 1uA |
| 08 | +/-10 V | +10.000 | -10.000 | 1mV |
| 09 | +/-5 V | +5.0000 | -5.0000 | 100.00uV |
| 0a | +/-1 V | +1.0000 | -1.0000 | 100.00uV |
| 0b | +/-500 mV | +500.00 | -500.00 | 10uV |
| 0c | +/-150 mV | +150.00 | -150.00 | 10uV |
| 0d | +/-20 mA | +20.000 | -20.000 | 1uA |

ADAM-4018 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|--|---|----------|
| %AANNTCCFF | Configuration | Set the address, input range, baud rate, data format, checksum status and/or integration time for the specified analog input module | 4 44 |
| #AAN | Read Analog Input from Channel N | Return the input value from channels number N of the specified analog input module | 4 54 |
| #AA | Read Analog Input from all Channels | Return the input value from all channels of the specified analog module | 4-52 |
| \$AA0 | Span Calibration | Calibrate the analog input module to correct for gain errors | 4 57 |
| \$AA1 | Offset Calibration | Calibrate the analog input module to correct for offset errors | 4 58 |
| \$AA2 | Configuration Status | Return the configuration parameters for the specified analog input module | 4 48 |
| \$AA5VV | Enable/disable Channels for Multiplexing | Enables/disables multiplexing simultaneously for separate channels of the specified input module | 4 55 |
| \$AA6 | Read Channel Status | Ask the specified input module to return the status of all eight channels | 4 56 |
| \$AAF | Read Version | Return the firmware version code from the specified analog input module | 4 50 |
| \$AAM | Read Module Name | Return the module name from the specified analog input module | 4 51 |
| \$AA3 | CJC Status | Returns the value of the CJC sensor for a specified analog input module | 4 63 |
| \$AA9 | CJC Offset Calibration | Calibrates the CJC sensor for offset errors | 4 64 |

Command Set

ADAM-4018+ Command Set

| Command Syntax | Command Response Syntax | Command Description | Command Example | Command response Example |
|---|--|---|----------------------|--|
| %AANNTCCFF(cr)- TT : All channel input range code. CC : Baud rate code, 03 : 1200 04 : 2400 05 : 4800 06 : 9600 07 : 19200 08 : 38400 09 : 57600 0A : 115200 FF : reference command description | !AA (cr) | TT : Initial Value FF (not use) If change FF to XX , All channel input range be seted to XX XX : +/-0~20mA: 0x06 +/-4~20mA: 0x07 J type : 0x0 K type : 0x0f T type : 0x10 E type : 0x11 R type : 0x12 S type : 0x13 B type : 0x14 FF: setting 7 6 x x x x 1 0 bit 7 : Integration time 0 : 50ms (60 HZ) 1 : 60ms (50 HZ) bit 6 : checksum 0 : disable 1 : enable bit 1,0 : Data Format 00 : Engineer unit 01 : % of FSR 10 : Two's complement | %0105FF06- 00(cr) | !05(cr) |
| #AAN(cr)N: 0 ~ 7 | >(data)(cr)data : snnnnnn s : + or -n : digit or dot | Read Analog Input from Channel N (N from 0 - 7) When over range the (data) will show>+999999(cr) When under range the (data) will show>-999999(cr) When burn-out the (data) will show>+888888(cr) | #035(cr) | >+09.789(cr) |
| #AA(cr) | >(data0)(data1- (data2)(data3- (data4)(data5) (cr) data0 ~ 5 : snnnnnn s : + or -n : digit or dot | Read Analog Input (for all 8 ch) When over range the (data n) will show>+999999(cr) When under range the (data n) will show>-999999(cr) When burn-out the (data) will show>+888888(cr) | #03(cr) | >+05.000-04.9- 00 +100.23- 089.32 +09.123 +123.45-09.13- 4(cr)(channel 5 be disable) |

| Command Syntax | Command Response Syntax | Command Description | Command Example | Command response Example |
|--|---|---|--|---|
| #AA(cr) | >(data0)(data1-)(data2)(data3-)(data4)(data5)(cr) data0 ~ 5: s: + or -n : digit or dot | Read Analog Input (for all 8 ch)When over range the (data n) will show>+999999(cr)When under range the (data n) will show>-999999(cr)When burn-out the (data) will show>+888888(cr) | #03(cr) | >+05.000-04.9-00 +100.23-089.32 +09.123 +123.45-09.13-4(cr)(channel 5 be disable) |
| \$AA0Ci(cr): 0 ~ 7 channel number | !AA(cr) | Gain error calibrate of assignment channel, User have to input the external standard source for calibrating process | \$050C7(cr) | !05(cr) |
| \$AA1Ci(cr): 0 ~ 7 channel number | !AA(cr) | Offset error calibrate of assignment channel, User have to input the external standard source for calibrating process | \$031C5(cr) | !03(cr) |
| \$AA2(cr) | !AATCCFF(cr) | The command requests the return of the configuration data from the analog input module at address AA | \$062(cr) | !06FF0600(cr) |
| \$AA5VV(cr)VV: 00 ~ FF each bit for each channel | !AA(cr) | Channel enable/disable setting | \$03507(cr)-ex. 07 mean channel 0,1 and 2 is enable and others will be disable | !03(cr) |
| \$AA6(cr) | !AAVV(cr) VV: 00 ~ FF | Get the channel enable/ disable status | \$036(cr) | !0307(cr) |
| \$AAF(cr) | !AA(version)(c-r)version : A1.0 | The command requests the module at address AA to return the version code of module firmware | \$03F(cr) | !03A1.0(cr) |

Command Set

| Command Syntax | Command Response Syntax | Command Description | Command Example | Command response Example |
|---|---|---|---------------------|---|
| \$AAM(cr) | !AA(name)(cr)-name : 4017P | The command request the module at address AA to return the module name | \$03M(cr) | !034018P(cr) |
| \$AA7CiRrr(cr)C : channeli : 0 ~ 7R : Rangerr : Range code | !AA(cr) | R : Range rr : +/-0~20mA: 0x06 +/-4~20mA: 0x07 J type : 0x0e K type : 0x0f T type : 0x10 E type : 0x11 R type : 0x12 S type : 0x13 B type : 0x14 | \$057R0EW-5C6(cr) | !05(cr) |
| \$AA8Ci(cr)i= 0 ~ 7 (channel number) | !AACiRrr(cr) | Get the input type of the assignment channel | \$058C6(cr) | !05C6R08(cr) |
| \$AAXnnnn(cr)nnnn: 0000 ~ 9999unit: 0.1sec | !AA(cr) | Set communication WDT value form 0000 ~ 9999 (if value is 0000 the communication WDT function will be disable) | \$05X0030(-cr) | !05(cr) |
| \$AAY(cr) | !AAnnnn(cr)n- nnn: 0000 ~ 9999 | Get communication WDT setting value | \$05Y(cr) | !050030(cr) |
| \$AAB(cr) | !AANN(cr)NN: 00 ~ FFbit value : 0 for normal, 1 for open wire | Get the channel operating statusWhen bit value is 0, it mean this channel is in normal operating, when bit value is 1, it mean this channel is opening wire | \$12B(cr) | !1205(cr)05 mean channel 0 and 2 is open |
| \$AA3(cr) | >snnnn.n(cr)s: + or -n: digit | Get the CJC current value | \$063(cr) | >+0035.8(cr) |
| \$AA9SNNNN(cr) S: + or -NNNN: 0000 ~ FFFF | !AA(cr) | Calibrates an analog input module to adjust for offset errors of its CJC sensors | \$079+0500- (cr) | !07(cr) |

ADAM-4018M Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|--|---|----------|
| %AANNTCCFF | Configuration | Set the address, input range, baud rate, data format, checksum status and/or integration time for the specified analog input module | 4 44 |
| #AAN | Read Analog Input from Channel N | Return the input value from channels number N of the specified analog input module | 4 54 |
| \$AA0 | Span Calibration | Calibrate the analog input module to correct for gain errors | 4 57 |
| \$AA1 | Offset Calibration | Calibrate the analog input module to correct for offset errors | 4 58 |
| \$AA2 | Configuration Status | Return the configuration parameters for the specified analog input module | 4 48 |
| \$AA5VV | Enable/disable Channels for Multiplexing | Enables/disables multiplexing simultaneously for separate channels of the specified input module | 4 55 |
| \$AA6 | Read Channel Status | Ask the specified input module to return the status of all eight channels | 4 56 |
| \$AAF | Read Version | Return the firmware version code from the specified analog input module | 4 50 |
| \$AAM | Read Module Name | Return the module name from the specified analog input module | 4 51 |
| \$AA3 | CJC Status | Returns the value of the CJC sensor for a specified analog input module | 4 63 |
| \$AA9 | CJC Offset Calibration | Calibrates the CJC sensor for offset errors | 4 64 |

(continued on following page)

ADAM-4019 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|--|--|----------|
| %AANNTTCCFF | Configuration | Sets the address, baud rate, data format, checksum status, and/or integration time for a specified analog input module | 4-44 |
| #AAN | Read Analog Input from Channel N | Returns the input value from a specified channel of analog input module in the currently configured data format | 4-54 |
| #AA | Analog Data In | Returns the input value from a specified analog input module in the currently configured data format | 4-52 |
| \$AA0Ci | Single Channel Span Calibration | Calibrates a specified channel to correct for gain errors | 4-65 |
| \$AA1Ci | Single Channel Offset Calibration | Calibrates a specified channel to correct for offset errors | 4-66 |
| \$AA2 | Configuration Status | Returns the configuration parameters for the specified analog input module | 4-48 |
| \$AA3 | CJC Status | Returns the value of the CJC sensor for a specified analog input module | 4-63 |
| *** | Synchronized Sampling | Orders all analog input modules to sample their input values and store them in special registers | 4-59 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified module's register after the *** command | 4-60 |
| \$AA5VV | Enable/Disable Channels for Multiplexing | Enable or disable the individual channels in an analog module | 4-55 |
| \$AA6 | Read Channel Status | Get the enable/disable status of all channels in an analog module | 4-56 |
| \$AAAi | CJC Setting | Configure the CJC setting | 4-75 |

ADAM-4021 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|-------------------------|---|---|----------|
| %AANNTTCFF | Configuration | Set the address, output range, baud rate, data format, slew rate and/or checksum status | 4 128 |
| #AA(data) | Analog Data Out | Directs output data to a specified module | 4 131 |
| \$AA4 | Start-up output current/voltage configuration | Stores a default output value in a specified module. The output value will take effect upon startup. | 4 133 |
| \$AA3(number of counts) | Trim Calibration | Trims specified module a number of units up/down | 4 134 |
| \$AA0 | 4 mA Calibration | Tells the module to store parameters for 4 mA Calibration | 4 136 |
| \$AA1 | 20 mA Calibration | Tells the module to store parameters for 20 mA Calibration | 4 137 |
| \$AA2 | Configuration Status | Reads configuration of specified module | 4 138 |
| \$AA6 | Last Value Readback | Returns either last value sent to specified module by #AA command, or start-up output current/voltage | 4 139 |
| \$AA8 | Current Readback | Returns measured value of the current/voltage flowing through current loop | 4 140 |
| \$AA5 | Reset Status | Checks if module has been reset since the last \$AA5 command | 4 141 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified analog output module | 4 142 |
| \$AAM | Read Module Name | Return the module name from the specified analog output module | 4 143 |

ADAM-4050 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|---|----------|
| %AANNTCCFF | Configuration | Sets address, baud rate, and/or checksum status, to a digital I/O module | 4 152 |
| \$AA6 | Digital Data In | Returns the values of the digital I/O channels of the addressed module | 4 154 |
| #AABB(data) | Digital Data Out | Writes specified values to either a single channel or all channels simultaneously | 4 156 |
| *** | Synchronized Sampling | Orders all digital I/O modules to sample their input values and store them in a special register | 4 158 |
| \$AA4 | Read Synchronized Data | Return the value of a specified digital I/O module that was stored after an *** command was issued | 4 159 |
| \$AA2 | Configuration Status | Returns the configuration parameters of a specified digital I/O module | 4 161 |
| \$AA5 | Reset Status | Indicates whether a specified digital I/O module was reset after the last time the \$AA5 command was issued | 4 163 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified digital I/O module | 4 165 |
| \$AAM | Read Module Name | Return the module name from the specified Digital I/O module | 4 166 |

ADAM-4051 Command Set Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|---|----------|
| %AANNTCCFF | Configuration | Sets the address, baud rate, and/or checksum status to a digital I/O module | 4-152 |
| *** | Synchronized Sampling | Orders all digital I/O modules to sample their input values and store them in special registers | 4-158 |
| \$AA2 | Configuration Status | Returns the configuration parameters of a specified digital I/O module | 4-161 |
| \$AA4 | Read Synchronized Data | Returns the value of a specified digital I/O module that was stored in the specified register after an *** command was issued | 4-159 |
| \$AA5 | Reset Status | Indicates whether a specified digital I/O module was reset after the last time that \$AA5 command was issued | 4-163 |
| \$AA6 | Digital Data In | Returns the values of the digital I/O channel in the addressed module | 4-154 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified digital I/O module | 4-165 |
| \$AAM | Read Module Name | Return the module name from the specified digital I/O module | 4-166 |

Note: For command sets and further application, please link to Advantech's web site to download the latest ADAM-4000 user's manual and Windows Utility.

- 1) Link to <http://www.advantech.com>
- 2) Click Support to get in eService Knowledge Center
- 3) Search for download and key-in "ADAM-4000" to get the latest ADAM-4000 User's Manual

ADAM-4052 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|---|----------|
| %AANNTCCFF | Configuration | Sets address, baud rate, and/or checksum status, to a digital I/O module | 4 152 |
| \$AA6 | Digital Data In | Returns the values of the digital I/O channels of the addressed module | 4 154 |
| *** | Synchronized Sampling | Orders all digital I/O modules to sample their input values and store them in a special register | 4 158 |
| \$AA4 | Read Synchronized Data | Return the value of a specified digital I/O module that was stored after an *** command was issued | 4 159 |
| \$AA2 | Configuration Status | Returns the configuration parameters of a specified digital I/O module | 4 161 |
| \$AA5 | Reset Status | Indicates whether a specified digital I/O module was reset after the last time the \$AA5 command was issued | 4 163 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified digital I/O module | 4 165 |
| \$AAM | Read Module Name | Return the module name from the specified digital I/O module | 4 166 |

ADAM-4053 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|---|----------|
| %AANNTTCFF | Configuration | Sets address, baud rate, and/or checksum status, to a digital I/O module | 4 152 |
| \$AA6 | Digital Data In | Returns the values of the digital I/O channels of the addressed module | 4 154 |
| *** | Synchronized Sampling | Orders all digital I/O modules to sample their input values and store them in a special register | 4 158 |
| \$AA4 | Read Synchronized Data | Return the value of a specified digital I/O module that was stored after an *** command was issued | 4 159 |
| \$AA2 | Configuration Status | Returns the configuration parameters of a specified digital I/O module | 4 161 |
| \$AA5 | Reset Status | Indicates whether a specified digital I/O module was reset after the last time the \$AA5 command was issued | 4 163 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified digital I/O module | 4 165 |
| \$AAM | Read Module Name | Return the module name from the specified digital I/O module | 4 166 |

ADAM-4055 Command Set Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|---|----------|
| %AANNTTCFF | Configuration | Sets the address, baud rate, and/or checksum status to a digital I/O module | 4-152 |
| #AABB(data) | Digital Data Out | Writes specified values to either a single channel or all channels simultaneously | 4-156 |
| *** | Synchronized Sampling | Orders all digital I/O modules to sample their input values and store them in special registers | 4-158 |
| \$AA2 | Configuration Status | Returns the configuration parameters of a specified digital I/O module | 4-161 |
| \$AA4 | Read Synchronized Data | Returns the value of a specified digital I/O module that was stored in the specified register after an *** command was issued | 4-159 |
| \$AA5 | Reset Status | Indicates whether a specified digital I/O module was reset after the last time that \$AA5 command was issued | 4-163 |
| \$AA6 | Digital Data In | Returns the values of the digital I/O channel in the addressed module | 4-154 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified digital I/O module | 4-165 |
| \$AAM | Read Module Name | Return the module name from the specified digital I/O module | 4-166 |

ADAM-4060/4068 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|------------------------|---|----------|
| %AANNTCCFF | Configuration | Sets address, baud rate, and/or checksum status, to a digital I/O module | 4 152 |
| \$AA6 | Digital Data In | Returns the values of the digital I/O channels of the addressed module | 4 154 |
| #AABB(data) | Digital Data Out | Writes specified values to either a single channel or all channels simultaneously | 4 156 |
| #** | Synchronized Sampling | Orders all digital I/O modules to sample their input values and store them in a special register | 4 158 |
| \$AA4 | Read Synchronized Data | Return the value of a specified digital I/O module that was stored after an #** command was issued | 4 159 |
| \$AA2 | Configuration Status | Returns the configuration parameters of a specified digital I/O module | 4 161 |
| \$AA5 | Reset Status | Indicates whether a specified digital I/O module was reset after the last time the \$AA5 command was issued | 4 163 |
| \$AAF | Read Firmware Version | Return the firmware version code from the specified digital I/O module | 4 165 |
| \$AAM | Read Module Name | Return the module name from the specified digital I/O module | 4 166 |

ADAM-4080 Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|---------------------------------|---|----------|
| %AANNTTCFF | Configuration | Sets the address, input mode, baud rate, checksum status and/or frequency gate time for a specified counter/ frequency module | 4 168 |
| \$AA2 | Configuration Status | Returns configuration parameters from the specified counter/frequency module | 4 170 |
| \$AAF | Read Firmware Version | Return firmware version code from the specified counter/frequency module | 4 171 |
| \$AAM | Read Module Name | Return the module name from the specified counter/frequency module | 4 172 |
| \$AABS | Set Input Signal Mode | Sets the input signal mode of the specified counter/frequency module to either non-isolated or photo-isolated input signals | 4 173 |
| \$AAB | Read Input Signal Mode | Read the input signal mode of the specified counter/frequency module. | 4 174 |
| #AAN | Read Counter or Frequency Value | Returns the value of counter 0 or counter 1 from a specified counter/ frequency module in hex format | 4 175 |

(continued on following page)

| Command Syntax | Command Name | Command Description | Page No. |
|-----------------------|--------------------------------|--|-----------------|
| \$AAAG | Set Gate Mode | Requests the specified counter/frequency module to set its gate mode to either high, low or disabled | 4 180 |
| \$AAA | Read Gate Mode | Requests the specified counter/frequency module to return the status of its gate mode | 4 181 |
| \$AA3N(data) | Set Maximum Counter Value | Sets the maximum value of counter 0 or counter 1 for the specified counter/frequency module | 4 182 |
| \$AA3N | Read Maximum Counter Value | Reads the maximum value of counter 0 or counter 1 of the specified counter/frequency module | 4 183 |
| \$AA5NS | Start/Stop Counter | The command orders the specified counter/frequency module to start or stop counting | 4 184 |
| \$AA5N | Read Counter Start/Stop Status | The addressed counter frequency module returns its status indicating whether counting is enabled or disabled | 4 185 |
| \$AA6N | Clear Counter | The command clears the counter 0 or counter 1 of the specified counter module | 4 186 |
| \$AA7N | Read Overflow Flag | The addressed module returns the status of the overflow flag of counter 0 or counter 1 | 4 187 |

(continued on following page)

Command Set

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|---|---|----------|
| \$AA4S | Enable/Disable Digital Filter | Enables or disables the digital filter of the addressed counter/frequency module | 4 190 |
| \$AA4 | Read Filter Status | The addressed counter frequency module returns the status of its digital filter | 4 191 |
| \$AA0H(data) | Set Minimum Input Signal Width at High Level | Sets the minimum input signal width at high level for a specified counter/frequency module | 4 192 |
| \$AA0H | Read Minimum Input Signal Width at High Level | Reads the minimum input signal width setting at high level for a specified counter/frequency module | 4 193 |
| \$AA0L(data) | Set Minimum Input Signal Width at Low Level | Sets the minimum input signal width at low level for a specified counter/frequency module | 4 194 |
| \$AA0L | Read Minimum Input Signal Width at Low Level | Reads minimum input signal width setting at low level for a specified counter/frequency module | 4 195 |
| \$AA1H(data) | Set Non-isolated High Trigger Level | Sets the high trigger level of non-isolated input signals for a specified counter/frequency module | 4 196 |
| \$AA1H | Read Non-isolated High Trigger Level | Requests the addressed counter frequency module to return the high trigger level for non-isolated input signals | 4 197 |
| \$AA1L(data) | Set Non-isolated Low Trigger Level | Sets the low trigger level of non-isolated input signals for a specified counter/frequency module | 4 198 |
| \$AA1L | Read Non-isolated Low Trigger Level | Requests the addressed counter/ frequency module to return the low trigger level for non-isolated input signals | 4 199 |

(continued on following page)

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|---------------------------------------|---|----------|
| @AAPN(data) | Set Initial Count Value of Counter N | Sets the initial count value of the module for counter 0 or counter 1 | 4 203 |
| @AAGN | Read Initial Count Value of Counter N | Reads the initial count value of counter 0 or counter 1 | 4 204 |
| @AAEAN | Enable Alarm of Counter N | Enable alarm for the specified counter 0 or counter 1 | 4 205 |
| @AADAN | Disable Alarm of Counter N | Disable alarm for the specified counter 0 or counter 1 | 4 206 |
| @AAPA(data) | Set Alarm Limit Value of Counter 0 | Download the alarm limit value for counter 0 of the specified module | 4 207 |
| @AASA(data) | Set Alarm Limit Value of Counter 1 | Download the alarm limit value for counter 1 of the specified module | 4 207 |
| @AARP | Read Alarm Limit Value of Counter 0 | Ask the module to return the alarm limit value of counter 0 | 4 208 |
| @AARA | Read Alarm Limit Value of Counter 1 | Ask the module to return the alarm limit value of counter 1 | 4 208 |
| @AADO (data) | Set Digital Output Values | Set the values of the module's two digital outputs (ON or OFF) | 4 209 |
| @AADI | Read Digital Output and Alarm Status | Ask the module to return the status state of its two digital outputs and the status of its alarm. | 4 210 |

ADAM-4080D Command Table

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|---------------------------------|---|----------|
| %AANNTCCFF | Configuration | Sets the address, input mode, baud rate, checksum status and/or frequency gate time for a specified counter/ frequency module | 4 168 |
| \$AA2 | Configuration Status | Returns configuration parameters from the specified counter/frequency module | 4 170 |
| \$AAF | Read Firmware Version | Return firmware version code from the specified counter/frequency module | 4 171 |
| \$AAM | Read Module Name | Return the module name from the specified counter/frequency module | 4 172 |
| \$AABS | Set Input Signal Mode | Sets the input signal mode of the specified counter/frequency module to either non-isolated or photo-isolated input signals | 4 173 |
| \$AAB | Read Input Signal Mode | Read the input signal mode of the specified counter/frequency module. | 4 174 |
| #AAN | Read Counter or Frequency Value | Returns the value of counter 0 or counter 1 from a specified counter/ frequency module in hex format | 4 175 |
| \$AA8V | Select LED Data Origin | Select whether LED will display data from the counter/frequency module directly or from the host computer | 4 176 |
| \$AA8 | Read LED Data Origin | Returns the status of the LED Data origin. Origin is either direct from module or from host computer | 4 177 |
| \$AA9(data) | Send Data to LED | PC sends data to LED display. This command is valid only after selectting to display host computer data (\$AA8V) | 4 178 |

(continued on following page)

| Command Syntax | Command Name | Command Description | Page No. |
|-----------------------|--------------------------------|--|-----------------|
| \$AAAG | Set Gate Mode | Requests the specified counter/frequency module to set its gate mode to either high, low or disabled | 4 180 |
| \$AAA | Read Gate Mode | Requests the specified counter/frequency module to return the status of its gate mode | 4 181 |
| \$AA3N(data) | Set Maximum Counter Value | Sets the maximum value of counter 0 or counter 1 for the specified counter/frequency module | 4 182 |
| \$AA3N | Read Maximum Counter Value | Reads the maximum value of counter 0 or counter 1 of the specified counter/frequency module | 4 183 |
| \$AA5NS | Start/Stop Counter | The command orders the specified counter/frequency module to start or stop counting | 4 184 |
| \$AA5N | Read Counter Start/Stop Status | The addressed counter frequency module returns its status indicating whether counting is enabled or disabled | 4 185 |
| \$AA6N | Clear Counter | The command clears the counter 0 or counter 1 of the specified counter module | 4 186 |
| \$AA7N | Read Overflow Flag | The addressed module returns the status of the overflow flag of counter 0 or counter 1 | 4 187 |

(continued on following page)

Command Set

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|---|---|----------|
| \$AA4S | Enable/Disable Digital Filter | Enables or disables the digital filter of the addressed counter/frequency module | 4 190 |
| \$AA4 | Read Filter Status | The addressed counter frequency module returns the status of its digital filter | 4 191 |
| \$AA0H(data) | Set Minimum Input Signal Width at High Level | Sets the minimum input signal width at high level for a specified counter/frequency module | 4 192 |
| \$AA0H | Read Minimum Input Signal Width at High Level | Reads the minimum input signal width setting at high level for a specified counter/frequency module | 4 193 |
| \$AA0L(data) | Set Minimum Input Signal Width at Low Level | Sets the minimum input signal width at low level for a specified counter/frequency module | 4 194 |
| \$AA0L | Read Minimum Input Signal Width at Low Level | Reads minimum input signal width setting at low level for a specified counter/frequency module | 4 195 |
| \$AA1H(data) | Set Non-isolated High Trigger Level | Sets the high trigger level of non-isolated input signals for a specified counter/frequency module | 4 196 |
| \$AA1H | Read Non-isolated High Trigger Level | Requests the addressed counter frequency module to return the high trigger level for non-isolated input signals | 4 197 |
| \$AA1L(data) | Set Non-isolated Low Trigger Level | Sets the low trigger level of non-isolated input signals for a specified counter/frequency module | 4 198 |
| \$AA1L | Read Non-isolated Low Trigger Level | Requests the addressed counter/ frequency module to return the low trigger level for non-isolated input signals | 4 199 |

(continued on following page)

| Command Syntax | Command Name | Command Description | Page No. |
|----------------|--|---|----------|
| @AAEAT | Enable Alarm | Enable the alarm in either momentary or latching mode | 4 32 |
| @AADA | Disable Alarm | Disable all alarm functions | 4 213 |
| @AACA | Clear Latch Alarm | The latch alarm is reset | 4 214 |
| @AAPA(data) | Set Low-Alarm Count Value of Counter 0 | Downloads the low-alarm count value for counter 0 of the specified counter/frequency module | 4 215 |
| @AASA(data) | Set High-Alarm Count Value of Counter 0 | Downloads the high-alarm count value for counter 0 of the specified counter/frequency module | 4 216 |
| @AARP | Read Low-Alarm Count Value of Counter 0 | Ask the addressed module to return its low-alarm count value for counter 0 | 4 217 |
| @AARA | Read High-Alarm Count Value of Counter 0 | Ask the addressed module to return its high-alarm count value for counter 0 | 4 218 |
| @AADO | Set Digital Output Values | Set the values of the module's two digital outputs (ON or OFF) | 4 209 |
| @AADI | Read Digital Output and Alarm Status | Ask the addressed module to return the state of its two digital output channels and the status of its alarm | 4 210 |

Command Set

4.4 Analog Input Module Command

4.4.1 Analog Input Command Set

| Command Syntax | Command Name | Description | I/O Module |
|----------------|--|---|--|
| %AANNTCCFF | Configuration | Sets the address, input range, baud rate, data format, checksum status, and/or integration time for a specified analog input module | 4011, 4011D, 4012, 4013, 4015, 4016, 4017, 4018, 4018M, 4019 |
| \$AA2 | Configuration Status | Returns the configuration parameters for the specified analog input module | 4011, 4011D, 4012, 4013, 4015, 4016, 4017, 4018, 4018M, 4019 |
| \$AAF | Read Firmware Version | Returns the firmware version code from the specified analog input module | 4011, 4011D, 4012, 4013, 4015, 4016, 4017, 4018, 4018M, 4019 |
| \$AAM | Read Module Name | Returns the module name from the specified analog input module | 4011, 4011D, 4012, 4013, 4015, 4016, 4017, 4018, 4018M, 4019 |
| #AA | Analog Data In | Returns the input value from a specified analog input module in the currently configured data format | 4011, 4011D, 4012, 4013, 4015, 4016, 4017, 4018, 4019 |
| #AAN | Read Analog Input from Channel N | Returns the input value from channel number n of the specified analog input module | 4015, 4017, 4018, 4018M, 4019 |
| #AA5VV | Enable/Disable Channels for Multiplexing | Enables/disables multiplexing simultaneously for separate channels of the specified input module | 4015, 4017, 4018, 4018M, 4019 |

(continued on following page)

Command Set

| Command Syntax | Command Name | Description | I/O Module |
|----------------|------------------------|--|--|
| \$AA6 | Read Channel Status | Ask the specified input module to return the status of all eight channels | 4015, 4017, 4018, 4018M, 4019 |
| \$AA0 | Span Calibration | Calibrate the analog input module to correct for gain errors | 4011, 4011D, 4012, 4013, 4016, 4017, 4018, 4018M |
| \$AA1 | Offset Calibration | Calibrate the analog input module to correct for offset errors. | 4011, 4011D, 4012, 4013, 4016, 4017, 4018, 4018M |
| *** | Synchronized Sampling | Orders all analog input modules to sample their input values and store them in special registers | 4011, 4011D, 4012, 4013, 4015, 4016, 4019 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified module's register after the *** command | 4011, 4011D, 4012, 4013, 4015, 4016, 4019 |
| \$AAB | Channel Diagnose | Ask the module to respond whether the wiring is open or closed | 4011D, 4015 |
| \$AA3 | CJC Status | Returns the value of the CJC sensor for a specified analog input module | 4011, 4011D, 4018, 4018M, 4019 |
| \$AA9 | CJC Offset Calibration | Calibrates the CJC sensor for offset errors | 4011, 4011D, 4018, 4018M, 4019 |

| Command Syntax | Command Name | Command Description | I/O Module |
|----------------|---|---|------------|
| \$AA0Ci | Single Channel Span Calibration | Calibrates a specified channel to correct for gain errors | 4015, 4019 |
| \$AA1Ci | Single Channel Offset Calibration | Calibrates a specified channel to correct for offset errors | 4015, 4019 |
| \$AA7CiRrr | Single Channel Range Configuration | Configure the input type and range of the specified channel in an analog input module | 4015, 4019 |
| \$AA8Ci | Read Single Channel Range Configuration | Get the input type and range of the specified channel in an analog input module | 4015, 4019 |
| \$AAxnnnn | Watchdog Timer Setting | Set WDT communication cycle | 4015, 4019 |
| \$AAy | Read Watchdog Timer Setting | Read the setting of WDT communication cycle | 4015, 4019 |
| \$AAS0 | Internal Calibration | Internal self-calibration for offset and gain errors | 4015 |
| \$AAS1 | Reload default calibrating parameter | Reload factory default calibrating parameter to overwrite current calibrating parameter | 4015 |
| \$AA9SNNNN | CJC Offset Calibration | Calibrates the CJC sensor for offset errors | 4019 |
| \$AAAi | CJC Setting | Configure CJC setting | 4019 |

%AANNTTCCFF

Name Configuration

Description Sets address, input range, baud rate, data format, checksum status, and/or integration time for an analog input module.

Syntax %AANNTTCCFF(cr)

% is a delimiter character.

AA(range 00-FF) represents the 2-character hexadecimal address of the analog input module you want to configure.

NN represents the new hexadecimal address of the analog input module. Range is from 00h to FFh.

TT represents the type (input range) code.(4015 and 4019 must be 00)

CC represents the baud rate code.

FF is a hexadecimal number that equals the 8-bit parameter representing the data format, checksum status and integration time. The layout of the 8-bit parameter is shown in figure 4-1. Bits 2 through 5 are not used and are set to 0.

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

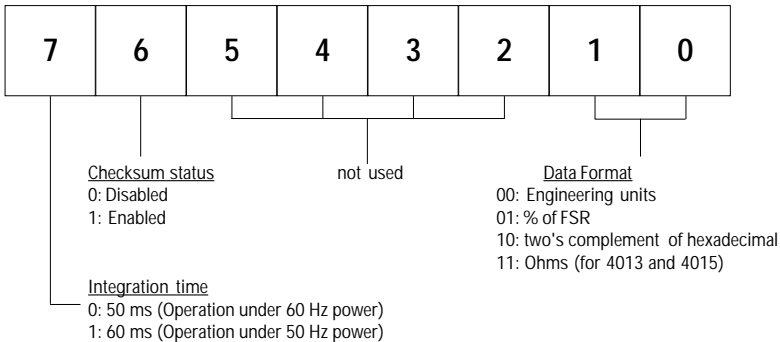


Figure 4-1 Data format for 8-bit parameter

%AANNTCCFF

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 or communication error or if the specified address does not exist.

! delimiter character indicates a valid command was received.

? delimiter character indicates the command was invalid

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.

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

Example

command: %2324050600(cr)

response: !24(cr)

The ADAM-4011 module with address 23h is configured to a new address of 24h, an input range ± 2.5 V, baud rate 9600, integration time 50 ms (60 Hz), engineering units data format and no checksum checking or generation.

The response indicates that the command was received.

Wait 7 seconds to let the new configuration settings take effect before issuing a new command to the module.

NOTICE: *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.*

NOTICE: *All configuration parameters can be changed dynamically, except checksum and baud rate parameters. They can only be altered when the INIT* terminal is grounded. (Refer to Baud rate and Checksum configuration in Chapter 2, for the correct procedure)*

Table 4-1 *Input Range Codes (Type Code)*

| Input Range Code (Hex) | Input Range for 4011,4011D,4018,4018+,4018M |
|-------------------------------|--|
| 00 | ± 15 mV |
| 01 | ± 50 mV |
| 02 | ± 100 mV |
| 03 | ± 500 mV |
| 04 | ± 1 V |
| 05 | ± 2.5 V |
| 06 | ± 20 mA ¹ |
| 0E | Type J Thermocouple 0° to 760° C |
| 0F | Type K Thermocouple 0° to 1370° C |
| 10 | Type T Thermocouple -100° to 400° C |
| 11 | Type E Thermocouple 0° to 1370° C |
| 12 | Type R Thermocouple 500° to 1750° C |
| 13 | Type S Thermocouple 500° to 1750° C |
| 14 | Type B Thermocouple 500° to 1800° C |

| Input Range Code(Hex) | Input Range for 4012,4017,4017+ |
|------------------------------|--|
| 08 | ± 10 V |
| 09 | ± 5 V |
| 0A | ± 1 V |
| 0B | ± 500 mV |
| 0C | ± 150 mV |
| 0D | ± 20 mA ¹ |

Note¹: The input range requires the useage of a 125 Ω current conversion resistor

(continued on following page)

| Input Range Code (Hex) | Input Range for 4013 |
|------------------------|---|
| 20 | Platinum, -100° to 100°C, $\alpha=0.00385$ |
| 21 | Platinum, 0° to 100°C, $\alpha=0.00385$ |
| 22 | Platinum, 0° to 200°C, $\alpha=0.00385$ |
| 23 | Platinum, 0° to 600°C, $\alpha=0.00385$ |
| 24 | Platinum, -100° to 100°C, $\alpha=0.003916$ |
| 25 | Platinum, 0° to 100°C, $\alpha=0.003916$ |
| 26 | Platinum, 0° to 200°C, $\alpha=0.003916$ |
| 27 | Platinum, 0° to 600°C, $\alpha=0.003916$ |
| 28 | Nickel, -80° to 100°C |
| 29 | Nickel, 0° to 100°C |

| Input Range Code (Hex) | Input Range for ADAM-4016 |
|------------------------|---------------------------|
| 00 | ± 15 mV |
| 01 | ± 50 mV |
| 02 | ± 100 mV |
| 03 | ± 500 mV |
| 06 | ± 20 mA |

Table 4-2 *Baud Rate Codes*

| Baud Rate Code (hex) | Baud Rate |
|----------------------|-----------|
| 03 | 1200 bps |
| 04 | 2400 bps |
| 05 | 4800 bps |
| 06 | 9600 bps |
| 07 | 19.2 kbps |
| 08 | 38.4 kbps |

\$AA2

| | |
|--------------------|---|
| Name | Configuration Status |
| Description | The command requests the return of the configuration data from the analog input module at address AA. |
| Syntax | <p>\$AA2(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module that you want to interrogate.</p> <p>2 is the Configuration Status command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AATTCCFF(cr) if the command is valid.</p> <p>?AA(cr)if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>!delimiter character indicates a valid command was received.</p> <p>?delimiter character indicates the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>TT represents the type code. Type code determines the input range.</p> <p>CC represents the baud rate code.</p> <p>FF is a hexadecimal number that equals the 8-bit parameter that represents the data format, checksum status and integration time . The layout of the 8-bit parameter is shown in figure 4-1. Bits 2 to 5 are not used, and are set to 0.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> <p>(Also see the %AANNTTCCFF configuration command)</p> |

\$AA2

Example

command: \$452(cr)
response: !45050600(cr)

The command asks the analog input module at address 45h to send its configuration data.

The analog input module at address 45h responds with an input range of 2.5 volts, a baud rate of 9600 bps, an integration time of 50 ms (60 Hz), engineering units are the currently configured data format, and no checksum function or checksum generation.

\$AAF

| | |
|--------------------|--|
| Name | Read Firmware Version |
| Description | The command requests the analog input module at address AA to return the version code of its firmware |
| Syntax | <p>\$AAF(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module that you want to interrogate.</p> <p>F identifies the version command.</p> <p>(cr) is the terminating character, carriage return (ODh)</p> |
| Response | <p>!AA(Version)(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>(Version) is the version code of the module's firmware at address AA.</p> <p>(cr) is the terminating character, carriage return (ODh).</p> |

\$AAM

Name Read Module Name

Description The command requests the analog input module at address AA to return its name

Syntax \$AAM (cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module that you want to interrogate.

M is the Read Module Name command.

(cr) is the terminating character, carriage return (ODh)

Response !AA(Module Name)(cr) if the command is valid.

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

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

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.

(Module Name) is the name of the module at address AA.

(cr) is the terminating character, carriage return (ODh).

#AA

Name Analog Data In

Description The command will return the input value from a specified (AA) module in the currently configured data format.

Syntax #AA(cr)

is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.

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

Response >(data)(cr)

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.

(data) is the input value in the configured data format of the interrogated module. (For data formats, see Appendix B).

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

Example command: #33(cr)

response: >+5.8222(cr)

The command interrogates the analog input module at address 33h for its input value.

The analog input module responds with +5.8222 volts. (The configured data format of the analog input module in this case is engineering units.)

Example command: #21(cr)

response: +7.2111+7.2567+7.3125+7.1000
+7.4712+7.2555+7.1234+7.5678 (cr)

The command interrogates the analog input module at address 21h for its input values of all channels.

The analog input module responds with channels from 0 to 7 with +7.2111 volts, +7.2567 volts, +7.3125 volts, +7.1000 volts, +7.4712 volts, +7.2555 volts, +7.1234 volts and +7.5678 volts.

#AA

Example command: #DE(cr)
 response: >FF5D(cr)

The analog input module at address DEh has an input value of FF5D. (The configured data format of the analog input module is twos complement)

| | Twos complement | % of Span | Engineering units |
|-------------|-----------------|-----------|-------------------|
| under-limit | 0000 | -0000 | -0000 |
| over-limit | FFFF | +9999 | +9999 |

NOTICE: *When modules measure **Thermocouple** or **RTD** input values that are outside their configured range they will send data that implies input out of bounds. The next table shows the values that the modules will return, depending on the configured data format and if the input value falls under or exceeds the configured range.*

Only when modules are configured for Thermocouple or RTD will this “input out of bounds” warning occur. When analog input modules measure voltage or current that falls outside the configured range, they will return the actual measured input!

In the next example the target module is configured for an input range of T/C type J (Input range: 0 - 760° C) and for a data format in engineering units. The module measures an input value of 820° C.

Example command: #D1(cr)
 response: >+9999(cr)

By returning a high value, +9999, the module at address D1h indicates that the measured input value exceeds the configured range.

#AAN

| | |
|--------------------|---|
| Name | Read Analog Input from Channel N |
| Description | The command will return the input value from one of the eight channels of a specified (AA) module in the currently configured data format. |
| Syntax | <p>#AAN(cr)</p> <p># is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.</p> <p>N identifies the channel you want to read. The value can range from 0 to 7 for 4017, 4018, 4018M, 4019. (the range of 4015 is from 0 to 5)</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>>(data)(cr)</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>> is a delimiter character.</p> <p>(data) is the input value of the channel number N. Data consists of a + or - sign followed by five decimal digits with a fixed decimal point.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: #120(cr)</p> <p>response: >+1.4567(cr)</p> <p>The command requests the analog input module at address 12h to return the input value of channel 0.</p> <p>The analog input module responds that the input value of channel 0 is equal to +1.4567 volts.</p> |

\$AA5VV

| | |
|--------------------|---|
| Name | Enable/disable channels for multiplexing |
| Description | Enables/disables multiplexing simultaneously for separate channels of a specified input module |
| Syntax | <p>\$AA5VV(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of analog input module.</p> <p>5 is the enable/disable channels command.</p> <p>VV are two hexadecimal values. The values are interpreted by the module as two binary words (4-bit). The first word represents the status of channel 4-7, the second word represents the status of channel 0-3. Value 0 means the channel is disabled, value 1 means the channel is enabled.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>!delimiter character indicates a valid command was received.</p> <p>?delimiter character indicates the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$00581(cr)</p> <p>response: !00(cr)</p> <p>Hexadecimal 8 equals binary 1000, which enables channel 7 and disables channels 4, 5, and 6.</p> <p>Hexadecimal 1 equals binary 0001, which enables channel 0 and disables channel 1,2, and 3.</p> |

\$AA6

Name Read Channel Status

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

Syntax \$AA6(cr)

AA (range 00-FF) represents the 2-character hexadecimal address of analog input module of which the channel status you want to send. The channel status defines whether a channel is enabled or disabled

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

Response !AAVV(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 indicates a valid command was received.

?delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.

VV are two hexadecimal values. The values are interpreted by the module as two binary words (4-bit). The first word represents the status of channel 4-7, the second word represents the status of channel 0-3. Value 0 means the channel is disabled, value 1 means the channel is enabled.

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

Example command: \$026(cr)

response: !02FF(cr)

The command asks the analog input module at address 02 to send the status of it input channels. The analog input module at address 02 responds that all its multiplex channels are enabled (FF equals 1111 and 1111).

\$AA0

| | |
|--------------------|--|
| Name | Span Calibration |
| Description | Calibrates an analog input module to correct for gain errors. |
| Syntax | \$AA0(cr) \$ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be calibrated. 0 represents the span calibration command. (cr) is the terminating character, carriage return (0Dh). |
| Response | !AA(cr) if the command was 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 indicates a valid command was received. ?delimiter character indicates the command was invalid. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module. (cr) represents terminating character, carriage return (0Dh). |

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. (See also Chapter 5, Calibration)

NOTICE: *An analog input module requires a maximum of 7 seconds to perform auto calibration and ranging after it received a Span Calibration command . During this interval, the module can not be addressed to perform any other actions.*

\$AA1

| | |
|--------------------|--|
| Name | Offset Calibration. |
| Description | Calibrates an analog input module to correct for offset errors. |
| Syntax | <p>\$AA1(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module you want to calibrate.</p> <p>1 represents the offset calibration command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>!delimiter character indicates a valid command was received.</p> <p>?delimiter character indicates the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |

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. (See also Chapter 5, Calibration)

NOTICE: *An analog input module requires a maximum of 7 seconds to perform auto calibration and ranging after it received an Offset Calibration command . During this interval, the module can not be addressed to perform any other actions.*

Name Synchronized Sampling

Description Orders all analog input modules to sample their input values and store the values in special registers.

Syntax ***

is a delimiter character.

** is the actual synchronized sampling command.

The terminating character, in the form of a carriage return (0Dh), is not required.

Response The analog input modules will send no response after executing the synchronized sampling command. In order to retrieve the data, a separate Read Synchronized Data command has to be issued for every analog input module.

The pound sign (#) followed by two asterisks (**) does not represent an optional value, but is the actual command string.

\$AA4

| | |
|--------------------|--|
| Name | Read Synchronized Data |
| Description | Returns the input value that was stored in the addressed module's register, after a Synchronized Sampling command #** was issued. |
| Syntax | <p>\$AA4(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module from which data is to be sent.</p> <p>4 is the Read Synchronized Data command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(status)(data)(cr) if the command was valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicates a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module that is responding.</p> <p>(status) will tell you if the data (data) from the last Synchronized Sampling command (#**) has already been sent. If status=1, then the data has been sent for the first time since a Synchronized Sampling command was issued. If status=0, then the data has been sent at least once before.</p> <p>(data) a value stored in a special register of the interrogated module in the configured data format. It has been sampled by the module after a Synchronized Sampling command. (For possible data formats, see Appendix B, Data Formats and I/O Ranges)</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |

Example

command: \$074(cr)
response: >071+5.8222(cr)

The command asks the analog input module at address 07h to send its analog input data.

The analog input module responds with status = 1, which means that this is the first time that the data has been sent and that the data = +5.8222 Volts.

(Configured data format of the analog input module in this case is engineering units.)

command: \$074(cr)
response: >070+5.8222(cr)

The command asks the analog input module at address 07h to send its analog input data.

The analog input module responds with status = 0, which means that it has sent the same data at least once before, and data = +5.8222 Volts. This could indicate that a previous Synchronized Sampling command was not received!

(Configured data format of the analog input module in this case is engineering units.)

\$AAB

Name Channel Diagnose

Description Diagnose channel status in over range, under range, and wire opening

Syntax \$AAB(cr)

\$ is a delimiter character

AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module to be detected.

B is the channel diagnose command.

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

Response !AA0(cr) if the module detects a close thermocouple. (4011D only)

!AA1(cr) if the module detects an open thermocouple. (4011D only)

!AANN(cr) if the command is valid when it applied with ADAM-4015.

?AA(cr) if an invalid command was 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 indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.

NN (range 00-FF) is a hexadecimal number that equals the 8-bit parameter, representing the status of analog input channels. Bit value 0 means normal status; and bit value 1 means channel over range, under range, or open wiring.

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

\$AA3

| | |
|--------------------|--|
| Name | CJC Status command |
| Description | Instructs the addressed analog input module to read its CJC (Cold Junction Compensation) sensors and return the acquired data. |
| Syntax | <p>\$AA3(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which contains the CJC Status you wish to retrieve.</p> <p>3 is CJC Status command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>>data(cr) if the command is valid.</p> <p>?AA(cr) if an invalid command was issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>!delimiter character indicates a valid command was received.</p> <p>?delimiter character indicates the command was invalid</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>(data) is the value that is retrieved by the module by reading its CJC sensor. The data format, in degrees Celsius, consists an “+” or “-” sign followed by five decimal digits and a fixed decimal point. The resolution of the data is 0.1 °C.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$093(cr)</p> <p>response: >+0036.8(cr)</p> <p>The command request the analog input module at address 09h to read its CJC sensor and return the data. The analog input module at address 09h responds with: 36.8°C.</p> |

\$AA9

| | |
|--------------------|---|
| Name | CJC Offset Calibration |
| Description | Calibrates an analog input module to adjust for offset errors of its CJC (Cold Junction Compensation) sensors. |
| Syntax | <p>\$AA9S(number of counts)(cr).</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which contains the CJC Status you wish to retrieve.</p> <p>9 is CJC Status command.</p> <p>S sign, + or -, indicates whether to increase or decrease the CJC offset value.</p> <p>(number of counts) a four character hexadecimal “count” value. Each count equals approximately 0.009° C. The value can range from 0000 to FFFF.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>?AA(cr) if an invalid command was issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>!delimiter character indicates a valid command was received.</p> <p>?delimiter character indicates the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$079+0042(cr)</p> <p>response: !07(cr)</p> <p>The command increases the CJC offset value of the analog input module at address 07h with 66 counts (42 hex) which equals about 0.6° C.</p> |

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

\$AA0Ci

| | |
|--------------------|---|
| Name | Single Channel Span Calibration |
| Description | The command calibrates a specified channel to correct for gain errors. |
| Syntax | <p>\$AA0Ci(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be calibrated.</p> <p>0 represents the span calibration command.</p> <p>Ci represent the specified input channel you want to calibrate.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |

| | |
|-----------------|---|
| Response | <p>!AA(cr) if the command was valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicates a valid command was received.</p> <p>? delimiter character indicates the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
|-----------------|---|

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. (See also Chapter 5, Calibration)

NOTICE: *An analog input module requires a maximum of 7 seconds to perform auto calibration and ranging after it received a Span Calibration command. During this interval, the module can not be addressed to perform any other actions.*

\$AA1Ci

Name Single Channel Offset Calibration

Description The command calibrates a specified channel to correct for offset errors.

Syntax \$AA1Ci(cr)
\$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be calibrated.
1 represents the offset calibration command.
Ci represent the specified input channel you want to calibrate.
(cr) is the terminating character, carriage return (0Dh).

Response !AA(cr) if the command was 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 indicates a valid command was received.
? delimiter character indicates the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.
(cr) represents terminating character, carriage return (0Dh).

Example command: \$021C5(cr)
response: !02(cr)
The command calibrates channel 5 of the analog input module at address 02 for correcting offset errors.

\$AA7CiRrr

| | |
|--------------------|---|
| Name | Single Channel Range Configuration |
| Description | This command configure the input type and range of the specified channel in an analog input module. |
| Syntax | <p>\$AA7CiRrr(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be configured.</p> <p>7 represents the range configuration command.</p> <p>Ci represent the specified input channel you want to configure.</p> <p>Rrr represent the type and range you want to set. (refer to Table 4-3 to check range code)</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command was valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicates a valid command was received.</p> <p>? delimiter character indicates the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$027C5R21(cr)</p> <p>response: !02(cr)</p> <p>The command configures the range of channel 5 in the analog input module at address 02 as Pt100 (IEC) 0~100oC.</p> |

Command Set

ADAM-4015 command codes against Input ranges table

| Command Code (Hex) | Input Type | Input Range |
|--------------------|--------------------|-----------------|
| 20 | Platinum 100 (IEC) | -50 C to 150 C |
| 21 | Platinum 100 (IEC) | 0 C to 100 C |
| 22 | Platinum 100 (IEC) | 0 C to 200 C |
| 23 | Platinum 100 (IEC) | 0 C to 400 C |
| 24 | Platinum 100 (IEC) | -200 C to 200 C |
| 25 | Platinum 100 (JIS) | -50 C to 150 C |
| 26 | Platinum 100 (JIS) | 0 C to 100 C |
| 27 | Platinum 100 (JIS) | 0 C to 200 C |
| 28 | Platinum 100 (JIS) | 0 C to 400 C |
| 29 | Platinum 100 (JIS) | -200 C to 200 C |
| 2A | Platinum 1000 | -40 C to 160 C |
| 2B | BALCO 500 | -30 C to 120 C |
| 2C | Ni 604 | -80 C to 100 C |
| 2D | Ni 604 | 0 C to 100 C |

IEC RTD 1000, $\alpha = 0.00385$

JIS RTD 1000, $\alpha = 0.00391$

ADAM-4019 command codes against Input ranges table

| Command Code (Hex) | Input Type | Input Range |
|--------------------|-----------------|-----------------|
| 02 | mV | - 100 mA |
| 03 | mV | - 500 mA |
| 04 | V | - 1 V |
| 05 | V | - 2.5 V |
| 08 | V | - 10 V |
| 09 | V | - 5 V |
| 0D | mV | - 20 mA |
| 0E | Thermocouple, J | 0 C to 760 C |
| 0F | Thermocouple, K | 0 C to 1370 C |
| 10 | Thermocouple, T | -100 C to 400 C |
| 11 | Thermocouple, E | 0 C to 1000 C |
| 12 | Thermocouple, R | 500 C to 1750 C |
| 13 | Thermocouple, S | 500 C to 1750 C |
| 14 | Thermocouple, B | 500 C to 1800 C |

Table 4-3 Input Range Code for ADAM-4015 and ADAM-4019

\$AA8Ci

Name Read Single Channel Range Configuration

Description This command read the input type and range configuration of the specified channel in an analog input module.

Syntax \$AA8Ci(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be read.

8 represents the read range configuration command.

Ci represent the specified input channel you want to read.

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

Response !AACiRrr(cr) if the command was 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 indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.

Ci represent the specified input channel you read.

Rrr represent the type and range setting in the specified channel. (refer to Table 4-3 to check range code)

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

Example command: \$028C5(cr)

response: !02C5R21(cr)

The command read the range of channel 5 in the analog input module at address 02. The response "R21" means Pt100 (IEC) 0~100° C.

\$AAXnnnn

| | |
|--------------------|---|
| Name | Watchdog Timer Setting |
| Description | This command set the Watchdog Timer communication cycle. |
| Syntax | <p>\$AAXnnnn(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be read.</p> <p>X represents the setting WDT command.</p> <p>nnnn (range 0000~9999) represent the specified value of communication cycle you want to set.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |

Response

!AA(cr) if the command was 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 indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.

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

Example

command: \$02X1234(cr)

response: !02(cr)

The command set the WDT cycle as 1234 in the input module at address 02.

NOTICE: *If the value of “nnnn” is 0000, the communication WDT function will be disable.*

\$AAY

Name Read Watchdog Timer Setting

Description This command read the setting of Watchdog Timer communication cycle.

Syntax \$AAY(cr)
\$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be read.
Y represents the reading WDT cycle command.
(cr) is the terminating character, carriage return (0Dh).

Response !AAnnnn(cr) if the command was 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 indicates a valid command was received.
? delimiter character indicates the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.
nnnn (range 0000~9999) represent the specified value of communication cycle you read.
(cr) represents terminating character, carriage return (0Dh).

Example command: \$02Y(cr)
response: !020030(cr)
The command read the WDT cycle as 0030 in the input module at address 02.

\$AAS0

| | |
|--------------------|---|
| Name | Internal Calibration |
| Description | This command execute Internal self-calibration for offset and gain errors. |
| Syntax | <p>\$AAS0(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be calibrated.</p> <p>S0 represents the internal calibration system command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command was valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicates a valid command was received.</p> <p>? delimiter character indicates the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |

\$AAS1

| | |
|--------------------|---|
| Name | Reload Default Calibrating Parameter |
| Description | Reload factory default calibrating parameter to overwrite current calibrating parameter |
| Syntax | <p>\$AAS1(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be reloaded.</p> <p>S1 represents the reload calibrating parameter system command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command was valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicates a valid command was received.</p> <p>? delimiter character indicates the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |

\$AAAi

Name CJC Setting

Description This command configure the CJC setting.

Syntax \$AAAi(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address. of the analog input module which is to be read.

Ai represents the command code of CJC setting:

If i=0, this command will stop CJC update.

If i=1, this command will start CJC update.

If i=2, this command will execute CJC update once only.

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

Response !AA(cr) if the command was 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 indicates a valid command was received.

? delimiter character indicates the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.

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

Example command: \$02A1(cr)

response: !02(cr)

The command starts the CJC update in the input module at address 02.

Command Set

4.4.2 Data Conversion and Display Command Set

| Command Syntax | Command Name | Description | I/O Module |
|---------------------------|---|--|------------|
| \$AA3 | Read Source High/Low Values for Linear Mapping | Read the high/low limit values from the specified module for linear mapping. | 4014D |
| \$AA5 | Read Target High/Low Values for Linear Mapping | Read the mapped input high/ low limit values from the specified module for linear mapping. | 4014D |
| \$AA6 (data_A)(data_B) | Write Source High/Low Values for Linear Mapping | Write the high/low limit values to the specified module for linear mapping. The module will only activate the source values after new target high/low values are written (Command \$AA7). | 4014D |
| \$AA7 (data_C)(data_D) | Write Target High/Low Values for Linear Mapping | Write the mapped input high/ low limit values to a specified module for linear mapping. This command is only valid if its was preceded by a \$AA6 command. | 4014D |
| \$AAAV | Enable/Disable Linear Mapping | Enables or disables the linear mapping function of the specified analog input module. | 4014D |
| \$AA8V | Select LED Data Origin | Select whether LED will display data from the input module directly or from the host PC | 4014D |
| \$AA9(sign_data) | Send LED Data | The PC sends data to the module's LED display. This command is valid only after selecting LED to display from PC (\$AA8V) | 4014D |

\$AA3

| | |
|--------------------|--|
| Name | Read Source High/Low Values for Linear Mapping |
| Description | Read the high/low limit values from a specified module for linear mapping. |
| Syntax | <p>\$AA3(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module that you want to interrogate.</p> <p>3 identifies the command to read the module's high/low limit value for linear mapping.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(data_A)(data_B)(cr) if the command is valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>? is a delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>(data_A) is the module's low limit value for linear mapping. The data must consist of an "+" or "-" sign followed by five decimal digits and a fixed decimal point.</p> <p>(data_B) is the module's high limit value for linear mapping. The data must consist of an "+" or "-" sign followed by five decimal digits and a fixed decimal point.</p> <p>(cr) the terminating character, carriage return (0Dh).</p> |

\$AA3**Example**

command: \$133(cr)

response: !13+04.000+20.000(cr)

The module is configured for an ± 20 mA input current range. The linear mapping function should already have been executed. The module's input high/low limit values are +20.000 and +04.000. The command requests the analog input module at address 13 to return its input limit values for linear mapping. The module at address 13 returns its input high/low limit values which are +20.000 and +04.000.

\$AA5

Name Read Target High/Low Values for Linear Mapping

Description Read the mapped input high/ low limit values from a specified module for linear mapping.

Syntax \$AA5(cr)
\$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module that you want to interrogate.
5 is the command to read the mapped high/low limit value for linear mapping.
(cr) is the terminating character, carriage return (0Dh)

Response !AA(data_C)(data_D)(cr) if the command is valid.
?AA(cr) if an invalid operation was entered.
There is no response if the module detects a syntax error or communication error or if the specified address does not exist.
? is a delimiter character indicating the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.
(data_C) is the mapped low limit value for linear mapping. The data must consist of a "+" or "-" sign followed by five decimal digits and a fixed decimal point.
(data_D) is the mapped high limit value for linear mapping. The data must consist of a "+" or "-" sign followed by five decimal digits and a fixed decimal point.
(cr) is the terminating character, carriage return (0Dh).

\$AA5**Example**

command: \$135(cr)
response: !13+000.000+200.00(cr)

The module is configured for a $\pm 20\text{mA}$ input current range. The linear mapping function had been executed and the mapped input high/low limit values were +200.00 and +000.00. The command requests the analog input module at address 13 to return its mapped input limit values for linear mapping. The addressed module returns the mapped input high/low limit values at address 13 are +200.00 and +000.00.

\$AA6(data_A)(data_B)

| | |
|--------------------|--|
| Name | Write Source High/Low Values for Linear Mapping |
| Description | Write the high/low limit values to a specified module for linear mapping. The module will only activate the source values after new target high/low values are written (Command \$AA7). |
| Syntax | <p>\$AA6(data_A)(data_B)(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.</p> <p>6 identifies the set high/low limit value for linear mapping command.</p> <p>(data_A) is the low limit input value of the module for linear mapping. This value must be lower than the module's high limit input value for linear mapping (data_B). The data format is the same as the module's input current range. The minimum value allowed is equal to the minimum input value of the current range.</p> <p>(data_B) is the high limit input value of the module for linear mapping. This value must be greater than the module's low limit input value for linear mapping (data_A). The data format is the same as the module's input current range, and the maximum value allowed is equal to the maximum input value of the current range.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>The addressed module stores the high/low input values in a buffer. The module updates these values only after the command \$AA7(data_C) (data_D) is executed.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>? is a delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |

\$AA6(data_A)(data_B)

Example command: \$136+04.000+20.000(cr)
 response: !13(cr)

The module is configured for a ± 20 mA input current range. The command orders the module at address 13 to change its analog input range from +04.000 mA to +20.000 mA. The addressed module stores these values in a buffer and will only update the high/low limit value of the input current range when command \$137(data_C)(data_D) is executed (see command \$AA7(data_C)(data_D)).

Example command: \$016-100.00+100.00(cr)
 response: !01(cr)

The module is configured for a ± 150 mV input range. The command orders the module at address 01 to change its analog input range from -100.00 mV to +100.00 mV. The addressed module stores these values in a buffer and will only update the high/low limit value of the input current range when command \$017(data_C)(data_D) is executed.

\$AA7(data_C)(data_D)

| | |
|--------------------|---|
| Name | Write Target High/Low Values for Linear Mapping |
| Description | Write the mapped input high/low limit values to a specified module for linear mapping. This command is only valid if it was preceded by a \$AA6 command. |
| Syntax | <p>\$AA7(data_C)(data_D)(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module that you want to interrogate.</p> <p>7 is the command to set the mapped high/low limit value for linear mapping.</p> <p>(data_C) is the low mapped limit input value for linear mapping from the addressed module's low limit input value. This value need not be lower than the mapped high limit input value (data_D). The data format must consist a "+" or "-" sign followed by five decimal digits and a fixed decimal point. The allowed maximum value is 19999.</p> <p>(data_D) is the high mapped limit input value for linear mapping from the addressed module's high limit input value. The data format must include a "+" or "-" sign followed by five decimal digits and a fixed decimal point. The allowed maximum value is 19999.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>The addressed module update the input limit value of the range, and transform the data to a new value by linear mapping. This command is valid only after the command \$AA6(data_A)(data_B) had been executed for each time. Notice that this command will enable linear mapping function.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>? is a delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |

\$AA7(data_C)(data_D)

Example command: \$137+000.00+200.00(cr)
 response: !13(cr)

The module is configured for ± 20 mA input current range. Previously the module executed the command \$136+04.000+20.000, which ordered the module with address 13 to map data from +4.0 mA and +20.0 mA. The current command defines the range (0 and 200) to which these values will be mapped to. From now on, if the input signal is 12 mA, then the output will be 100.00. Notice that the addressed module will also update the input limit values to +04.000 mA and +20.000 mA for the previous command \$136+04.000+20.000. Input signals out of this range will cause invalid output data.

Example command: \$017+100.00-100.00(cr)
 response: !01(cr)

The module is configured for a ± 150 mV input voltage range and already executed command \$016-100.00+100.00. The current command requests the module at address 01 to map data from the earlier specified -100.00 mV and +100.00 mV to the range +100.00 and -100.00. Therefore, if an input signal of 50 mV is received, then the output will be -50.00. Notice that the addressed module will also update the input limit value to be -100.000 mV and +100.00 mV because of the previous command \$136-100.00+100.00. Input signals out of this range will cause invalid output data.

\$AAAV

Name Enable/Disable Linear Mapping

Description Enables or disables the linear mapping function of the specified analog input module.

Syntax \$AAAV(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of analog input module

A selects the linear mapping function

V is either 0 or 1 which determines the linear mapping function to be enable or disabled. 1 means linear mapping function is enable, and 0 means disabled.

(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.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.

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

Example command: \$01A1(cr)

response: !01(cr)

The command enables the linear mapping function of the analog input module at address 01.

\$AA8V

| | |
|--------------------|--|
| Name | Select LED Data Origin |
| Description | Select whether LED will display data from the input module directly or from the host PC |
| Syntax | <p>\$AA8V(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of analog input module that you want to interrogate.</p> <p>8 is the select LED driver command.</p> <p>V is either 1 or 2 to select which is going to send data to the LED. 1 means the driver is the addressed module, and 2 means the host PC will send data..</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>? is a delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$0181(cr)</p> <p>response: !01(cr)</p> <p>The command sets the analog input modules at address 01 to display data sent by the host PC. After this command, the PC can use command \$AA9(data) to send the data to the addressed module.</p> |

\$AA9(data)

Name Send LED Data

Description PC sends data to LED display. This command is valid only after selecting LED to display from PC (\$AA8V)

Syntax \$AA9(data)(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module

9 identifies the PC send data to LED command.

(data) is the data which must consist of an “+” or “-” sign followed by five decimal digits and a fixed decimal point. The maximum value is 19999.

(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

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.

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

Example command: \$019+1999.9(cr)

response: !01(cr)

The command sends display data (+1999.9) to the analog input module at address 01. This command is only valid when it is preceded by command \$0181. (See command \$AA8V.)

Example command: \$019-00290.(cr)

response: !01(cr)

The command sends display data -00290 to the analog input module at address 01. Note that even when sending an integer the data must contain a decimal point.

4.4.3 Analog Input Data Logger Command Set

| Command Syntax | Command Name | Description | I/O Module |
|----------------|----------------------------|---|------------|
| @AACCCSDMTTTT | Set Memory Configuration | Set the channel storage status, standalone mode, data logger mode, storage type and sampling interval for the specified analog input data logger. | 4018M |
| @AAD | Read Memory Configuration | Return the configuration parameters for the specified analog input data logger. | 4018M |
| @AASO | Set Memory Operation Mode | Start/stop the recording function of the memory module. | 4018M |
| @AAT | Read Memory Operation Mode | Read the recording status of the memory module. | 4018M |
| @AAL | Event Record Count | Read the number of stored event records in the memory module. | 4018M |
| @AAN | Standard Record Count | Read the number of stored standard records in the memory module. | 4018M |
| @AARNNNN | Read Record Content | Read the contents of the specified record. | 4018M |
| @AACSDHHHTEIII | Set Alarm Limit | Set the high/low alarm settings for the specified channel. | 4018M |
| @AABC | Read Alarm Limit | Read the high/low alarm settings for the specified channel. | 4018M |

@AACCCSDMTTTT

Name Set Memory Configuration
Description Sets the channel storage status, standalone mode, data logger mode storage type and sampling interval for the specified analog input data logger.

Syntax @AACCCSDTTTT(cr)
 @ is a delimiter character.
 AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.
 C identifies the Set Memory Configuration command.
 CC (range 00-FF) represents the data storage status of each channel. The ADAM-4018M has 8 channels, bit 0 representing channel 0, and bit 7 representing channel 7. A mask bit value of '1' enables data storage in the specified channel, while a mask bit value of '0' disables data storage.

Channel 7

Channel 0

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|

S represents the standalone mode. In order for the ADAM-4018M to operate in the field, you must power on the memory module by setting this value to '1.' Otherwise, the data will not be recorded.

D represents the data logging mode. '0' enables Standard Mode, where all eight channels record the normal data according to the sampling interval. '1' enables Event Mode, where all eight channels record the data if its value is over the High Alarm limit or under the Low Alarm limit. '2' enables Mixed Mode, where channels 0 - 3 act as the standard logger and the channels 4 - 7 act as the event logger.

M represents the storage type. "0" represents writing to the end of memory. "1" represents circular memory mode.

TTTT (range 2-65535) represents the sampling interval in seconds.

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

@AACCCSDMTTTT

- Response** !AA(cr) if the configuration is successful.
 ?AA(cr) if the configuration fails.
 ! and ? are delimiter characters.
 AA (range 00-FF) represents the 2-character hexadecimal
 address of an analog input module.
- Example** command: @0DCFF111012C(cr)
 response: !0D(cr)
- The ADAM-4018M module at address 0D is configured as
 such:
 All eight data storage channels enabled
 Standalone mode enabled
 Event logger selected
 Circular memory mode
 Sampling interval 300 seconds
 The response indicates the command is successful.

@AAD

Name Read Memory Configuration

Description The command requests the configuration data from the analog input data logger at address AA.

Syntax @AAD (cr)

@ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.

D identifies the Read Memory Configuration command.

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

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

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

AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.

CC (range 00-FF) represents the data storage status of each channel. The ADAM-4018M has 8 channels, bit 0 representing channel 0, and bit 7 representing channel 7. A mask bit value of '1' enables data storage in the specified channel, while a mask bit value of '0' disables data storage.

S represents the standalone mode. In order for the ADAM-4018M to operate in the field, you must power on the memory module by setting this value to '1.' Otherwise, the data will not be recorded.

D represents the data logging mode. '0' enables Standard Mode, where all eight channels record the normal data according to the sampling interval. '1' enables Event Mode, where all eight channels record the data if its value is over the High Alarm limit or under the Low Alarm limit. '2' enables Mixed Mode, where channels 0 - 3 act as the standard logger and the channels 4 - 7 act as the event logger.

TTTT (range 2-65535) represents the sampling interval in seconds.

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

@AASO

Name Set Memory Operation Mode

Description Sets the operation mode of the analog input data logger at address AA to Start or Stop.

Syntax @AASO(cr)
@ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.
S identifies the Set Memory Operation Mode command.
O represents the operation mode: '1' enables the recording of data. '0' disables the recording of data.
(cr) is the terminating character, carriage return (0Dh).

Response !AA(cr) if the command is valid.
?AA(cr) if an invalid parameter 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.
? is a delimiter character indicating the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.
(cr) is the terminating character, carriage return (0Dh).

Example command: @03S1(cr)
response: !03(cr)
The command enables the analog input data logger at address 03 to record data.
The response indicates that the command was received.

@AAT

| | |
|--------------------|---|
| Name | Read Memory Operation Mode |
| Description | Request the memory operation status of the analog input data logger at address AA. |
| Syntax | <p>@AAT (cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.</p> <p>T identifies the Read Memory Operation Mode command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AAO(cr) if the command is valid.</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>O represents the operation mode: '1' enables the recording of data. '0' disables the recording of data.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @F3T(cr)</p> <p>response: !F31(cr)</p> <p>The command requests the memory operation status of the analog input data logger at address F3.</p> <p>The response indicates that data recording is enabled.</p> |

@AAL

| | |
|--------------------|--|
| Name | Event Record Count |
| Description | Request the number of event records stored in the analog input data logger at address AA. |
| Syntax | <p>@AAL (cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.</p> <p>L identifies the Event Record Count command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AAHHHH(cr) if the command is valid.</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>HHHH represents the 4-character hexadecimal number of event records stored in the analog input data logger at address AA.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @F3L(cr)</p> <p>response: !F30096(cr)</p> <p>The command requests the number of event records stored in the analog input data logger at address F3.</p> <p>The module currently has 150 event records.</p> |

@AAN

| | |
|--------------------|---|
| Name | Standard Record Count |
| Description | Request the number of standard records stored in the analog input data logger at address AA. |
| Syntax | <p>@AAN (cr)</p> <p>@ is a delimiter character</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.</p> <p>N identifies the Standard Record Count command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AAHHHH(cr) if the command is valid.</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>HHHH represents the 4-character hexadecimal number of data records stored in the analog input data logger at address AA.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @A3N(cr)</p> <p>response: !A30320(cr)</p> <p>The command requests the number of data records stored in the analog input data logger at address A3.</p> <p>The module currently has 800 data records.</p> |

@AARNNNN

Name Read Record Content

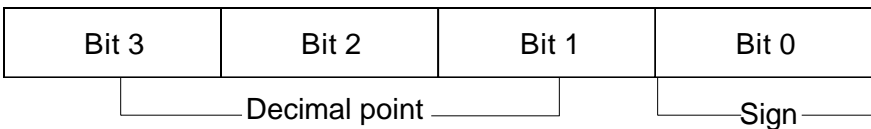
Description Request the content of record NNNN stored in the analog input data logger at address AA.

Syntax @AARNNNN (cr)
@ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.
R identifies the Read Record Content command.
NNNN represents the 4-character decimal number of stored record index. Its value is from 0 to (total record number - 1).
(cr) is the terminating character, carriage return (0Dh).

NOTE:

NNNN is from 0 to 9999 for Standard Mode.
NNNN is from 0 to 4599 for Event Mode.
NNNN is from 0 to 7299 for Mixed Mode. (0-4999 are data records, 5000-7299 are event records)

Response !AACDHHHH(cr) if the returned data are data records.
!AACDHHHHHTTTTTTTT(cr) if the returned data are event records.
! is a delimiter character indicating a valid command was received.
AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.
C represents the channel index. Its value is from 0 to 7.
D represents the 4-bit binary number. Bit 0 represents the sign of the number HHHH. '0' means positive. '1' means negative.
Bits 1-3 represents the decimal point of the number HHHH.



@AARNNNN

HHHH represents the 4-character hexadecimal number of returned record stored in the analog input data logger at address AA.

TTTTTTTTT represents elapsed time.

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

Example

command: @F3R1000(cr)

response: !F30799AA00001000(cr)

The command requests the analog input data logger at address F3 to return its contents in the 1001st record.

The returned content is valid. The event data number is - 39.338 for channel 0 in the 4096 seconds from the start of the module.

@AAACSDHHHTEIII

| | |
|--------------------|---|
| Name | Set Alarm Limit |
| Description | Set high/low alarm limits for the channel C in the analog input data logger at address AA |
| Syntax | <p>@AAACSDHHHTEIII(cr)</p> <p>@ is a delimiter character</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.</p> <p>A identifies the Set Alarm Limit command.</p> <p>C represents the channel index. Its value is from 0 to 7.</p> <p>S represents the sign of the high alarm limit. '0' is for positive. And '1' is for negative.</p> <p>D represents the decimal point of the high alarm limit. Its value is from 0 to 5.</p> <p>HHHH represents the 4-character hexadecimal number of high alarm limit.</p> <p>T represents the sign of the low alarm limit. '0' is for positive. And '1' is for negative.</p> <p>E represents the decimal point of the low alarm limit. Its value is from 0 to 5.</p> <p>IIII represents the 4-character hexadecimal number of low alarm limit.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>?AA(cr) if an invalid parameter was entered. There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>? is a delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |

@AAACSDHHHTEIII

Example command: @EFA0020400020100(cr)

 response: !EF(cr)

The command sets channel 0 of the analog input data logger at address EF as such:

high alarm limit = 10.24

low alarm limit = 2.56

The response indicates the command was received.

@AABC

| | |
|--------------------|---|
| Name | Read Alarm Limit |
| Description | Request the alarm limits for the specified channel in the analog input data logger at address AA. |
| Syntax | <p>@AABC(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.</p> <p>B identifies the Read Alarm Limit command.</p> <p>C represents the channel index. Its value is from 0 to 7.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AASDHHHTEIII</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input data logger.</p> <p>S represents the sign of the high alarm limit. '0' is for positive. And '1' is for negative.</p> <p>D represents the decimal point of the high alarm limit. Its value is from 0 to 5.</p> <p>HHHH represents the 4-character hexadecimal number of high alarm limit.</p> <p>T represents the sign of the low alarm limit. '0' is for positive. And '1' is for negative.</p> <p>E represents the decimal point of the low alarm limit. Its value is from 0 to 5.</p> <p>IIII represents the 4-character hexadecimal number of low alarm limit.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |

4.4.4 Digital I/O, Alarm and Event Command Set

| Command Syntax | Command Name | Description | I/O Module |
|----------------|-----------------------------------|---|--------------------------------|
| @AADI | Read Digital I/O and Alarm Status | The addressed module returns the state of its digital input and digital output channels and the status of its alarm | 4011, 4011D, 4012, 4014D, 4016 |
| @AADO(data) | Set Digital Output Values | Set the values of the module's digital outputs (ON or OFF) | 4011, 4011D, 4012, 4014D, 4016 |
| @AAEAT | Enable Alarm | Enables the alarm in either Momentary or Latching mode | 4011, 4011D, 4012, 4014D, 4016 |
| @AAHI(data) | Set High Alarm Value | Downloads the High alarm limit value | 4011, 4011D, 4012, 4014D, 4016 |
| @AALO(data) | Set Low Alarm Value | Downloads the Low alarm limit value | 4011, 4011D, 4012, 4014D, 4016 |
| @AADA | Disable Alarm | Disables all alarm functions | 4011, 4011D, 4012, 4014D, 4016 |
| @AACA | Clear Latch Alarm | Resets the module's Latch alarm to zero | 4011, 4011D, 4012, 4014D, 4016 |
| @AARH | Read High Alarm Value | Ask the addressed module to return its high alarm value | 4011, 4011D, 4012, 4014D, 4016 |
| @AARL | Read Low Alarm Value | Ask the addressed module to return its low alarm value | 4011, 4011D, 4012, 4014D, 4016 |
| @AARE | Read Event Counter | Ask the addressed module to return its event counter value | 4011, 4011D, 4012, 4014D |
| @AACE | Clear Event Counter | Reset the module's event counter to zero | 4011, 4011D, 4012, 4014D |

@AADI

| | |
|--------------------|---|
| Name | Read Digital I/O and Alarm State |
| Description | The addressed analog input module is instructed to return the value of its digital input and output channels and the state of its alarm (Momentary or Latching). |
| Syntax | <p>@AADI(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>DI is the Read Digital I/O and Alarm Status command.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Response | <p>!AASOOII(cr) if the command was valid (ADAM-4011/4011D/4012/4014D).</p> <p>!AASOO00(cr) if the command was valid (ADAM-4016)</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicating a valid command was received.</p> <p>AA represents the 2-character hexadecimal address of the responding analog input module.</p> <p>S hexadecimal number that represents the alarm state (0h = disabled, 1h = MOMENTARY mode enabled, 2h = LATCH mode enabled).</p> <p>OO (for ADAM-4011/4011D/4012/4014D) is a hexadecimal number representing the Digital Output port's channel 0 and 1 status (00h = D/O channels 0 and 1 are both OFF, 01h = channel 0 is ON, channel 1 is OFF, 02h = channel 0 is OFF, channel 1 is ON, 03h = channel 0 and 1 are both ON).</p> |

@AAI

OO (for ADAM-4016) is a hexadecimal number representing the status of the four digital output channels. The corresponding table is shown in the following table:

| Status Code | D00 | D01 | D02 | D03 |
|-------------|-----|-----|-----|-----|
| 00 | OFF | OFF | OFF | OFF |
| 01 | OFF | OFF | OFF | ON |
| 02 | OFF | OFF | ON | OFF |
| 03 | OFF | OFF | ON | ON |
| 04 | OFF | ON | OFF | OFF |
| 05 | OFF | ON | OFF | ON |
| 06 | OFF | ON | ON | OFF |
| 07 | OFF | ON | ON | ON |
| 08 | ON | OFF | OFF | OFF |
| 09 | ON | OFF | OFF | ON |
| 0A | ON | OFF | ON | OFF |
| 0B | ON | OFF | ON | ON |
| 0C | ON | ON | OFF | OFF |
| 0D | ON | ON | OFF | ON |
| 0E | ON | ON | ON | OFF |
| 0F | ON | ON | ON | ON |

II is a hexadecimal number representing the Digital input port's channel status(00h = D/I channel is Low, 01h = channel is High).

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

Example

command: @15DI(cr)

response: !510001(cr)

The analog input module at address 15h is instructed to return digital I/O data and alarm status.

The module responds that both digital output channels are OFF, digital input is HIGH, and alarm state is Momentary.

@AADO

| | |
|--------------------|--|
| Name | Set Digital Output |
| Description | Sets the values of the module's digital outputs (ON or OFF). |
| Syntax | <p>@AADO(data)(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>DO is the Set Digital Output command</p> <p>(data) is the two-character parameter that sets the state for the digital output bits of the module, as shown below:</p> <p>ADAM-4011/4011D/4012/4014D:</p> <p>00 all D/O bits are OFF</p> <p>01 DO0 is ON, DO1 is OFF</p> <p>02 DO0 is OFF, DO1 is ON</p> <p>03 all bits are ON</p> <p>ADAM-4016:</p> <p>00 DO0 and DO1 are OFF</p> <p>01 DO0 is ON, DO1 is OFF</p> <p>02 DO0 is OFF, DO1 is ON</p> <p>03 DO0 and DO1 are ON</p> <p>10 DO2 and DO3 are OFF</p> <p>11 DO2 is ON, DO3 is OFF</p> <p>12 DO2 is OFF, DO3 is ON</p> <p>13 DO2 and DO3 are ON</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command was valid.</p> <p>?AA(cr) if an invalid parameter was entered.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist</p> <p>!delimiter character indicating a valid command was received.</p> <p>?delimiter character indicating the command was invalid.</p> <p>AA represents the 2-character hexadecimal address of the responding analog input module.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |

@AADO

Example command: @05DO01(cr)
 response: !05(cr)

The analog input module at address 05h is instructed to set digital output channel 1 to ON and digital output channel 2 to OFF. The module confirms the settings.

@AAEAT

| | |
|--------------------|---|
| Name | Enable Alarm |
| Description | The addressed analog input module is instructed to enable its alarm in either Latching or Momentary mode. |
| Syntax | <p>@AAEAT(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>EA is the Enable Alarm command.</p> <p>T indicates alarm type and can have the value M = Momentary alarm state, or L = Latching alarm state.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command was valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicating a valid command was received.</p> <p>AA represents the 2-character hexadecimal address of the responding analog input module.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |

NOTICE: *An analog input module requires a maximum of 2 seconds after it received an Enable Alarm command to let the settings take effect . During this interval, the module can not be addressed to perform any other actions.*

@AAEAT

Example command: @03EAL(cr)
 response: !03(cr)

The analog input module at address 03h is instructed to enable its alarm in Latching mode.

The module confirms that the command has been received.

@AAHI

| | |
|--------------------|---|
| Name | Set High Alarm Limit |
| Description | Downloads High alarm limit value into the addressed module. |
| Syntax | <p>@AAHI(data)(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>HI is the Set High Limit command.</p> <p>(data) represent the value of the desired high limit setting. The format is always engineering units.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command was valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist</p> <p>!delimiter character indicating a valid command was received.</p> <p>AA represents the 2-character hexadecimal address of the responding analog input module.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @04HI+080.00(cr)</p> <p>response: !04(cr)</p> <p>Presume the analog input module at address 04h is configured to accept T-type thermocouple input. The command will set the High alarm limit to 80°C.</p> <p>The module responds that the command has been received.</p> |

NOTICE: *An analog input module requires a maximum of 2 seconds after it received an Set High Alarm command to let the settings take effect . During this interval, the module can not be addressed to perform any other actions.*

@AALO

Name Set Low Alarm Limit

Description Downloads Low alarm limit value into the addressed module.

Syntax @AALO(data)(cr)
 @ is a delimiter character.
 AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.
 LO is the Set Low Limit command.
 (data) represent the value of the desired low limit setting. The format is always engineering units.
 (cr) represents terminating character, carriage return (0Dh).

Response !AA(cr) if the command was valid.
 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.
 AA represents the 2-character hexadecimal address of the responding analog input module.
 (cr) represents terminating character, carriage return (0Dh).

Example command: @04LO-020.00(cr)
 response: !04(cr)
 Presume the analog input module at address 04h is configured to accept T-type thermocouple input. The command will set the Low alarm limit to -20°C.
 The module responds that the command has been received.

NOTICE: *An analog input module requires a maximum of 2 seconds after it received an Set Low Alarm command to let the settings take effect . During this interval, the module can not be addressed to perform any other actions.*

@AADA

| | |
|--------------------|--|
| Name | Disable Alarm |
| Description | Disables all alarm functions of the addressed analog input module. |
| Syntax | @AADA(cr) @ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module. DA is the Disable Alarm command. (cr) represents terminating character, carriage return (0Dh). |
| Response | !AA(cr) if the command was valid. There is no response if the module detects a syntax error or communication error or if the specified address does not exist. ! delimiter character indicating a valid command was received. AA represents the 2-character hexadecimal address of the responding analog input module. (cr) represents terminating character, carriage return (0Dh). |
| Example | command: @07DA (cr) response: !07(cr) The analog input module at address 07h is instructed to disable all alarm functions. The module confirms its alarm functions have been disabled. |

NOTICE: *An analog input module requires a maximum of 2 seconds after it received an Disable Alarm command to let the settings take effect. During this interval, the module can not be addressed to perform any other actions.*

@AACA

| | |
|--------------------|---|
| Name | Clear Latch Alarm |
| Description | Both alarm states (High and Low) of the addressed analog input module are set to OFF, no alarm. |
| Syntax | <p>@AACA(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>CA is the Clear Latch Alarm command.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command was valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! a delimiter character indicating a valid command was received.</p> <p>AA represents the 2-character hexadecimal address of the responding analog input module.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @05CA(cr)</p> <p>response: !05(cr)</p> <p>The analog input module at address 05h is instructed to set both alarm states (High and Low) to OFF.</p> <p>The module confirms it has done so accordingly.</p> |

@AARH

| | |
|--------------------|--|
| Name | Read High Alarm Limit |
| Description | The addressed module is asked to return its High alarm limit value. |
| Syntax | <p>@AARH(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>RH is the Read High Alarm Limit command.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(data)(cr) if the command was valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicating a valid command was received.</p> <p>AA represents the 2-character hexadecimal address of the responding analog input module.</p> <p>(data) is the value of the High alarm limit in engineering units.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @07RH(cr)</p> <p>response: !07+2.0500(cr)</p> <p>Presume the analog input module at address 07h is configured to accept 5 V input. The command instructs the module to return it High alarm limit value.</p> <p>The module responds its High alarm limit value is 2.0500 V.</p> |

@AARL

| | |
|--------------------|--|
| Name | Read Low Alarm Limit |
| Description | The addressed module is asked to return its Low alarm limit value. |
| Syntax | @AARL(cr) @ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module. RL is the Read Low Alarm Limit command. (cr) represents terminating character, carriage return (0Dh). |
| Response | !AA(data)(cr) if the command was valid. There is no response if the module detects a syntax error or communication error or if the specified address does not exist. ! delimiter character indicating a valid command was received. AA represents the 2-character hexadecimal address of the responding analog input module. (data) is the value of the Low alarm limit in engineering units. (cr) represents terminating character, carriage return (0Dh). |
| Example | command: @05RL(cr) response: !05-0.3750(cr) Presume the analog input module at address 05h is configured to accept 1 V input. The command instructs the module to return its Low alarm limit value. The module responds its Low alarm limit value is -0.3750 V. |

@AARE

| | |
|--------------------|---|
| Name | Read Event Counter |
| Description | The addressed module is instructed to return its event counter value. |
| Syntax | <p>@AARE(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.</p> <p>RE Read Event Counter command.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(data)(cr) if the command was valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicating a valid command was received.</p> <p>AA represents the 2-character hexadecimal address of the responding analog input module.</p> <p>(data) represents the stored value, from '00000' to '65535' (The max value that can be held by the counter register is 65535). The number 65535 is held when the actual total counts exceed this number.</p> <p>(cr) represents terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @08RE(cr)</p> <p>response: !0832011(cr)</p> <p>The command instructs the module at address 08h to return its counter value.</p> <p>The module responds that its counter value equals 32011.</p> |

@AACE

| | |
|--------------------|--|
| Name | Clear Event Counter |
| Description | The addressed module is instructed to reset its event counter to zero. |
| Syntax | @AACE(cr) @ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module. CEClear Event Counter command. (cr)represents terminating character, carriage return (0Dh). |
| Response | !AA(cr) if the command was valid. There is no response if the module detects a syntax error or communication error or if the specified address does not exist. !delimiter character indicating a valid command was received. AA represents the 2-character hexadecimal address of the responding analog input module. (cr)represents terminating character, carriage return (0Dh). |
| Example | command: @09CE(cr) response: !09(cr) The command instructs the module at address 09h to set its event counter to zero. The module responds that its counter has been reset. |

4.4.5 Excitation Voltage Output Command Set

| Command Syntax | Command Name | Description | I/O module |
|----------------|---------------------------------------|--|------------|
| \$AA6 | Get Excitation Voltage Output Value | Returns either last value sent to specified module by \$AA7 command, or start-up output voltage. | 4016 |
| \$AA7 | Excitation Voltage Output | Direct output excitation voltage data to a specified module | 4016 |
| \$AAS | Start-up Voltage Output Configuration | Stores a default value in a specified module. The output value will take effect upon startup. | 4016 |
| \$AAE | Trim Calibration | Trims specified module a number of units up/down | 4016 |
| \$AAA | Zero Calibration | Tells the module to store parameters for zero calibration | 4016 |
| \$AAB | Span Calibration | Tells the module to store parameters for span calibration | 4016 |

\$AA6

Name Get Excitation Voltage Output Value

Description The addressed strain gauge input module is instructed to return the latest output value it received from Excitation Voltage Output command. If the module hasn't received an Excitation Voltage Output command since startup, it will return its Start-up Output value.

Syntax \$AA6(cr)

\$ is a delimiter character.

AA(range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module.

6 is the get excitation voltage output command.

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

Response !AA(data)(cr) if the command is valid.

?AA(cr) if an invalid command was issued.

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

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module.

(data) is the value that is returned by analog output channel.

The format of the data is in engineering unit.

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

Example command: \$0A6(cr)

response: !0A+03.000(cr)

The command tells the strain gauge input module at address 0Ah to return the last excitation voltage output value it received from an Excitation Voltage Output command.

The strain gauge input module returns the value +03.000V.

\$AA7

Name Excitation Voltage Output

Description Send a value to the analog output channel of the addressed strain gauge input module. Upon receipt, the analog output channel will output this value.

Syntax \$AA7(data)(cr)

\$ is a delimiter character.

AA(range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module.

7 is the excitation voltage output command.

(data) is the value that has to be output through the analog output channel. The data format is engineering unit and the range is between 0 to 10 V.

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

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

?AA(cr) if a value was sent that is out of range.

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

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module.

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

Example command: \$337+05.000(cr)

response: !33(cr)

The command sends the value +05.000V to the analog output channel of the strain gauge input module at address 33h. The module responds that the command is valid. Its output data format is in engineering unit, the value is +05.000V.

\$AAS

| | |
|--------------------|--|
| Name | Start-up Voltage Output Configuration |
| Description | Stores the present analog output value of the strain gauge input module with address AA in the module's non-volatile register. The output value will take effect upon start-up or after a brownout. |
| Syntax | <p>\$AAS(cr)</p> <p>\$ is a delimiter character.</p> <p>AA(range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module whose analog output is to be set.</p> <p>S is the start-up voltage output configuration command.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>?AA(cr) if an invalid command was issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicating a valid command was received.</p> <p>? delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |

NOTICE: *A strain gauge input module requires a maximum of 6 milliseconds after it received a Startup Voltage Output Configuration command to let the settings take effect. During this interval, the module can not be addressed to perform any other actions.*

\$AAS**Example**

command: \$0AS(cr)

response: !0A(cr)

Presume the present output value of the output channel of the strain gauge input module with address 0A is +05.000V. The command tells the module store the present output value, in its non-volatile memory. When the module is powered up or reset, its default output value will be +05.000V.

The response of the strain gauge input module indicates that the command has been received.

\$AAE

Name Trim Calibration

Description Trims the output voltage of the strain gauge input module a specified number of units up or down.

Syntax \$AAE(number of counts)(cr)

\$ is a delimiter character.

AA(range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module to be calibrated.

E is the trim calibration command.

(number of counts) is the 2-character twos complement hexadecimal value that represents the number of counts by which to increase or decrease the output voltage. Each count equals approximately 1mV. Values range from 00 to 7F and from 80 to FF, where 00 represents 0 counts, 7F represents 127 counts, 80 represents -128 counts and FF represents -1 counts. Negative values decrease and positive values increase the output voltage according to the number of counts.

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

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

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

! delimiter character indicating a valid command was received.

AA (range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module.

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

Example command: \$07E14(cr)

response: !07

The command tells the analog output of the strain gauge input module at address 07h to increase its output value by 20 (14h) counts which is approximately 20 mV.

The strain gauge input module confirms the increase.

In order to perform this trim calibration, a voltmeter should be connected to the module's output. (See also the zero calibration command and span calibration command of the strain gauge input module and Chapter 5, Calibration, for a detailed description.)

\$AAA

Name Zero Calibration

Description Stores the voltage output value of the addressed strain gauge input module as zero voltage reference.

Syntax \$AAA(cr)
\$ is a delimiter character.
AA(range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module who’s output channel is to be calibrated.
A is 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 command was issued.
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 the strain gauge input module.
(cr) is the terminating character, carriage return (0Dh)

Before issuing the Zero Calibration command, the analog output should be trimmed to the correct value with the Trim Calibration command. A voltmeter should be connected to the module’s output channel.(See also the strain gauge input module’s Trim Calibration command and Chapter 5, Calibration, for a detailed description.)

\$AAB**Name** Span Calibration**Description** Stores the voltage output value of the addressed strain gauge input module as 10V reference.**Syntax** \$AAB(cr)

\$ is a delimiter character.

AA(range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module who's output channel is to be calibrated.

B is the span calibration command.

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

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

?AA(cr) if an invalid command was issued.

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

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the strain gauge input module.

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

Before issuing the Span Calibration command, the analog output should be trimmed to the correct value with the Trim Calibration command. A voltmeter should be connected to the module's output channel.(See also the strain gauge input module's Trim Calibration command and Chapter 5, Calibration, for a detailed description.)

4.5 Analog Output Module Command

| Command Syntax | Command Name | Description | I/O Module |
|--------------------------|--|---|------------|
| %AANNTTCFF | Configuration | Sets the address, output range, baud rate, data format, slew rate and/or checksum status | 4021 |
| #AA(data) | Analog Data Out | Directs output data to a specified module | 4021 |
| \$AA4 | Start-Up Output Current or Voltage Configuration | Stores a default output value in a specified module. The output value will take effect upon startup | 4021 |
| \$AA3 (number of counts) | Trim Calibration | Trims specified module a number of units up/down | 4021 |
| \$AA0 | 4 mA Calibration | Tells the module to store parameters for 4mA Calibration | 4021 |
| \$AA1 | 20 mA Calibration | Tells the module to store parameters for 20 mA Calibration | 4021 |
| \$AA2 | Configuration Status | Reads the configuration of specified module | 4021 |
| \$AA6 | Last Value Readback | Returns either last value sent to specified module by #AA command, or start-up output current/voltage | 4021 |
| \$AA8 | Current Readback | Returns measured value of the current/voltage flowing through current loop | 4021 |
| \$AA5 | Reset Status | Checks if module has been reset since the last \$AA5 command | 4021 |
| \$AAF | Read Firmware Version | Returns the firmware version code from the specified analog output module | 4021 |
| \$AAM | Read Model Name | Return the module name from specified analog output module | 4021 |

%AANNTTCCFF

Name Configuration

Description Sets address, input range, baud rate, data format, checksum status, and/or integration time for an analog output module.

Syntax %AANNTTCCFF(cr)
% is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module to be configured.

NN represents the new hexadecimal address of the analog output module. Range is from 00h to FFh.

TT represents the type (output range) code. (See table 4-4)

CC represents the baud rate code. (See table 4-3).

FF is a hexadecimal number that equals the 8-bit parameter representing the status of data format, slew rate, and checksum. The layout of the 8-bit parameter is shown in figure 4-2. Bit 7 is not used and must be set to 0.

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

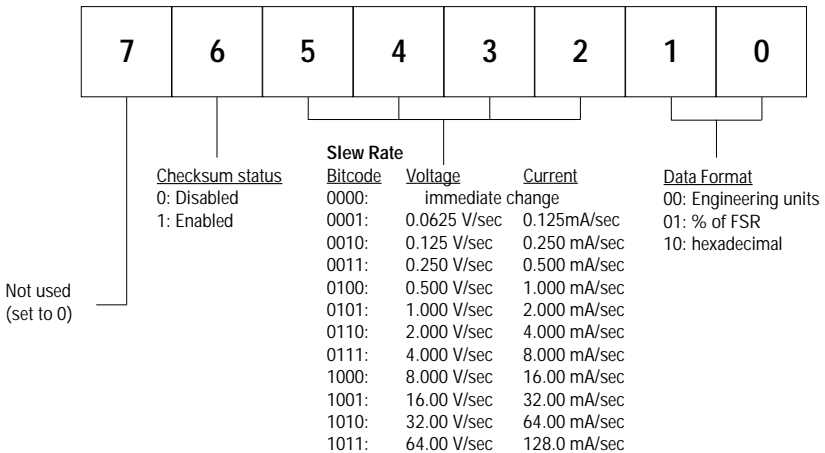


Figure 4-2 Data format for 8-bit parameter

%AANNTTCCFF

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

Table 4-3
Baudrate Codes

| Baud Rate Code (Hex) | Baud Rate |
|----------------------|-----------|
| 03 | 1200 bps |
| 04 | 2400 bps |
| 05 | 4800 bps |
| 06 | 9600 bps |
| 07 | 19.2 Kbps |
| 08 | 38.4 Kbps |

Table 4-4
Output Range Code (Type Code)

| Output Range Code (Hex) | Output Range for ADAM 4021 |
|-------------------------|----------------------------|
| 30 | 0 to 20 mA |
| 31 | 4 to 20 mA |
| 32 | 0 to 10 V |

NOTICE: *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 can not be addressed to perform any other actions.*

NOTICE: *All configuration parameters can be changed dynamically, except the checksum and baud rate parameters. They can only be altered when the INIT* terminal is grounded. (Refer to Baud rate and Checksum in Chapter 2).*

%AANNTTCFF

Example command: %310A310610(cr)
 response: !24(cr)

The analog input module at addresses 31h is configured to a new address 0Ah, output range 4 to 20 mA, baud rate 9600, engineering units data format, a slew rate of 1.0 mA/sec and no checksum checking.

The response indicates that the command has been received.

#AA

Name Analog Data Out

Description Send a value to the addressed analog output module. Upon receipt, the analog output module will output this value.

Syntax #AA(data)(cr)

is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog output module.

(data) is the value that has to be output through the analog output module. Range and value depend on the module's configured data format. Possible formats are: engineering units, % of FSR or hexadecimal. (See also Appendix B, Data Formats and I/O Ranges.)

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

Response

>(cr) if the command was 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.

>delimiter character.

?delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog output module.

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

#AA

Example command: #3315.000(cr)

response: >(cr)

The command sends the value 15 mA to the analog output module at address 33h that is configured for engineering units.

The analog output module responds that the command was valid. Its output value will be 15 mA.

command: #0A+030.00(cr)

response: >(cr)

The module is configured for a 4 to 20 mA output range and a percent of span data format. The command sends the value 8.8 mA ($4 \text{ mA} + 0.30 \times 16 \text{ mA} = 8.8 \text{ mA}$) 30% of the span to the analog output module at address 0Ah.

The analog output module responds that the command was valid. Its output value is set to 8.8 mA.

command: #1B7FF(cr)

response: >(cr)

The command sends the hexadecimal value 7FF to the analog output module at address 1Bh. The module is configured for a 0 to 20 mA output range and a hexadecimal data format. It will output 10 mA ($(7FFH/FFFH) \times 20 \text{ mA} = 10 \text{ mA}$).

\$AA4

| | |
|--------------------|--|
| Name | Start-up Voltage/Current Output Configuration |
| Description | Stores the present output value of an analog output module with address AA in the module's non-volatile register. The output value will take effect upon start-up or after a brownout. |
| Syntax | <p>\$AA4(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module whose output current is to be set.</p> <p>4 is the Start-up Output Current Configuration command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>?AA(cr) if an invalid command was issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicating a valid command was received.</p> <p>? delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$0A4(cr)</p> <p>response: !0A(cr)</p> <p>Presume the present output value of the analog output module with address 0A is 9.4 mA. The command tells the analog output module store the present output value, in its non-volatile memory. When the module is powered up or reset, its default output value will be 9.4mA.</p> <p>The response of the analog output module indicates that the command has been received.</p> |

NOTICE: *An analog output module requires a maximum of 6 milliseconds after it received a Startup Voltage/Current Output Configuration command to let the settings take effect. During this interval, the module can not be addressed to perform any other actions.*

\$AA3**Name** Trim Calibration**Description** Trims the addressed analog output module a specified number of units up or down.**Syntax** \$AA3(number of counts)(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module to be calibrated.

3 is the trim calibration command.

(number of counts) 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 1.5 μ A. 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 count. Negative values decrease and positive numbers increase the output current according to the number of counts.

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

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

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

! delimiter character indicating a valid command was received

AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module.

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

\$AA3

Example command: \$07314(cr)
 response: !07(cr)

The command tells the analog output module at address 07h to increase its output value by 20 (14h) counts which is approximately 30 μ A.

The analog output module confirms the increase.

In order to perform this trim calibration, either a millimeter or a resistor and voltmeter should be connected to the module's output. (See also the 4 mA Calibration command and the 20 mA Calibration command of the analog output module's command set. Refer also to Chapter 5, Calibration for a detailed description.)

\$AA0

Name 4 mA Calibration command

Description Stores the current output value of the addressed analog output module as 4 mA reference.

Syntax \$AA0(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module who's data is to be sent.

0 is the 4 mA Calibration command.

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

Response !AA(cr) if the command was valid.

?AA(cr) if an invalid command was issued.

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

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module that is responding.

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

Before issuing the 4 mA Calibration command, the analog output module should be trimmed to the correct value with the Trim Calibration command. Either a millimeter or a resistor and voltmeter should be connected to the module's output. (See also the analog output module's Trim Calibration command and Chapter 5, Calibration, for a detailed description.)

\$AA1

| | |
|--------------------|---|
| Name | 20 mA Calibration command |
| Description | Stores the current output value of the addressed analog output module as 20 mA reference. |
| Syntax | <p>\$AA1(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module who's data is to be sent.</p> <p>1 is the 20 mA Calibration command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command was valid.</p> <p>?AA(cr) if an invalid command was issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exists.</p> <p>! delimiter character indicating a valid command was received.</p> <p>? delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module that is responding.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |

Before issuing the 20 mA Calibration command, the analog output module should be trimmed to the correct value with the Trim Calibration command. Either a millimeter or a resistor and voltmeter should be connected to the module's output. (See also the analog output module's Trim Calibration command and Chapter 5, Calibration, for a detailed description.)

\$AA2

Name Read Configuration Status command

Description Instruct the addressed analog output module to return its configuration data.

Syntax \$AA2(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module to be interrogated.

2 is the Read Configuration Status command.

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

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

?AA(cr) if an invalid command was issued.

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

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of an analog output module.

TT represents the type code, which determines the output range.

CC represents the baud rate code.

FF is a hexadecimal number that equals the 8-bit parameter representing the data format, slew rate, and checksum status.

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

For tables, reference the %AANNTTCCFF Configuration command.

Example command: \$452(cr)

response: !45300614(cr)

The command asks the analog output module at address 45h to send its configuration data.

The analog output module at address 45h responds with output range 0 to 20 mA, baud rate 9600, engineering units as the currently configured data format, slew rate 2 mA per second, and no checksum checking.

\$AA6

| | |
|--------------------|---|
| Name | Last Value Readback |
| Description | The addressed analog output module is instructed to return the latest output value it received from an Analog Data Out command. If the module hasn't received an Analog Data Out command since startup, it will return its Start-up Output value. |
| Syntax | <p>\$AA6(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module whose value you want to return.</p> <p>6 is the Last Value Readback command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(data)(cr) if the command was valid.</p> <p>?AA(cr) if an invalid command was issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>!delimiter character indicating a valid command was received.</p> <p>?delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module.</p> <p>(data) is the value that is returned by the analog output module. The format of the data depends on the module's configured data format.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$0A6(cr)</p> <p>response: !0A03.000(cr)</p> <p>The command tells the analog output module at address 0Ah to return the last output value it received from an Analog Out command.</p> <p>The analog output module returns the value 3.000 mA.</p> |

\$AA8**Name** Current Readback**Description** The addressed analog output module is instructed to measure the current flowing through its current/voltage loop and return the measured data in the module's configured data format. The value returned may be a rough estimate of the real value.**Syntax** \$AA8(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module that you want to measure, its current/voltage loop and return the measured value.

8 is the Current Readback command.

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

Response !AA(data)(cr) if the command was valid.

?AA(cr) if an invalid command was issued.

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

! delimiter character indicating a valid command was received.

? delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module.

(data) is the value returned by the analog output module.

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

Example command: \$0A8(cr)

response: !0A18.773(cr)

The command tells the analog output module at address 0Ah to measure its current loop and return the measured value.

The analog output module returns the value 18.773 mA.

\$AA5

| | |
|--------------------|--|
| Name | Reset Status command |
| Description | Checks the Reset Status of the addressed analog output module to see whether it has been reset since the last Reset Status command was issued to the module. |
| Syntax | <p>\$AA5(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module who's Reset Status is to be returned.</p> <p>5 is the Reset Status command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AAS(cr) if the command was valid.</p> <p>?AA(cr) if an invalid command was issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exists.</p> <p>! delimiter character indicating a valid command was received.</p> <p>? delimiter character indicating the command was valid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module.</p> <p>S represents the Status bit that is returned by the analog output module. If S=1, the module has been reset since the last time it was issued a Reset Status command. If S=0, the module has not been reset since the last Reset Status command was issued.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$395(cr)</p> <p>response: !391(cr)</p> <p>The analog output module at address 39h was reset or powered up after the last execution of the Reset Status command. When executed once more the analog output module will reply with:</p> <p>!390(cr)</p> |

\$AAF

Name Read Firmware Version

Description The command requests the analog output module at address AA to return the version code of its firmware

Syntax \$AAF (cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module that you want to interrogate.

F identifies the version command.

(cr) is the terminating character, carriage return (ODh)

Response !AA(Version)(cr) if the command is valid.

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

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

AA (range 00-FF) represents the 2-character hexadecimal address of an analog output module.

(Version) is the version code of the module's firmware at address AA.

(cr) is the terminating character, carriage return (ODh).

\$AAM

| | |
|--------------------|---|
| Name | Read Module Name |
| Description | The command requests the analog output module at address AA to return its name |
| Syntax | <p>\$AAM (cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module that you want to interrogate.</p> <p>M is the Read Module Name command.</p> <p>(cr) is the terminating character, carriage return (ODh)</p> |
| Response | <p>!AA(Module Name)(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an analog output module.</p> <p>(Module Name) is the name of the module at address AA.</p> <p>For example: 4021</p> <p>(cr) is the terminating character, carriage return (ODh).</p> |

Spec :

1. 4 CH 12-Bits Analog Output With 4 Isolation General Purpose / Emergency Control Digital Input
2. Flexible Output Range : +/- 10 V, 0 ~ 20 mA, 4 ~ 20 mA
3. User Defined Start Up Output / Emergency Stop Output
4. Voltage : +/- 0.1% Of FSR
5. Current : +/- 0.1% Of FSR
6. 4 Isolation Digital Input / Emergency Input

ADAM4024 Command Review :

| About System / CH Setting | | | | | |
|---|---|-----------|-------------|------|-------|
| Command Set | Function | Response | Example | Note | T0 |
| %AANNTTCCFF | Module Configuration : Set Address, BaudRate, CheckSum AA : Current Module Address NN : New Module Address TT : Not Used For ADAM4024 And Must Be 00 CC : BaudRate Index 03: 1200 bps 04 2400 bps 05: 4800 bps 06: 9600 bps 07: 19200 bps 08: 38400 bps 09: 57600 bps 0A: 115200 bps FF : CheckSum Enable Flag Bit 7: Reserved Bit 6: CS: 0-Disable 1-Enable Bit 5 – 2 : Slew Rate Bit 1 – 0 : Reserved Slew Rate : Voltage (V/sec) Current (mA/sec) 00 : immediately immediately 01 : 0.0625 0.125 02 : 0.125 0.25 03 : 0.25 0.5 04 : 0.5 1.0 05 : 1.0 2.0 06 : 2.0 4.0 07 : 4.0 8.0 08 : 8.0 16.0 09 : 16.0 32.0 0A : 32.0 64.0 0B : 64.0 128.0 | !NN (?AA) | %0203000600 | !03 | 220us |
| About Output Data Command (f CMD Data Is Out Of Range Then The Response Will Indicate 0 Ex.!020) n = 0,1,2,3 Data : Engineer Unit -10V ~ +10V -10.000 ~ +10.000 #AACn+yy.yyy / #AACn-yy.yyy 0 ~ 20 mA +00.000 ~ +20.000 #AASCn+yy.yyy / #AAECn-yy.yyy (4 ~ 20 mA) +04.000 ~ +20.000 #AAECn+yy.yyy / #AA ECn-yy.yyy | | | | | |

| About System / CH Setting | | | | | |
|---------------------------|---|------------------|---|--------------------|-------------------|
| Command Set | Function | Response | Example | Note | T0 |
| #AACn(data) | Direct Output CHn Data | !AACn (data) | #02C2+07.- 456 #02C1-03.- 454 #02C0+11.- 234 | !02 !02 !020 | 1.5ms 33 ms |
| #AASCn(data) | Set data As CHn StartUp Data | !AASCn (data) | #02SC2 +07.456 #02SC1 -03.454 #02SC0 +11.234 | !02 !02 !020 | 1.5ms |
| #AAECn(data) | Set data As CHn Emergency Stop Data | !AAECn (data) | #02EC2 +07.456 #02EC1 -03.454 #02EC0 +11.234 | !02 !02 !020 | 1.5ms |
| #** | Synchronous Sample IDI | No Response | #** | | |
| \$AA0Cn | Set Current Trim Data As CHn 4mA Calibration Parameter/EEPROM | !AA | \$020C2 | !02 | 1ms |
| \$AA1Cn | Set Current Trim Data As CHn 20mA Calibration Parameter/EEPROM | !AA | \$021C2 | !02 | 1ms |
| \$AA2 | Read Back Module Status | !AATCCFF | \$022 | !0200CCFF | 1ms |
| \$AA3Cn(m) | Set Trim Data For CHn m : 0 ~ 127 And If Negative Then Add 0x80 (m=xx) Ex. m=89 → -9 m=09 → +9 | !AA | \$02308 \$02388 | !02 !02 | 1ms |
| \$AA4 | Read Back The IDI By Synchronous CMD(#**) Read IDI Input | !AAx | \$024 | !02C (IDI=C) | 1ms |
| \$AA5 | Read Back Reset Times After Last \$AA5 CMD And Clear This counter | !AAxx | \$025 | !0205 | 1ms |
| \$AA6Cn | Read Back CHn Last Output Value | !AA(data) | \$026C2 | !02+07.456 | 4ms |
| \$AA7CnRxx | Set CHn Output Type AA : Current Module Address Cn : 'C' Command Decode String, n=0,1,2,3 For 4 CH Rxx : 'R' Command Decode String, n=0,1,2 For 3 Output Type xx = 32 -10 V ~ +10 V xx = 30 0 ~ 20 mA xx = 31 4 ~ 20 mA | !AA (?AA) | \$027C2R32 | !02 | 1ms |

| About System / CH Setting | | | | | |
|---------------------------|---|------------------|----------------------|------------------|-------|
| Command Set | Function | Response | Example | Note | T0 |
| \$AA8Cn | Read Back CHn Output Range | !AACnxx | \$027C2 | !02C232 | 1ms |
| \$AAACnZ | Enable/Disable (Z=1/0) CHn EMS Flag | !AA | \$02AC21 \$02AC20 | !02 | 1ms |
| \$AABCn | Read CHn EMS Flag | !AACn1 !AACn0 | \$02BC2 | !02C21 !02C20 | 1ms |
| \$AADCn | Read Back CHn StartUp Data | !AA(data) | \$02DC2 | !02+07.456 | 4ms |
| \$AAECn | Read Back CHn Emergency Stop Data | !AA(data) | \$02EC2 | !02+07.456 | 4ms |
| \$AAF | Read Firmware Version | !AAAx.xx | \$02F | !02A0.00 | 300us |
| \$AAG | Reset Current Trim Data Variable To 0 | !AA | \$02G | !02 | 300us |
| \$AAH | Read Current Trim Data Variable | !AAxx (xx=m) | \$02H | !0208 | 300us |
| \$AAI | Read IDI | !AAx | \$02I | !02F | 300us |
| \$AAM | Read Module Name | !AA4024 | \$02M | !024024 | 300us |
| \$AANCn | Read Back CHn 4mA Calibration Parameter | !AAxx | \$02NC2 | !0206 | 900us |
| \$AAOCn | Read Back CHn 20mA Calibration Parameter | !AAxx | \$02OC2 | !020D | 900us |
| \$AAPCn | Clear CHn 4mA Calibration Parameter / EEPROM | !AA | \$02PC2 | !02 | 300us |
| \$AAQCn | Clear CHn 20mA Calibration Parameter / EEPROM | !AA | \$02QC2 | !02 | 300us |

Isolation Digital Input Definition :

Active : No Matter Active High Or Low The Firmware Must Read Set Status (On , '1')

Command : \$AA5

This Command Return The Module Reset Times From Last \$AA5 Executed, And Then The Reset Times Memory Clear To 0 EX. The Following Procedure Never Reset Module

\$025 !0203

\$025 !0200

\$025 !0200

EX. The Following Procedure

\$025 !0203

\$025 !0200

Now Reset Module

\$025 !0201

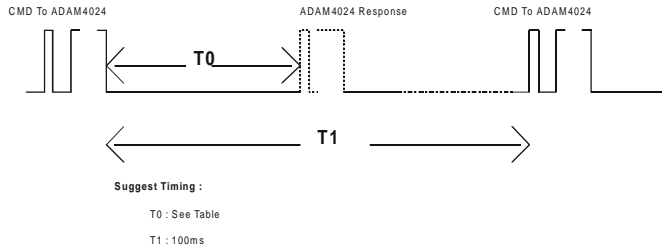
\$025 !0200

Now Reset Module 12 Times Never Use \$025 Command

\$025 !020C

\$025 !0200

If Input DAC Data Is Overflow Range Then The Return CMD Will Be :
!AAO



TP1 : -2.5 V Test Point

TP2 : +2.5 V Test Point

VR1 : Trim Reference Voltage -2.5 V That Is If -2.5 V Is Not
Equal To -2.499 ~ -2.501 Then Trim VR1

JP3 : Watch Dog Enable Select

If Short Then Enable WDT

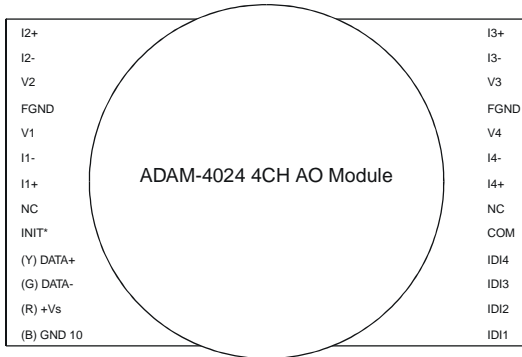
If Open Then Disable WDT

Trim Current Procedure : Cause 4 Separate DAC So All 4 CH
Current Must Trim Separate (Ex. Address=02 CH=2)

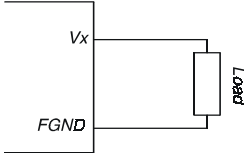
1. Set CHx Output Range To 0 ~ 20 mA Type
(\$027C2R30)
2. Reset Current Trim Data Variable To 0 (\$02G)
3. Output 4 mA Current To Meter (#02C2+04.000)
4. If Meter Is Not Equal 4 mA Then
 - (1) If Higher Than 4 mA Then Trim Data Down
(\$023C281)
 - (2) If Lower Than 4 mA Then Trim Data Up
(\$023C201)

- (3) If Equal 4 mA Then Save Parameter
(\$020C2) And Goto Next Step Otherwise (1) Or (2)
- 5. Reset Current Trim Data Variable To 0 (\$02G)
- 6. Output 20 mA Current To Meter (#02C2+20.000)
- 7. If Meter Is Not Equal 20 mA Then
 - (1) If Higher Than 20 mA Then Trim Data Down
(\$023C281)
 - (2) If Lower Than 20 mA Then Trim Data Up
(\$023C201)
 - (3) If Equal 20 mA Then Save Parameter
(\$021C2) And Goto Next Step Otherwise (1) Or (2)
- 8. Double Check 4 mA And 20 mA Is Right Or Not, If Wrong
Then Re-Calibration, If Right Then Next Step
- 9. Calibration OK !

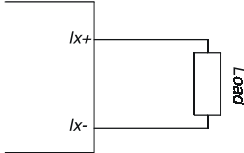
ADAM4024 Pin Define And Wiring Diagram:



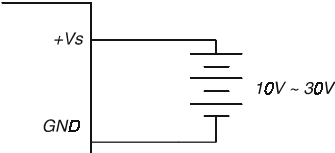
Voltage :



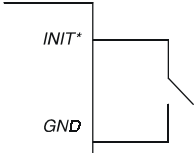
Current :



Power Supply :

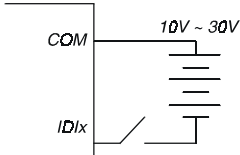


Initial :

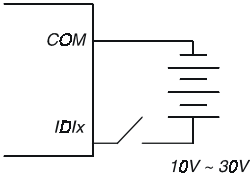


Isolation Input :

Type I : Input Active Low



Type II : Input Active High



4.6 Digital I/O and Relay Output Module Command

| Command Syntax | Command Name | Description | I/O Module |
|----------------|------------------------|---|--|
| %AANNTCCFF | Configuration | Sets the address, input range, baud rate, and/or checksum status, to a digital I/O module | 4050, 4052, 4053, 4060, 4051, 4055, 4068 |
| \$AA6 | Digital Data In | Returns the values of digital I/O channels of the addressed module | 4050, 4052, 4053, 4060, 4051, 4055, 4068 |
| #AAB(data) | Digital Data Out | Writes specified values to either a single channel or all channels simultaneously | 4050, 4060, 4055, 4068 |
| *** | Synchronized Sampling | Orders all digital I/O modules to sample their input values and store them in a special register | 4050, 4052, 4053, 4060, 4051, 4055, 4068 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified digital I/O module that was stored after an *** command | 4050, 4052, 4053, 4060, 4051, 4055, 4068 |
| \$AA2 | Configuration Status | Returns the configuration parameters for the specified digital I/O module | 4050, 4052, 4053, 4060, 4051, 4055, 4068 |
| \$AA5 | Reset Status | Indicates whether a specified digital I/O module was reset after the last time the \$AA5 command was issued | 4050, 4052, 4053, 4060, 4051, 4055, 4068 |
| \$AAF | Read Firmware Version | Return firmware version code from the specified digital I/O module | 4050, 4052, 4053, 4060, 4051, 4055, 4068 |
| \$AAM | Read Module Name | Return the module name from the specified digital I/O module | 4050, 4052, 4053, 4060, 4051, 4055, 4068 |

%AANNTTCFF

Name Configuration

Description Configure address, baud rate and/or checksum status of the addressed digital I/O module.

Syntax %AANNTTCFF(cr)

% is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module to be configured.

NN represents the new hexadecimal address of the digital I/O module. Range is from 00h to FFh.

TT represents the type code which is always set to 40 for a digital I/O module. (ADAM-4050, 4052, 4053, 4060, 4068)

CC represents the baud rate code. (See next page, Table 4-5)

FF is a hexadecimal number that equals the 8-bit parameter that represents the checksum status. (See Figure 4-3). Bits 0 through 5 and bit 7 are not used and set to 0.

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

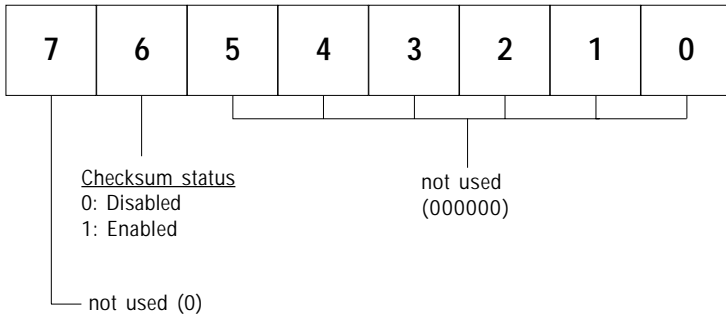


Figure 4-3 Checksum Parameter

%AANNTTCFF

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 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 a digital I/O module.
 (cr) is the terminating character, carriage return (0Dh)

Example command: %2324400600(cr)
 response: !24(cr)
 The command tries to configure module with address 23h to address 24h, baud rate 9600 no checksum checking. The response indicates that the configuration was successful.

NOTICE: *All configuration parameters can be changed dynamically,*

Table 4-5
Baudrate Codes

| Baud Rate Code (Hex) | Baud Rate |
|----------------------|-----------|
| 03 | 1200 bps |
| 04 | 2400 bps |
| 05 | 4800 bps |
| 06 | 9600 bps |
| 07 | 19.2 Kbps |
| 08 | 38.4 Kbps |

except checksum and baud rate parameters. They can only be altered when the INIT terminal is grounded. (Refer to Baud rate and Checksum, in Chapter 2 for the correct procedure.)*

\$AA6

| | |
|--------------------|--|
| Name | Digital Data In |
| Description | This command requests that the specified (AA) module returns the status of its digital input channels and returns a readback value of its digital output channels. |
| Syntax | <p>\$AA6(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module.</p> <p>6 is the Digital Data In command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!(dataOutput)(dataInput)00(cr) if the command was valid. (ADAM-4050)</p> <p>!(dataInput)0000(cr) if the command was valid. (ADAM-4052)</p> <p>!(dataInput)(dataInput) 00 (cr) if the command was valid. (ADAM-4053)</p> <p>!(dataOutput)0000(cr) if the command was valid. (ADAM-4060/4068)</p> <p>?AA(cr) if an invalid command has been issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicating a valid command was received.</p> <p>? delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module that is responding.</p> <p>(dataOutput) two-character hexadecimal value which either is the readback of a digital output channel or a relay.</p> <p>(dataInput) two-character hexadecimal value representing the input values of the digital I/O module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |

\$AA6

Example command: \$336(cr)
 response: !112200(cr)

The first two characters of the response, value 11h (00010001), indicate that digital output channels 0 and 4 are ON, channels 1, 2, 3, 5, 6, 7 are OFF. The second two characters of the response, value 22h (00100010), indicate that digital input channels 1 and 5 are HIGH, channels 0, 2, 3, 4, 6, 7 are LOW.

Example command: \$036(cr)
 response: !BEDE00(cr)

The first two characters of the response, value BEh(10111110), indicate that digital input channels 8 and 14 are LOW, channels 9, 10, 11, 12, 13 and 15 are HIGH. The second two characters of the response value DEh (11011110) indicate that digital input channels 0 and 5 are LOW, channels 1, 2, 3, 4, 6, 7 are HIGH.

#AABB

Name Digital Data Out

Description The command either sets a single digital output channel or sets all digital output channels simultaneously.

Syntax #AABB(data)(cr)

is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module you want to set its output value.

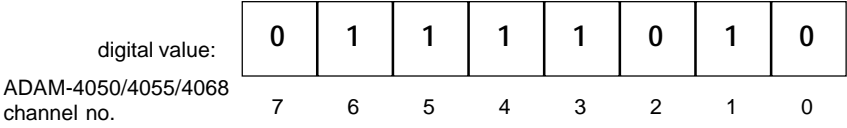
BB is used to indicate whether all channels will be set or a single channel will be set. In the last case BB also indicates which channel. Writing to all channels (write a byte): both characters should be equal to zero (BB=00). Writing to a single channel (write a bit): First character is 1, second character indicates channel number which can range from 0 to 7.

(data) is the hexadecimal representation of the digital output value(s).

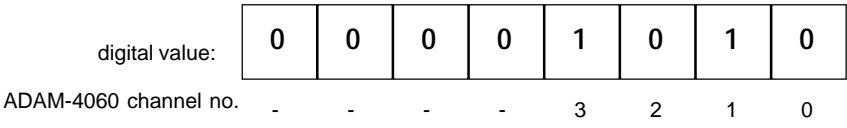
When writing to a single channel (bit) the first character is always 0. The value of the second character is either 0 or 1.

When writing to all channels (byte), both characters are significant (range 00h-FFh). The digital equivalent of these two hexadecimal characters represent the channels values.

The amount of channels on the ADAM-4050, ADAM-4055, ADAM-4060 and ADAM-4068 differs. The value 7A would mean the following for the 8 channels on the ADAM-4050, ADAM-4055 and ADAM-4068:



Since the ADAM-4060 has only four output channels all the meaning full values lie between 00h and 0Fh. The value 0Ah would mean the following for the ADAM-4060:



#AABB**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 valid command was received.

? delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module that is responding.

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

Examples

command: #140005(cr)

response: >(cr)

An output byte with value 05h (00000101) is sent to the digital I/O module at address 14h (Either ADAM-4050 or ADAM-4060). Its channels 0 and 2 will be set to ON. Other channels are set to OFF.

command: #151201(cr)

response: >(cr)

An output bit with value 1 is sent to channel 2 of a digital I/O module at address 15h (Either ADAM-4050 or ADAM-4060).

Channel two of the digital I/O module is set to ON.

*********Name** Synchronized Sampling**Description** Orders all (analog or digital) input modules to sample their input values and store them in a special register.**Syntax** ***

is a delimiter character.

** is the actual Synchronized Sampling command.

The terminating character, in the form of a carriage return (0Dh), is not required.

Response The digital I/O modules will not respond to the Synchronized Sampling command. In order to retrieve the data, you must execute a Read Synchronized Data command for every module separately.

\$AA4

| | |
|--------------------|--|
| Name | Read Synchronized Data |
| Description | The addressed digital I/O module is instructed to return the value that was stored in its register by a Synchronized Sampling command. |
| Syntax | <p>\$AA4(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module who's data is to be returned.</p> <p>4 is the Read Synchronized Data command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!(status)(dataOutput)(dataInput)00(cr) if the command was valid. (ADAM-4050)</p> <p>!(status)(dataInput)0000(cr) if the command was valid. (ADAM-4052)</p> <p>!(status)(dataInput)(dataInput)00(cr) if the command was valid. (ADAM-4053)</p> <p>!(status)(dataOutput)0000(cr) if the command was valid. (ADAM-4060/4068)</p> <p>?AA(cr) if an invalid command has been issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>!delimiter character indicating a valid command was received</p> <p>?delimiter character indicating the command was invalid</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module that is responding.</p> <p>(status) will tell you if the data (data) from the last Synchronized Sampling command (**#) has already been sent. If (status=1), then data has been sent for the first time after an Synchronized Sampling command was issued. If (status=0), then the data has been sent at least once after a Synchronized Sampling command was issued.</p> |

\$AA4

(dataOutput) two-character hexadecimal value which either is the readback of a digital output channel or a relay.

(dataInput) two-character hexadecimal value representing the input values of the digital I/O module.

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

Example

command: \$064(cr)

response: !1055100(cr)

The command asks the ADAM-4050 digital I/O module at address 06h to send its digital input data that was gathered during the last Synchronized Sampling command. The module returns 1055100, meaning status = 1 : data has not been sent before; dataOutput = 05h (00000101); digital output channels 0 and 2 are ON and channels 1, 3, 4, 5, 6, 7 are OFF; dataInput = 51h (01010001); digital input channels 0, 4, 6 are HIGH and channels 1, 2, 3, 4, 5 are LOW.

The digital I/O module responds with data = 055100 and status = 1, which means that this is the first time that the data has been sent.

command: \$064(cr)

response: !0055100(cr)

The command asks the digital I/O module at address 06h to send its digital input data.

The digital I/O module responds with data = 055100 and status = 0, which means that it has sent the same data at least once before. This may indicate that a previous Synchronized Sampling command was not received!

\$AA2

| | |
|--------------------|---|
| Name | Configuration Status command |
| Description | Returns the configuration parameters of the addressed digital I/O module. |
| Syntax | <p>\$AA2(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module to be interrogated.</p> <p>2 is Configuration Status command.</p> <p>(cr) is the terminating character, carriage return (ODh).</p> <p>This command requests the return of the configuration data from the digital I/O module at address AA.</p> |
| Response | <p>!AATTCCFF(cr) if the command is valid.</p> <p>?AA(cr) if an invalid command has been issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>!delimiter character indicating a valid command was received.</p> <p>?delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a digital I/O module.</p> <p>TT represents the type code, which is always 40.</p> <p>CC represents the baud rate code. (See next page, Table 4-6).</p> <p>FF is a hexadecimal number that equals the 8-bit parameter that represents checksum status and module identification. Bits 3 through 5 and bit 7 are not used and are being set to 0. (See figure 4-4 on the next page.)</p> <p>(cr) is the terminating character, carriage return (ODh)</p> |

\$AA2

Example command: \$452 (cr)
 response: !45400600 (cr)

The command asks the digital I/O module at address 45h to send its configuration data.

The digital I/O module at address 45h responds with baud rate 9600, no checksum function and the module is identifies as ADAM-4050

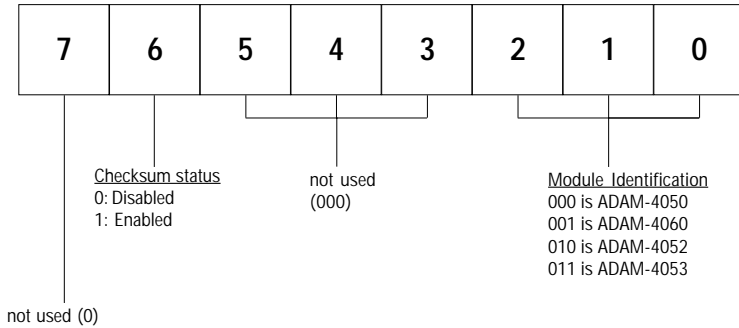


Figure 4-4 *Checksum & Identification Parameter*

Table 4-6
Baudrate Codes

| Baud Rate Code (Hex) | Baud Rate |
|----------------------|-----------|
| 03 | 1200 bps |
| 04 | 2400 bps |
| 05 | 4800 bps |
| 06 | 9600 bps |
| 07 | 19.2 Kbps |
| 08 | 38.4 Kbps |

\$AA5

| | |
|--------------------|--|
| Name | Reset Status command |
| Description | Requests the Reset Status of the addressed digital I/O module to see whether it has been reset since the last Reset Status command. |
| Syntax | <p>\$AA5(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog output module whose Reset Status is to be returned.</p> <p>5 is the Reset Status command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AAS(cr) if the command was valid.</p> <p>?AA(cr) if an invalid command has been issued.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>!delimiter character indicating a valid command was received.</p> <p>?delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module that is calibrated.</p> <p>S represents the Status bit that is returned by the digital I/O module. If S=1, the module has been reset since the last time it was issued. If S=0, the module has not been reset since the last Reset Status command was issued.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |

\$AA5

Example command: \$395(cr)
 response: !390(cr)

The command tells the digital I/O module at address 39h to return its Reset Status.

The digital I/O module at address 39h returns the value S=0, which indicates that the digital I/o module has not been reset or powered on since it was last issued a Reset Status command.

\$AAF

| | |
|--------------------|--|
| Name | Read Firmware Version |
| Description | The command requests the digital I/O module at address AA to return the version code of its firmware |
| Syntax | <p>\$AAF (cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module that you want to interrogate.</p> <p>F identifies the version command.</p> <p>(cr) is the terminating character, carriage return (ODh)</p> |
| Response | <p>!AA(Version)(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a digital I/O module.</p> <p>(Version) is the version code of the module's firmware at address AA.</p> <p>(cr) is the terminating character, carriage return (ODh).</p> |

\$AAM

| | |
|--------------------|--|
| Name | Read Module Name |
| Description | The command requests the digital I/O module at address AA to return its name |
| Syntax | <p>\$AAM (cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the digital I/O module that you want to interrogate.</p> <p>M is the Read Module Name command.</p> <p>(cr) is the terminating character, carriage return (ODh)</p> |
| Response | <p>!AA(Module Name)(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a digital I/O module.</p> <p>(Module Name) is the name of the module at address AA.</p> <p>For example: 4052</p> <p>(cr) is the terminating character, carriage return (ODh).</p> |

4.7 Counter/Frequency Module Command

4.7.1 Configuration, Counter Input and Display Command Set

| Command Syntax | Command Name | Description | I/O Module |
|----------------|------------------------|---|------------------------------------|
| %AANNTTCCFF | Configuration | Sets the address, input range, baud rate, and/or checksum status, to a digital I/O module | 4050, 4052, 4053, 4060, 4051, 4055 |
| \$AA6 | Digital Data In | Returns the values of digital I/O channels of the addressed module | 4050, 4052, 4053, 4060, 4051, 4055 |
| #AAB(data) | Digital Data Out | Writes specified values to either a single channel or all channels simultaneously | 4050, 4060, 4055 |
| *** | Synchronized Sampling | Orders all digital I/O modules to sample their input values and store them in a special register | 4050, 4052, 4053, 4060, 4051, 4055 |
| \$AA4 | Read Synchronized Data | Returns the value that was stored in the specified digital I/O module that was stored after an *** command | 4050, 4052, 4053, 4060, 4051, 4055 |
| \$AA2 | Configuration Status | Returns the configuration parameters for the specified digital I/O module | 4050, 4052, 4053, 4060, 4051, 4055 |
| \$AA5 | Reset Status | Indicates whether a specified digital I/O module was reset after the last time the \$AA5 command was issued | 4050, 4052, 4053, 4060, 4051, 4055 |
| \$AAF | Read Firmware Version | Return firmware version code from the specified digital I/O module | 4050, 4052, 4053, 4060, 4051, 4055 |
| \$AAM | Read Module Name | Return the module name from the specified digital I/O module | 4050, 4052, 4053, 4060, 4051, 4055 |

%AANNTTCCFF

Name: Configuration

Description: Sets the address, input mode, baud rate, checksum status and/or frequency gate time for a specified counter/frequency module

Syntax: %AANNTTCCFF (cr)

% is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter module that you want to interrogate.

NN represents the new hexadecimal address of the counter/frequency module. Range is from 00h to FFh.

TT represents the input mode:

50h : counter input mode

51h : frequency measurement mode

CC represents the baud rate code (See table 4-7).

FF is a hexadecimal number that equals the 8-bit parameter representing the checksum status and frequency gate time. The layout of the 8-bit parameter is shown in figure 4-5.

Bits 0, 1, 3, 4, 5 and 7 are not used and set to 0.

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

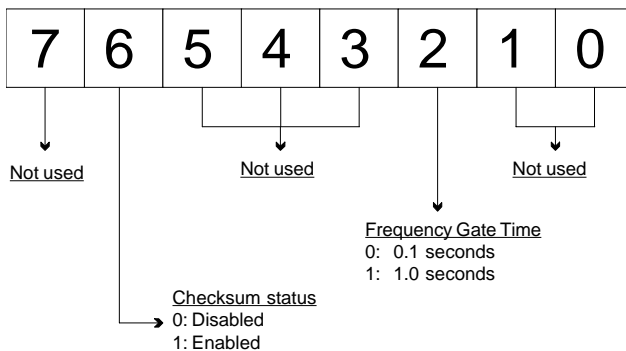


Figure 4-5 Data format for 8-bit parameter

%AANNTTCFF

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 setting.
 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.
 ? is a delimiter character indicating the command was invalid.
 AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency module.
 (cr) is the terminating character, carriage return (0Dh).

Example

command: %0120510600(cr)
 response: !20(cr)

The ADAM 4080D module at address 01 is configured to new address 20, frequency measurement module, baud rate 9600, frequency gate time 0.1 second and no checksum checking or generation.

The response indicates that the command was received. Wait seven seconds to let the new configuration settings take effect before you issue a new command to the module.

| Baud Rate Code (Hex) | Baud Rate |
|----------------------|-----------|
| 03 | 1200 bps |
| 04 | 2400 bps |
| 05 | 4800 bps |
| 06 | 9600 bps |
| 07 | 19.2 Kbps |
| 08 | 38.4 Kbps |

Table 4-7:
Baudrate Codes

NOTICE: *You can change all configuration parameters dynamically except checksum and baud rate parameters. They can only be altered when the INIT* terminal is grounded.*

\$AA2

| | |
|--------------------|---|
| Name | Configuration Status |
| Description | The command requests the return of the configuration data from the counter/frequency module at address AA. |
| Syntax | <p>\$AA2(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module that you want to interrogate.</p> <p>2 is the Configuration Status command.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AATTCCFF (cr) if the command is valid.</p> <p>?AA(cr)if an invalid operation was entered</p> <p>There is no response if the module detects a syntax error or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command was received</p> <p>? is a delimiter character indicating the command was invalid</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.</p> <p>TT (50h or 51h) represents the type code. 50h means that using module as counter. 51h means using module as frequency measurement module.</p> <p>CC represents the baud rate code.</p> <p>FF is a hexadecimal number that equals the 8-bit parameter representing the checksum status and frequency gate time. The layout of the 8-bit parameter is shown in figure 4-5. bits not used are set to 0.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> <p>(Also see the %AANNTTCCFF configuration command)</p> |

\$AAF

| | |
|--------------------|--|
| Name | Read Version |
| Description | The command requests the analog input module at address AA to return the version code of its firmware |
| Syntax | <p>\$AAF(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the counter/frequency module that you want to interrogate.</p> <p>F identifies the version command.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(Version)(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command was received.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module.</p> <p>(version) is the version code of the module's firmware at address AA.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |

\$AAM

Name Read Module Name

Description The command requests the counter/frequency module at address AA to return its name

Syntax \$AAM (cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

M identifies the Read Module Name command.

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

Response

!AA(Module Name)(cr) if the command is valid.

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

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

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module.

(Module Name) is the name of the module at address AA. For example: 4080D

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

\$AABS

| | |
|--------------------|--|
| Name | Set Input Mode |
| Description | Set the input signal mode of the specified counter/frequency module to either non-isolated (TTL) or photo-isolated. |
| Syntax | <p>\$AABS(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency that you want to interrogate.</p> <p>B identifies the Set Input Signal Mode command.</p> <p>S indicates the input signal mode. S = 0 configures the module for non-isolated (TTL) input and S = 1 configures the module for photo-isolated input .</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exists.</p> <p>! is a delimiter character indicating the command was valid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$03B0(cr)</p> <p>response: !03(cr)</p> <p>The command configures the counter/frequency modules at address 03 to receive non-isolated input. The addressed module replies by sending its address to indicate that it executed the command.</p> |

NOTICE: *The input mode command is not related to a specific channel. When the input mode is set both channels are changed accordingly.*

\$AAB

Name Read Input Mode.

Description Read the input mode of the specified counter/frequency module.

Syntax \$AAB(cr)
\$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency that you want to interrogate.
B identifies the Read Input Signal Mode command.
(cr) is the terminating character, carriage return (0Dh)

Response !AAS(cr) if the command is valid.
There is no response if the module detects a syntax error or communication error, or if the specified address does not exist.
! delimiter character indicating the command was valid.
AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency module.
S indicates the input signal mode. S = 0 configures the module for non-isolated (TTL) input and S = 1 configures the module for photo-isolated input.
(cr) is the terminating character, carriage return (0Dh).

Example command: \$03B(cr)
response: !030(cr)

The command requests the counter/frequency module at address 03 to return its input mode. The addressed module replies that its input mode is set to receive non-isolated input.

#AAN

Name: Read Counter or Frequency Value

Description: Instructs the addressed counter/frequency module at address AA to read the counter or frequency value of counter 0 or counter 1 and return the acquired data.

Syntax: #AAN(cr)
is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.
N represents the counter number.
N=0 represents counter 0
N=1 represents counter 1
(cr) is the terminating character, carriage return (0Dh)

Response: >data(cr) if the command is valid.
?AA(cr) if an invalid operation was entered.
There is no response if the module detects a syntax error or communication error or if the specified address does not exist.
? is a delimiter character indicating the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency input module.
(data) is the value that is retrieved by the module by reading counter 0 or counter 1. The data format consists of eight hexadecimal digits.
(cr) is the terminating character, carriage return (0Dh).

Example: command: #120(cr)
response: >000002FE(cr)

The command requests the counter/frequency module at address 12 to read the counter 0 and return the data. The counter/frequency module at address 12 responds that the value of counter 0 is 000002FE (hexadecimal) which equals 766 (decimal).

\$AA8V

Name Select LED Data Origin

Description Select whether LED will display data from the counter/frequency module directly or from the host computer.

Syntax \$AA8V(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

8 identifies the Select LED Data Origin command.

V indicates the origin of the data to be displayed on the LED:

V=0 sets the LED data origin to the module's counter 0

V=1 set the LED data origin to the module's counter 1

V=2 set the LED data origin to the host computer

(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.

! is a delimiter character indicating the command was valid.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency input module.

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

Example command: \$0182(cr)

response: !01(cr)

The command sets the counter/frequency modules at address 01 to display data sent by the host computer. After this command has been issued the host computer can use command \$AA9(data) to send the data to the addressed module.

\$AA8

| | |
|--------------------|--|
| Name | Read LED Data Origin |
| Description | Read the LED Data Origin status which determines whether LED will display data from the counter/frequency module directly or from the host computer |
| Syntax | <p>\$AA8(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.</p> <p>8 identifies the Read LED Data Origin command.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AAV(cr) if the command is valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax error or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating the command was valid.</p> <p>? is a delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency input module.</p> <p>V indicates the origin of the data to be displayed on the LED: V=0 sets the LED data origin to the module's counter 0 V=1 set the LED data origin to the module's counter 1 V=2 set the LED data origin to the host computer</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$018(cr)</p> <p>response: !011(cr)</p> <p>The command requests the counter/frequency module at address 01 to return its LED Data Origin status. The module replies that it currently displays data from counter 1.</p> |

\$AA9(data)

Name Send Data to LED

Description The host computer sends data to the addressed module to display on its LED.

Syntax \$AA9(data)(cr)
\$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.
9 identifies the Send LED Data command.
(data) is a floating point numeral consisting of five number digits and one digit for the decimal point. Its maximum value is 99999.

(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.
! is a delimiter character indicating the command was valid.
? is a delimiter character indicating the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency module.
(cr) is the terminating character, carriage return (0Dh).

Example command: \$0198999.9(cr)
response: !01(cr)

The command requests the host computer to send 8999.9 to the counter/frequency module at address 01 to display on its LED display. This command is only valid after the command \$0182 has been issued.

4.7.2 Counter Setup Command Set

| Command Syntax | Command Name | Description | I/O Module |
|----------------|--------------------------------|--|-------------|
| \$AAAG | Set Gate Mode | Requests the specified counter/ frequency module to set its gate mode to either high, low or disabled | 4080, 4080D |
| \$AAA | Read Gate Mode | Requests the specified counter/ frequency module to return the status of its gate mode | 4080, 4080D |
| \$AA3N(data) | Set Maximum Counter Value | Sets the maximum value of counter 0 or counter 1 for the specified counter/frequency module | 4080, 4080D |
| \$AA3N | Read Maximum Counter Value | Reads the maximum value of counter 0 or counter 1 of the specified counter/ frequency module | 4080, 4080D |
| \$AA5NS | Start/Stop Counter | The command orders the specified counter/frequency module to start or stop counting | 4080, 4080D |
| \$AA5N | Read Counter Start/Stop Status | The addressed counter frequency module returns its status indicating whether counting is enabled or disabled | 4080, 4080D |
| \$AA6N | Clear Counter | The command clears the counter 0 or counter 1 of the specified counter module | 4080, 4080D |
| \$AA7N | Read Overflow Flag | The addressed module returns the status of the overflow flag of counter 0 or counter 1 | 4080, 4080D |

\$AAAG

Name Set Gate Mode.

Description Request the specified counter/frequency module to set its gate to either high, low or disabled.

Syntax \$AAAG(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency that you want to interrogate.

A identifies the Gate Mode command.

G determines the gate mode.

G = 0 the gate is low

G = 1 the gate is high

G = 2 the gate is disabled.

(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 or communication error, or if the specified address does not exist.

! is a delimiter character indicating the command was valid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency module.

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

Example command: \$01A1(cr)

response: !01(cr)

The command requests the counter/frequency module at address 01 to set its gate high. The addressed module replies with its address to indicate that it has executed the command.

\$AAA

| | |
|--------------------|---|
| Name | Read Gate Mode. |
| Description | Request the specified counter/frequency module to return its gate status. |
| Syntax | <p>\$AAA(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency that you want to interrogate.</p> <p>A identifies the Read Gate Mode command.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AAG(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating the command was valid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency module.</p> <p>G determines the gate mode.</p> <p>G = 0 the gate is low</p> <p>G = 1 the gate is high</p> <p>G = 2 the gate is disabled.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$01A(cr)</p> <p>response: !011(cr)</p> <p>The command requests the addressed counter/frequency module to return its gate status. The module at address 01 replies that its gate is high.</p> |

\$AA3N(data)

Name Set Maximum Counter Value

Description Set the maximum value of counter 0 or counter 1 for a specified counter/frequency module.

Syntax \$AA3N(data)(cr)

\$ is a delimiter character.

AA(range 00-FF) represents the 2-character hexadecimal address of the counter/frequency module that you want to interrogate.

3 identifies the Set Maximum Counter Value command

N determines the counter for which the maximum counter value is to be set.

N = 0 represents counter 0

N = 1 represents counter 1

(data) is the maximum count value which consists of eight hexadecimal digits. The addressed module will accumulate the input counts until it reaches the maximum value. When counting exceeds the maximum counter value, the counter will stop counting. The programmer should use the command \$AA6N to reset the counter to 1.

(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.

! is a delimiter character indicating the command was valid.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter input module.

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

Example command: \$24300000ffff(cr)

response: !24(cr)

The command requests the counter/frequency module at address 24 to set the maximum counter value for counter 0 to 65535 (0x0000ffff). The module replies that it has executed the command.

\$AA3N

| | |
|--------------------|--|
| Name | Read Maximum Counter Value |
| Description | Read the maximum counter value of the counter 0 or counter 1 for a specified counter/frequency module. |
| Syntax | <p>\$AA3N(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.</p> <p>3 identifies the Read Maximum Counter Value command</p> <p>N determines the counter for which the maximum counter value is to be set.</p> <p>N = 0 represents counter 0</p> <p>N = 1 represents counter 1</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(data)(cr) if the command is valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating the command was valid.</p> <p>? is a delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter input module.</p> <p>(data) is the maximum counter value which consists of eight hexadecimal digitals.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$2430(cr)</p> <p>response: !240000ffff(cr)</p> <p>The command requests the counter/frequency module at address 24 to the maximum count number of counter 0. The addressed module replies that the maximum count number of channel 0 is 65535 (0000ffff)</p> |

\$AA5NS

Name Start/Stop Counter

Description Request the addressed counter/frequency module to start or stop the counting for a counter 0 or counter 1.

Syntax \$AA5NS(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

5 identifies the Start/Stop Counter command

N determines the counter that should be enabled or disabled.

N = 0 represents counter 0

N = 1 represents counter 1

S represents the counter status.

S = 0 stops counting

S = 1 starts counting

(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 or communication error, or if the specified address does not exist.

! is a delimiter character indicating the command was valid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter input module.

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

Example command: \$06501(cr)

response: !06(cr)

The command requests the counter/frequency module at address 06 to start counter 0. The addressed module replies with its address to indicate the command has been executed and counter 0 has started.

\$AA5N

| | |
|--------------------|--|
| Name | Read Counter Start/Stop Status |
| Description | Requests the addressed counter/frequency module to indicate whether counter 0 or counter 1 is active. |
| Syntax | <p>\$AA5N(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.</p> <p>N determines the counter for which the status should be returned.</p> <p>N = 0 represents counter 0</p> <p>N = 1 represents counter 1</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AAS(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating the command was valid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter input module.</p> <p>S represents the counter status.</p> <p>S = 0 indicates counting</p> <p>S = 1 indicates not counting</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$0650(cr)</p> <p>response: !061(cr)</p> <p>The command requests the counter/frequency module at address 06 to return the status of counter 0. The addressed module replies that counter 0 is counting</p> |

\$AA6N**Name** Clear Counter**Description** Clears the counter 0 or counter 1 of the specified counter/frequency module.**Syntax** \$AA6N(cr)
\$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.
6 the Clear Counter command.
N determines the counter which should be cleared.
N = 0 represents counter 0
N = 1 represents counter 1
(cr) is the terminating character, carriage return (0Dh)**Response** !AA(cr) if the command is valid.
There is no response if the module detects a syntax error or communication error or if the specified address does not exist.
! is a delimiter character indicating the command was valid.
AA (range 00-FF) represents the 2-character hexadecimal address of a counter input module.
(cr) is the terminating character, carriage return (0Dh).**Example** command: \$1361(cr)
response: !13(cr)

The command requests the counter/frequency module at address 13 to clear counter 1. The addressed module replies with its address to indicate the counter has been cleared.

\$AA7N

| | |
|--------------------|--|
| Name | Read/Clear Overflow Flag. |
| Description | The command requests the addressed module to return the status the overflow flag of counter 0 or counter 1 and clear the flag afterwards. |
| Syntax | <p>\$AA7N(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.</p> <p>7 identifies the Read/Clear Overflow Flag command</p> <p>N determines the channel which overflow flag status should be read and cleared. N = 0 represents counter 0 and N = 1 represents counter 1</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AAV(cr) if the command is valid.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating the command was valid.</p> <p>? is a delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter input module.</p> <p>V represents the status of the Nth channel's overflow flag.</p> <p>V = 1 means that the overflow flag has been set because the counting has exceeded the maximum count. V = 0 means that the overflow flag has not been set.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$1371(cr)</p> <p>response: !131(cr)</p> <p>The command requests the counter/frequency module at address 13 to return the status of the overflow flag of counter 1 and reset it. The addressed module replies that the overflow flag for counter 1 indicated overflow and was reset.</p> |

4.7.3 Digital Filter and Programmable Threshold Command Set

| Command Syntax | Command Name | Description | I/O Module |
|----------------|---|---|-------------|
| \$AA4S | Enable/Disable Digital Filter | Enables or disables the digital filter of the addressed counter/frequency module | 4080, 4080D |
| \$AA4 | Read Filter Status | The addressed counter frequency module returns the status of its digital filter | 4080, 4080D |
| \$AA0H(data) | Set Minimum Input Signal Width at High Level | Sets the minimum input signal width at high level for a specified counter/frequency module | 4080, 4080D |
| \$AA0H | Read Minimum Input Signal Width at High Level | Reads the minimum input signal width setting at high level for a specified counter/frequency module | 4080, 4080D |
| \$AA0L(data) | Set Minimum Input Signal Width at Low Level | Sets the minimum input signal width at low level for a specified counter/frequency module | 4080, 4080D |
| \$AA0L | Read Minimum Input Signal Width at Low Level | Reads minimum input signal width setting at low level for a specified counter/frequency module | 4080, 4080D |
| \$AA1H(data) | Set Non-isolated High Trigger Level | Sets the high trigger level of non-isolated input signals for a specified counter/frequency module | 4080, 4080D |
| \$AA1H | Read Non-isolated High Trigger Level | Requests the addressed counter frequency module to return the high trigger level for non-isolated input signals | 4080, 4080D |
| \$AA1L(data) | Set Non-isolated Low Trigger Level | Sets the low trigger level of non-isolated input signals for a specified counter/frequency module | 4080, 4080D |
| \$AA1L | Read Non-isolated Low Trigger Level | Requests the addressed counter/ frequency module to return the low trigger level for non-isolated input signals | 4080, 4080D |

\$AA4S

Name Enable/Disable Digital Filter

Description Enables or disables the digital filter of the addressed counter/frequency module

Syntax \$AA4S(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

4 identifies the Enable/Disable Filter command

S is the digital filter mode.

S = 0 means disable filter

S = 1 means enable filter

(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 or communication error, or if the specified address does not exist.

! is a delimiter character indicating the command was valid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter input module.

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

Example command: \$0340(cr)

response: !03(cr)

The command orders the counter/frequency module at address 03 to disable its digital filter. The addressed module returns its address to indicate that it has executed the command successfully.

\$AA4

| | |
|--------------------|---|
| Name | Read Filter Status |
| Description | Read the digital filter status of the addressed counter/frequency module |
| Syntax | <p>\$AA4(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.</p> <p>4 identifies the Read Filter Status command</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AAS(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! is a delimiter character indicating the command was valid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter input module.</p> <p>S is the digital filter mode.</p> <p>S = 0 means filter is disabled</p> <p>S = 1 means filter is enabled</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: \$034(cr)</p> <p>response: !030(cr)</p> <p>The command requests the counter/frequency module at address 03 to return whether its digital filter is on or not. The addressed module returns its address to indicate that its digital filter is disabled.</p> |

\$AA0H(data)

Name Set Minimum Input Signal Width at High Level

Description: Set the minimum input signal width at high level for a specified counter/frequency module to filter the noise.

Syntax: \$AA0H(data)(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

0H identifies the Set Minimum Input Signal Width at High Level command

(data) is the minimum width at high level. The unit is μsec (microseconds) and its resolution is 1 μsec . The format is a five digit integer that can range from 2 μsec to 65535 μsec . Out of range values will cause errors.

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

Response: !AA(cr) if the command is valid. The addressed module stores the value and will recognize the input signal "high" only after the input signal continues "high" and for longer than the specified value.

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

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

! is a delimiter character indicating the command was valid.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency input module.

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

Example: command: \$130H00020(cr)

response: !13(cr)

The command requests the counter/frequency module at address 13 to set the minimum input width at high level to 20 μsec . The addressed module stores the value and will recognize an input signal to be "high" only if the signal continues to be "high" longer than 20 μsec . This function can be used as a digital filter.

\$AA0H

Name: Read Minimum Input Signal Width at High Level.

Description: Read the minimum input signal width at high level for a specified counter/frequency module.

Syntax: \$AA0H(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

0H identifies the Read Minimum Input Signal Width at High Level 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 or communication error, or if the specified address does not exist.

! is a delimiter character indicating the command was valid.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency input module.

(data) is the minimum width at high level. The unit is μsec (microseconds) and its resolution is 1 μsec . The format is a five digit integer that ranges from 2 μsec to 65535 μsec .

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

Example: command: \$130H(cr)

response: !1300020(cr)

The command requests the counter/frequency module at address 13 to read its minimum input signal width at high level. The addressed module replies that its minimum input signal width at high level is 20 μsec .

\$AA0L(data)

| | |
|---------------------|---|
| Name | Set Minimum Input Signal Width at Low Level |
| Description: | Set the minimum input signal width at low level for a specified counter/frequency module to filter noise. |
| Syntax: | <p>\$AA0L(data)(cr)</p> <p>\$ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.</p> <p>0H identifies the Set Minimum Input Signal Width at Low Level command</p> <p>(data) is the minimum width at low level. The unit is μsec (microseconds) and its resolution is 1 μsec. The format is a five digit integer that can range from 2 μsec to 65535 μsec. Out of range values will cause errors.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response: | <p>!AA(cr) if the command is valid. The addressed module stores the value and will recognize the input signal "low" only if the input signal continues to be "low" longer than the specified time.</p> <p>?AA(cr) if an invalid operation was entered.</p> <p>There is no response if the module detects a syntax or communication error, or if the specified address does not exist.</p> <p>! is a delimiter character indicating the command was valid.</p> <p>? is a delimiter character indicating the command was invalid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency input module.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Example: | <p>command: \$050L00084(cr)</p> <p>response: !05(cr)</p> <p>The command requests the counter/frequency module at address 05 to set the minimum input width at low level to 84 μsec. The addressed module stores the value and will recognize an input signal to be "low" only if the signal continues to be "low" longer than 84 μsec. This function can be used as digital filter.</p> |

\$AAOL

Name: Read Minimum Input Signal Width at Low Level.

Description: Read the minimum input signal width at low level for a specified counter/frequency module to filter noise.

Syntax: \$AAOL(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of the counter/frequency module that you want to interrogate.

OH identifies the Read Minimum Input Signal Width at Low Level 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 or communication error, or if the specified address does not exist.

! is a delimiter character indicating the command was valid.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency input module.

(data) is the minimum width at low level. The unit is μsec (microsecond) and its resolution is $1 \mu\text{sec}$. The format is a five digit integer that ranges from $2 \mu\text{sec}$ to $65535 \mu\text{sec}$.

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

Example: command: \$050L(cr)

response: !0500084(cr)

The command requests the counter/frequency module at address 05 to read its minimum input signal width at low level.

The addressed module replies that its minimum input signal width at low level is $84 \mu\text{sec}$.

\$AA1H(data)

Name: Set Non-isolated High Trigger Level.

Description: Set the high trigger level for non-isolated input signals for a specified counter/frequency module.

Syntax \$AA1H(data)(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

1H identifies the Set Non-isolated High Trigger Level command

(data) is the high trigger level for non-isolated input

The unit and resolution are both 0.1 V (voltage). The format is a two digit integer that can range from 1 to 50 (i.e. 0.1 to 5 V). This high trigger level must at all times be higher than the low trigger level, set by the \$AA1L(data) command. When the high trigger level is out of range or lower than the low trigger level an error will be the result.

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

Response: !AA(cr) if the command is valid. The addressed module stores the value and will recognize input signal as “high” only when they exceed the high trigger level

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

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

! is a delimiter character indicating the command was valid.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency input module.

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

Example: command: \$131H30(cr)

response: !13(cr)

The command requests the counter/frequency module at address 13 to set its non-isolated high trigger level to 3 V. The addressed module stores the value and will recognize the input signals to be “high” only after the signals exceed 3 V. This function can be used as a level filter.

\$AA1H

Name: Read Non-isolated High Trigger Level.

Description: Read the high trigger level for non-isolated input signals of a specified counter/frequency module.

Syntax: \$AA1H(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

1H identifies the Read Non-isolated High Trigger Level

(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.

! is a delimiter character indicating the command was valid.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency module.

(data) is the high trigger level of non-isolated input signals

The unit and resolution are both 0.1 V (voltage). The format is a two digit integer that can range from 1 to 50 (i.e. 0.1 V to 5 V).

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

Example: command: \$131H(cr)

response: !1330(cr)

The command requests the counter/frequency module at address 13 to read its non-isolated high trigger level. The addressed module replies that the high trigger level is 3 V.

\$AA1L(data)

Name: Set Non-isolated Low Trigger Level.

Description: Set the low trigger level of non-isolated input signals for a specified counter/frequency module.

Syntax \$AA1L(data)(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

1L identifies the Set Non-isolated Low Trigger Level command

(data) is the low trigger level for non-isolated input signals. The unit and resolution are both 0.1 V (voltage). The format is a two digit integer that can range from 1 to 50 (i.e. 0.1 to 5 V). This low trigger level must at all times be lower than the high trigger level, set by \$AA1H(data) command. When the low trigger level is out of range or higher than the high trigger level an error will result.

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

Response: !AA(cr) if the command is valid. The addressed module stores the value and will recognize input signal as “low” only when it exceeds the low trigger level

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

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

! is a delimiter character indicating the command was valid.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency input module.

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

Example: command: \$051L08(cr)

response: !05(cr)

The command requests the counter/frequency module at address 05 to set its non-isolated low trigger level to 0.8 V. The addressed module stores the value and will recognize the TTL input signal to be “low” only if the signal exceeds 0.8 V. This function can be used as a level filter.

\$AA1L

Name: Read Non-isolated Low Trigger Level.

Description: Read the low trigger level for non-isolated input signals of a specified counter/frequency module.

Syntax: \$AA1L(cr)

\$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of counter/frequency module that you want to interrogate.

1L identifies the Read Non-isolated Low Trigger Level

(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 or communication error, or if the specified address does not exist.

! is a delimiter character indicating the command was valid.

? is a delimiter character indicating the command was invalid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter/frequency module.

(data) is the low trigger level of non-isolated input signal. The unit and resolution are both 0.1 V (voltage). The format is a two digit integer that can range from 1 to 50 (i.e. 0.1 V to 5 V).

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

Example: command: \$051L(cr)

response: !0508(cr)

The command requests the counter/frequency module at address 05 to read its non-isolated input signal low trigger level. The addressed module replies that its low trigger level is 0.8 V.

4.7.4 Digital Output and Alarm Command Set

| Command Syntax | Command Name | Description | I/O Module |
|----------------|---------------------------------------|---|-------------|
| @AAPN(data) | Set Initial Count Value of Counter N | Sets the initial count value of the module for counter 0 or counter 1 | 4080 |
| @AAGN | Read Initial Count Value of Counter N | Read the initial count value of counter 0 or counter 1 | 4080 |
| @AAEAN | Enable Alarm of Counter N | Enable alarm for the specified counter 0 or counter 1 | 4080 |
| @AADAN | Disable Alarm of Counter N | Disable alarm for the specified counter 0 or counter 1 | 4080 |
| @AAPA(data) | Set Alarm Limit Value of Counter 0 | Download the alarm limit value for counter 0 of the specified module | 4080 |
| @AASA(data) | Set Alarm Limit Value of Counter 1 | Download the alarm limit value for counter 1 of the specified module | 4080 |
| @AARP | Read Alarm Limit Value of Counter 0 | Ask the module to return the alarm limit value of counter 0 | 4080 |
| @AARA | Read Alarm Limit Value of Counter 1 | Ask the module to return the alarm limit value of counter 1 | 4080 |
| @AADO(data) | Set Digital Output Values | Set the values of the module's two digital outputs (ON or OFF) | 4080, 4080D |
| @AADI | Read Digital Output and Alarm Status | Ask the addressed module to return the state of its two digital output channels and the status of its alarm | 4080, 4080D |

Command Set

| Command Syntax | Command Name | Description | I/O Module |
|----------------|--|--|------------|
| @AAEAT | Enable Alarm of Counter 0 | Enable the alarm of counter 0 in either momentary or latching modes | 4080D |
| @AADA | Disable Alarm of Counter 0 | Disable all alarm functions of counter 0 | 4080D |
| @AACA | Clear Latch Alarm | Both alarm states of the counter are set to OFF, no alarm | 4080D |
| @AAPA(data) | Set Low-Alarm Count Value of Counter 0 | Downloads the low-alarm count value for counter 0 of the specified module | 4080D |
| @AASA(data) | Set High-Alarm Count Value of Counter 0 | Downloads the high-alarm count value for counter 0 of the specified module | 4080D |
| @AARP | Read Low-Alarm Count Value of Counter 0 | Ask the module to return the low-alarm count value for counter 0 | 4080D |
| @AARA | Read High-Alarm Count Value of Counter 0 | Ask the module to return the high-alarm count value for counter 0 | 4080D |

@AAPN(data)

Name Set Initial Count Value of Counter 0 (or 1)
Description Set the initial count value for counter 0 or counter 1 of the specified counter module at address AA.

Syntax @AAPN(data)(cr)
@ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.
PN represents the Set Initial Count Value command.
N = 0 represents counter 0
N = 1 represents counter 1
(data) is the initial count value which must consist of eight hexadecimal digits.
(cr) is the terminating character, carriage return (0Dh).

Response !AA(cr) if the command is valid.
There is no response if the module detects a syntax error or communication error or if the specified address does not exist.
! is a delimiter character indicating a command was valid.
AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.
(cr) is the terminating character, carriage return (0Dh).

Example command: @12P0000000FF(cr)
response: !12(cr)
The command sets the initial count value 000000FF for counter 0 of the counter module at address 12. The addressed module replies that the command has been received.

@AAGN

| | |
|--------------------|---|
| Name | Read Initial Count Value of Counter 0 (or 1) |
| Description | Read the initial count value of counter 0 or 1 of the specified counter module at address AA. |
| Syntax | <p>@AAGN(data)(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>GN represents the Read Initial Count Value for counter command.</p> <p>N=0 represents counter 0</p> <p>N=1 represents counter 1</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(data)(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! is a delimiter character indicating a command was valid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>(data) is the initial count value which must consist of eight hexadecimal digits.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @12G0(cr)</p> <p>response: !12000000FF(cr)</p> <p>The command instructs the module at address 12h to return the initial value of counter 0. The module replies that initial count value of counter 0 is 000000FF.</p> |

@AAEAN

Name Enable Alarm

Description Enable Alarm for the specified counter

Syntax @AAEAN(cr)

@ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.

EAN represents the Enable Alarm for counter command.

N=0 Represents counter 0

N=1 Represents counter 1

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

Response

!AA(cr) if the command is valid.

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

! is a delimiter character indicating a command was valid.

AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.

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

Example

command: @12EA0(cr)

response: !12(cr)

The command sets the Alarm Status as 'Enable' for counter 0 of the counter module at address 12h. It means that the digital output will be triggered while the count value of the counter 0 reaches the alarm limit value. The addressed module replies that the command has been received.

@AADAN

| | |
|--------------------|--|
| Name | Disable Alarm |
| Description | Disable Alarm for the specified counter |
| Syntax | <p>@AADAN (cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>DAN represents the Disable Alarm Mode for counter command.</p> <p>N=0 Represents counter 0</p> <p>N=1 Represents counter 1</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! is a delimiter character indicating a command was valid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @12DA0(cr)</p> <p>response: !12(cr)</p> <p>The counter module at address 12h is instructed to disable the alarm functions for counter 0. The module confirms its alarm functions have been disabled.</p> |

@AAPA(data)**@AASA(data)**

| | |
|--------------------|--|
| Name | Set Alarm Limit Value of Counter 0 (or 1) |
| Description | Set the Alarm limit value of counter 0 (or 1) of the specified counter module at address AA. |
| Syntax | <p>@AAPA(data)(cr) @AASA(data)(cr) @ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of a counter module. PA represents the Set Alarm Limit value for counter 0 command. SA represents the Set Alarm Limit Value for counter 1 command. (data) is the alarm limit value which must consist of eight hexadecimal digits. (cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(cr) if the command is valid. There is no response if the module detects a syntax error or communication error or if the specified address does not exist. ! is a delimiter character indicating a command was valid. AA (range 00-FF) represents the 2-character hexadecimal address of a counter module. (cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @\$12PA0000FFFF(cr) response: !12(cr) The command sets the alarm limit value 0000FFFF for counter 0 of the counter module at address 12h. The addressed module replies that the command has been received.</p> |

@AARP

@AARA

| | |
|--------------------|---|
| Name | Read Alarm Limit Value of Counter 0 (or 1) |
| Description | Read the alarm limit value of counter 0 (or 1) of the specified counter module at address AA. |
| Syntax | <p>@AARP(data)(cr)</p> <p>@AARA(data)(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>RP represents the Read Alarm Limit Value for counter 0 command.</p> <p>RA represents the Read Alarm Limit Value for counter 1 command.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Response | <p>!AA(data)(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! is a delimiter character indicating a command was valid.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>(data) is the alarm limit value which must consist of eight hexadecimal digits.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @12RP(cr)</p> <p>response: !120000FFFF(cr)</p> <p>The command instructs the module at address 12h to return the alarm limit value of counter 0. The module replies that the alarm limit value of counter 0 is 0000FFFF.</p> |

@AADO

| | |
|--------------------|--|
| Name | Set Digital Output |
| Description | Set the values of the module's two digital outputs (ON or OFF). |
| Syntax | <p>@AADO(data)(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>DO is the Set Digital Output command</p> <p>(data) is the two-character parameter that sets the state for the two digital output bits of the module, as shown below:</p> <p>00 all D/O bits are OFF</p> <p>01 bit 0 is ON, bit 1 is OFF</p> <p>02 bit 0 is OFF, bit 1 is ON</p> <p>03 all bits are ON</p> <p>(cr) represents terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(cr) if the command was valid</p> <p>?AA(cr) if an invalid parameter was entered</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! delimiter character indicating a valid command was received</p> <p>? delimiter character indicating the command was invalid</p> <p>AA represents the 2-character hexadecimal address of the responding counter module.</p> <p>(cr) represents terminating character, carriage return (0Dh)</p> |
| Example | <p>command: @05DO01(cr)</p> <p>response: !05(cr)</p> <p>The counter module at address 05h is instructed to set digital output channel 1 to ON and digital output channel 2 to OFF. The module confirms the settings.</p> |

@AAI

| | |
|--------------------|--|
| Name | Read Digital Output and Alarm State |
| Description | The addressed counter module is instructed to return the value of its two digital output channels and the state of its alarm |
| Syntax | <p>@AAI(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an counter module.</p> <p>DI identifies the Read Digital Output and Alarm Status command.</p> <p>(cr) represents terminating character, carriage return (0Dh)</p> |
| Response | <p>!AASOO0(cr) if the command was valid</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exists.</p> <p>! delimiter character indicating a valid command was received</p> <p>AA represents the 2-character hexadecimal address of the responding counter module.</p> <p>S (for ADAM-4080) is a hexadecimal number that represents the disable or enable status of alarm mode:</p> <p>0h Counter 0 alarm is disabled, counter 1 alarm is disabled</p> <p>1h Counter 0 alarm is enabled, counter 1 alarm is disabled</p> <p>2h Counter 0 alarm is disabled, counter 1 alarm is enabled</p> <p>3h Counter 0 alarm is enabled, counter 1 alarm is enabled</p> <p>S (for ADAM-4080D) is a hexadecimal number that represents the alarm state of counter 0:</p> <p>0h Alarm of counter 0 is disabled</p> <p>1h Alarm of counter 0 is in MOMENTARY mode</p> <p>2h Alarm of counter 0 is in LATCH mode</p> <p>OO is a hexadecimal number representing the Digital Output port's channel 0 and 1 status (00h = D/O channels 0 and 1 are both OFF, 01h = channel 0 is ON, channel 1 is OFF, 02h = channel 0 is OFF, channel 1 is ON, 03h = channel 0 and 1 are both ON).</p> <p>(cr) represents the terminating character, carriage return (0Dh)</p> |

@AADI**Example**

command: @15DI(cr)

response: !1510000(cr)

(ADAM-4080D)

The counter module at address 15h is instructed to return digital output data and alarm status. The module responds that both digital output channels are OFF and alarm state is Momentary

Example

command: @05DI(cr)

response: !0530000(cr)

(ADAM-4080)

The counter module at address 05H is instructed to return digital output and alarm status. The module responds that both digital output channels are OFF and the alarm status for both counters are ENABLED.

@AAEAT

Name Enable Alarm
Description The addressed counter module is instructed to enable its alarm for counter 0 in either Latching or Momentary mode.
Syntax @AAEAT(cr)

@ is a delimiter character.
 AA (range 00-FF) represents the 2-character hexadecimal address of an counter module.
 EA identifies the Enable Alarm command.
 T indicates the alarm type and can have the value M = Momentary alarm state, or L = Latching alarm state.
 (cr) represents terminating character, carriage return (0Dh)

Response !AA(cr) if the command was valid
 There is no response if the module detects a syntax or communication error, or if the specified address does not exist.
 ! is a delimiter character indicating a valid command was received
 AA represents the 2-character hexadecimal address of the responding counter module.
 (cr) represents terminating character, carriage return (0Dh)

Example
 command: @03EAL(cr)
 response: !03(cr)
 The counter module at address 03h is instructed to enable its alarm for counter 0 in Latching mode.
 The module confirms that the command has been received.

NOTICE: *A counter module requires a maximum of two seconds after it receives an Enable Alarm command to let the settings take effect. During this time, the module can not be addressed to perform any other actions.*

@AADA

| | |
|--------------------|---|
| Name | Disable Alarm |
| Description | Disable all alarm functions for counter 0 of the addressed counter module. |
| Syntax | <p>@AADA(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of an counter module.</p> <p>DA identifies the Disable Alarm command.</p> <p>(cr) represents terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(cr) if the command was valid</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exists.</p> <p>! is a delimiter character indicating a valid command was received</p> <p>AA represents the 2-character hexadecimal address of the responding counter module.</p> <p>(cr) represents terminating character, carriage return (0Dh)</p> |
| Example | <p>command: @07DA(cr)</p> <p>response: !07(cr)</p> <p>The counter module at address 07h is instructed to disable all alarm functions for channel 0. The module confirms its alarm functions have been disabled.</p> |

NOTICE: *A counter module requires a maximum of two seconds after it receives a Disable Alarm command before the settings take effect. During this interval, the module cannot be addressed to perform any other actions.*

@AACA

| | |
|--------------------|---|
| Name | Clear Latch Alarm |
| Description | Both alarm states (High and Low) of the addressed counter module are set to OFF, no alarm. |
| Syntax | @AACA(cr) @ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of an counter module. CA is the Clear Latch Alarm command (cr) represents terminating character, carriage return (0Dh) |
| Response | !AA(cr) if the command was valid 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 indicating a valid command was valid AA represents the 2-character hexadecimal address of the responding counter module. (cr) represents terminating character, carriage return (0Dh) |
| Example | command: @05CA(cr) response: !05(cr) The counter module at address 05h is instructed to set both alarm states (High and Low) to OFF. The module confirms that it has done so. |

@AAPA(data)

| | |
|--------------------|--|
| Name | Set Low-alarm Count Value for Counter 0. |
| Description | Set the low-alarm count value for counter 0 of the specified counter module. |
| Syntax | <p>@AAPA(data)(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of the counter module.</p> <p>PA identifies the Set Low-alarm Count Value command.</p> <p>(data) is the low-alarm count value which must consist of eight hexadecimal digits. This value should be lower than the high alarm value.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command was valid</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @12PA0000FFFF(cr)</p> <p>response: !12(cr)</p> <p>The command sets the low-alarm count number 0000FFFF(hex) for channel 0 of the counter modules at address 12. The addressed module replies that the command has been received.</p> |

@AASA(data)

| | |
|--------------------|---|
| Name | Set Hi-alarm Count Value of Counter 0. |
| Description | Set the high-alarm count value for counter 0 of the specified counter module. |
| Syntax | <p>@AASA(data)(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address a the counter module.</p> <p>SA identifies the Set High-alarm Count Value command.</p> <p>(data) is the high-alarm count value which must consist of eight hexadecimal digits. This value should be higher than the low-alarm count value.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @12SAF0000000(cr)</p> <p>response: !12(cr)</p> <p>The command sets the low-alarm count number F0000000(hex) for channel 0 of the counter modules at address 12. The addressed module replies that the command has been received.</p> |

@AARP

| | |
|--------------------|--|
| Name | Read Low-alarm Count Value of Counter 0 |
| Description | Read the low-alarm value of counter 0 of the specified counter module. |
| Syntax | <p>@AARP(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter.</p> <p>RP identifies the Read Low-alarm Count Value command.</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(data)(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! is a delimiter character indicating a valid command</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>(data) is the low-alarm count value which consists of eight hexadecimal digits.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @12RP(cr)</p> <p>response: !120000FFFF(cr)</p> <p>The command instructs the module at address 12 to return its low-alarm count value. The module replies that counter 0's low-alarm count value is 0000FFFF(hex).</p> |

@AARA

| | |
|--------------------|--|
| Name | Read High-alarm Count Value for Counter 0. |
| Description | Requests the addressed counter module to return its high-alarm count value of counter 0. |
| Syntax | <p>@AARA(cr)</p> <p>@ is a delimiter character.</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>RA identifies the Read High-alarm Count Value command</p> <p>(cr) is the terminating character, carriage return (0Dh)</p> |
| Response | <p>!AA(data)(cr) if the command is valid.</p> <p>There is no response if the module detects a syntax error or communication error or if the specified address does not exist.</p> <p>! a delimiter character indicating a valid command</p> <p>AA (range 00-FF) represents the 2-character hexadecimal address of a counter module.</p> <p>(data) is the high-alarm count value which consists of eight hexadecimal digits.</p> <p>(cr) is the terminating character, carriage return (0Dh).</p> |
| Example | <p>command: @12RA(cr)</p> <p>response: !12F0000000(cr)</p> <p>The command instructs the module at address 12 to return its high-alarm count value. The module replies that counter 0's high-alarm count value is F0000000(hex).</p> |

Calibration

5

Analog input/output modules are calibrated when you receive them. However, calibration is sometimes required. No screwdriver is necessary because calibration is done in software, with calibration parameters stored in the ADAM module's onboard EEPROM.

The ADAM modules come with utility software that supports the calibration of analog input and analog output. Besides the calibration that is carried out through software, the modules incorporate automatic Zero Calibration and automatic Span Calibration at bootup or reset.

5.1 Analog Input Module Calibration

Models: ADAM 4011, 4011D, 4012, 4014D, 4016, 4017, 4017+, 4018, 4018+, 4018M

1. Apply power to the module and let it warm up for a bout 30 minutes
2. Assure that the module is correctly installed and is properly configured for the input range you want to calibrate. You can do this by using the ADAM utility software. (Refer to Appendix D, Utility Software.)
3. Use a precession voltage source to apply a calibration voltage to the module's +IN and -IN terminals of the ADAM-4011, 4011D and 4012. Use a precession voltage source to apply a calibration voltage to the module's Vin+ and Vin- terminals (or Iin+ and Iin-) for the ADAM-4014D and 4016. Use a precession voltage source to apply a calibration voltage to the module's Vin0+ and Vin0- terminals for ADAM-4017, 4017+, 4018, 4018+ and 4018M.

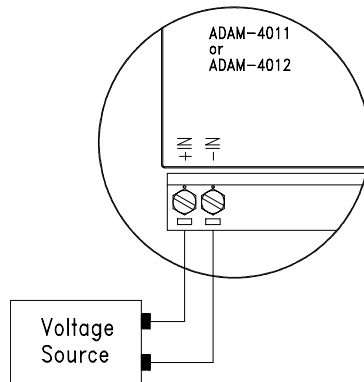


Figure 5-1 *Applying Calibration Voltage*

- Execute the Offset Calibration command. This is also done with the ADAM utility software. (See “Offset Calibration” option in the Calibration sub-menu of the ADAM utility software).



Figure 5-2 Zero Calibration

- Execute the Span Calibration command. This can be done with the ADAM utility software. (See “Span Calibration” option in the Calibration sub-menu of the ADAM utility software).

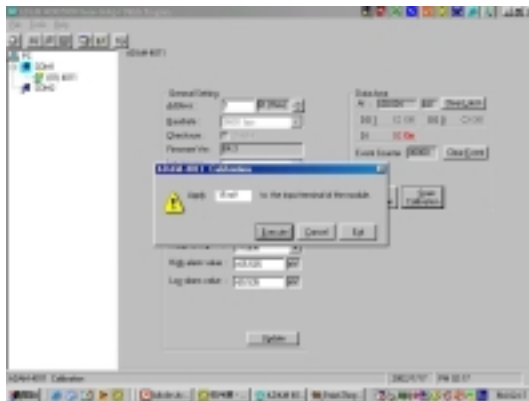


Figure 5-3 Span Calibration

Calibration

6. Only for ADAM 4011, 4011D, 4018, 4018+, 4018M

Execute the CJC (cold junction sensor) calibration command. This is also done with the ADAM utility software. (See “CJC Calibration” option in the Calibration sub-menu of the ADAM utility software).

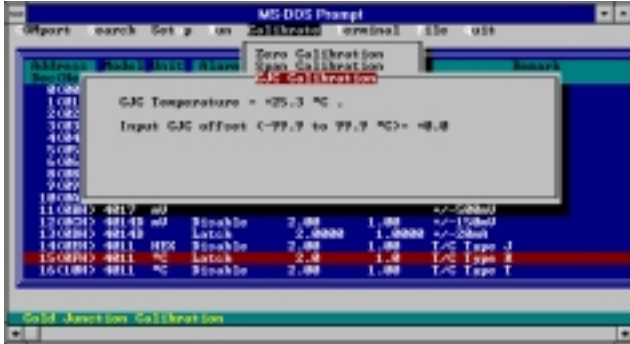


Figure 5-4 Cold Junction Calibration

Table 5-1 Calibration Voltages

| Module | Input Range Code (hex) | Input Range | Span Calibration Voltage |
|---|------------------------|------------------------------|--------------------------|
| 4011 4011D 4018 4018+ 4018M | 00 | -15 mV | +15 mV |
| | 01 | -50 mV | +50 mV |
| | 02 | -100 mV | +100 mV |
| | 03 | -500 mV | +500 mV |
| | 04 | -1 V | +1 V |
| | 05 | -2.5 V | +2.5 V |
| | 06 | -20 mA | +20 mA _r |
| | 0E | J thermocouple 0 to 760 C | +50 mV |
| | 0F | K thermocouple 0 to 1000 C | +50 mV |
| | 10 | T thermocouple -100 to 400 C | +22 mV |
| | 11 | E thermocouple 0 to 1000 C | +80 mV |
| | 12 | R thermocouple 500 to 1750 C | +22 mV |
| | 13 | S thermocouple 500 to 1750 C | +22 mV |
| | 14 | B thermocouple 500 to 1800 C | +15 mV |
| 4012 4017 4017+ | 07 | not used | |
| | 08 | -10 V | +10 V |
| | 09 | -5 V | +5 V |
| | 0A | -1 V | +1 V |
| | 0B | -500 mV | +500 mV |
| | 0C | -150 mV | +150 mV |
| | 0D | -20 mA | +20 mA _r |

NOTE:¹ You can substitute 2.5 V for 20 mA if you remove the current conversion resistor for that channel. However, the calibration accuracy will be limited to 0.1% due to the resistor's tolerance.

(continued on following page)

Calibration

| Module | Input Range Code (Hex) | Input Range | Span Calibration Voltage |
|--------|------------------------|-------------|--------------------------|
| 4016 | 00 | -15 mV | +15 mV |
| | 01 | -50 mV | +50 mV |
| | 02 | -100 mV | +100 mV |
| | 03 | -500 mV | +500 mV |
| | 06 | -20 mA | +20 mA |
| 4014D | 07 | not used | |
| | 08 | -10 V | +10 V |
| | 09 | -5 V | +5 V |
| | 0A | -1 V | +1 V |
| | 0B | -500 mV | +500 mV |
| | 0C | -150 mV | +150 mV |
| | 0D | -20 mA | +20 mA |

5.2 Analog Input Resistance Calibration

Model: ADAM-4013

1. Apply power to the module and let it warm up for about 30 minutes.
2. Assure that the module is correctly installed and is properly configured for the input range you want to calibrate. You can do this by using the ADAM utility software. (Refer to Appendix D, Utility Software.)
3. Apply the reference Span resistance to the terminals of the screw terminals using a 4-wire connection. Refer to Figure 5-5 for the correct wiring diagram and to Table 5-2 to choose the right Span calibration resistance).

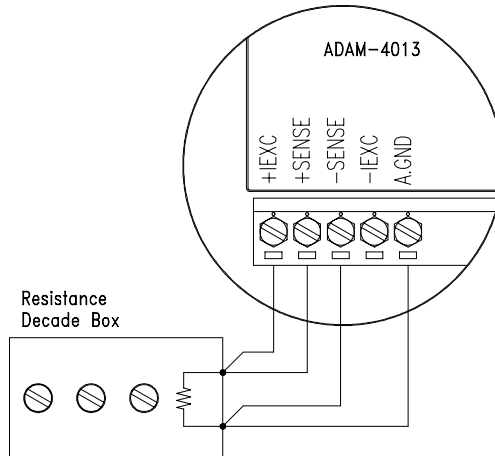


Figure 5-5 *Applying calibration resistance*

Use a precision resistance decade box or discrete resistors with values: 10 W, 15 W, 60 W, 140 W, 200 W and 440 W.

4. Apply the reference Offset resistance to the terminals of the module. (Refer to Figure 5-5 for the correct wiring diagram and to Table 5-2 to choose the right Offset calibration resistance.)

Calibration

5. Issue an Offset Calibration command to the module. This can be done by either using the Calibrate option in the ADAM utility software (See Appendix D, Utility Software)
6. Issue a Span Calibration command to the module. This can be done with the Calibrate option in the ADAM utility software (See Appendix D, Utility Software)

Table 5-2 *Calibration Resistance*

| Module | Input Range Code (Hex) | Input Range | Span Calibration Resistance | Offset Calibration Resistance |
|--------|------------------------|--|-----------------------------|-------------------------------|
| 4013 | 20 | Pt, -100° C to 100° C $\alpha = 0.00385$ | 140 Ω | 60 Ω |
| | 21 | Pt, 0° C to 100° C $\alpha = 0.00385$ | 140 Ω | 60 Ω |
| | 22 | Pt, 0° C to 200° C $\alpha = 0.00385$ | 200 Ω | 60 Ω |
| | 23 | Pt, 0° C to 600° C $\alpha = 0.00385$ | 440 Ω | 60 Ω |
| | 24 | Pt, -100° C to 100° C $\alpha = 0.003916$ | 140 Ω | 60 Ω |
| | 25 | Pt, 0° C to 100° C $\alpha = 0.003916$ | 140 Ω | 60 Ω |
| | 26 | Pt, 0° C to 200° C $\alpha = 0.003916$ | 200 Ω | 60 Ω |
| | 27 | Pt, 0° C to 600° C $\alpha = 0.003916$ | 440 Ω | 60 Ω |
| | 28 | Ni, -80 C to 100 C | 200 Ω | 60 Ω |
| | 29 | Ni, 0 C to 100 C | 200 Ω | 60 Ω |

5.3 Analog Output Calibration

Model: ADAM-4021

The output current of analog output modules can be calibrated by using a low calibration value and a high calibration value. The analog output modules can be configured for one of two ranges: 0-20 mA and 4-20 mA. Since the low limit of the 0 - 20 mA range, 0 mA, is internally an absolute reference (no power, or immeasurably small power) just two levels are needed for calibration: 4 mA and 20 mA.

1. Apply power to the analog output module and let it warm up for about 30 minutes.
2. Assure that the module is correctly installed and that its configuration is according your specifications and matches the output range you want to calibrate. You can do this by using the ADAM utility software. (Refer to Appendix D, Utility Software.)
3. Connect either a 5-digit millimeter or voltmeter with a shunt resistor (250 Ω , 0.01% , and 10ppm) to the screw terminals of the module

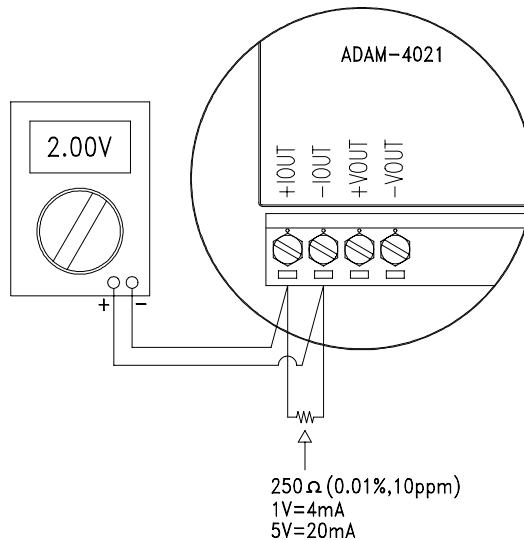


Figure 5-6 Setup for Analog Output Calibration

Calibration

4. Issue the Analog Data Out command to the module with an output value of 4 mA.
5. Check the actual output value at the modules terminals . If this does not equals 4 mA, use the “Trim” option in the “CaLibrate” sub menu to change the actual output. Trim the module until the milliammeter indicates exactly 4 mA or in case of an voltage meter with shunt resistor, the voltage meter indicates exactly 1 V. (When calibrating for 20 mA using voltage meter and shunt resistor, the correct voltage would be 5 V)
6. Issue the 4 mA Calibration command to indicate that the output is calibrated and calibration parameters should be stored in the modules EEPROM.
7. Execute an Analog Data Out command with an output value of 20 mA. The module’s output will be approximately 20 mA.
8. Execute the Trim Calibration command as often as necessary until the output current is equal to exactly 20 mA.
9. Execute the 20 mA Calibration command to indicate that the present output is exactly 20 mA. The analog output module will store its calibration parameters in EEPROM.

Module: ADAM-4016

1. Apply power to the strain gauge input module and let it warm up for about 30 minutes.
2. Assure that the module is correctly installed. Connect a voltmeter to the screw terminals of the module.

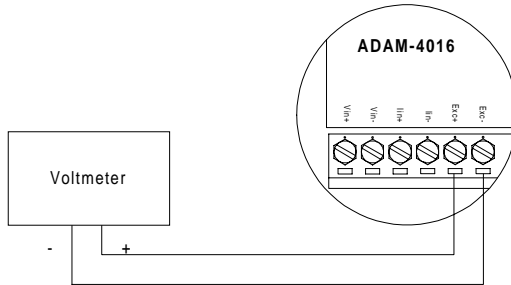


Figure 5-7 Setup for Voltage Output Calibration

3. Execute the Zero Calibration command. This is also done with the ADAM utility software. (See "A/O 0V Calibration" option in the Calibration sub-menu of the ADAM utility software.)



Figure 5-8 Zero Calibration

Calibration

- Execute the Span Calibration command. This can be done with the ADAM utility software. (See "A/O 10 V Calibration" option in the Calibration sub-menu of the ADAM utility software.)

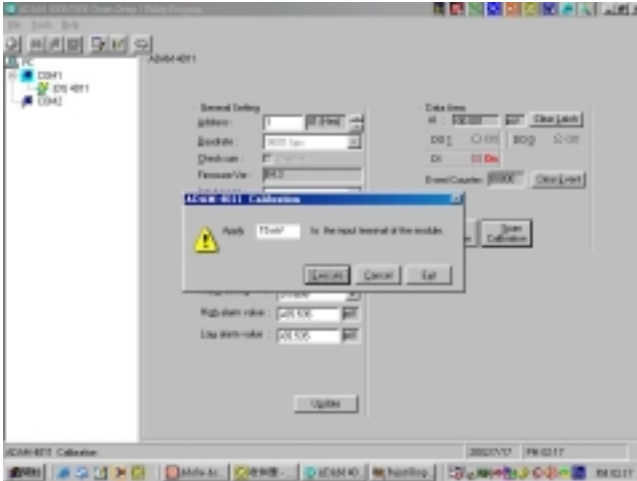


Figure 5-9 *Span Calibration*

- Check the actual output value at the module's terminals. If this does not equal 0 V or 10 V, use the Trim Calibration command to change the output value. This is also done with the ADAM utility software. (Use the arrow key to adjust the output value after executing the "A/O 0V Calibration" or "A/O 10 V Calibration", use the arrow key to adjust the output value. Right and Up keys mean increasing, Left and Down keys mean decreasing).

Technical Specifications

A

A.1 ADAM-4011 Thermocouple Input Module

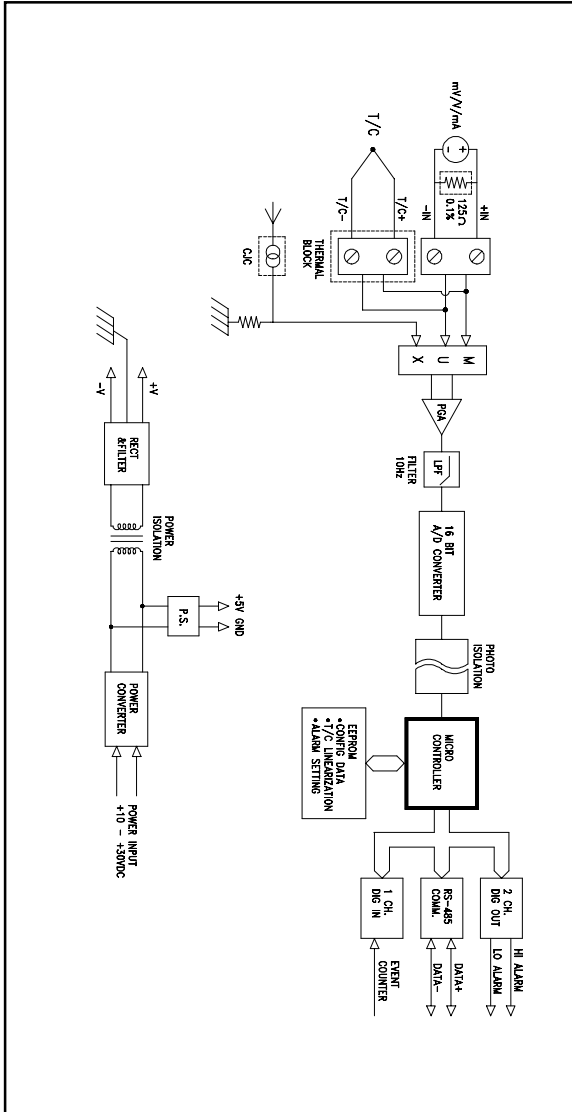
Table A-1 ADAM-4011 Specifications

| | |
|--------------------------------|---|
| Input range | J, K, T, E, R, S and B Thermocouple ± 15 mV, ± 50 mV, ± 100 mV, ± 500 mV, ± 1 V, ± 2.5 V, and ± 20 mA |
| Output | RS-485 (2-Wire) |
| speed (in bps) | 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| maximum distance | 4000 ft. (1200 m.) |
| Accuracy | $\pm 0.05\%$ or better |
| Zero drift | ± 0.3 μ V/ $^{\circ}$ C |
| Span drift | ± 25 ppm / $^{\circ}$ C |
| Isolation-rated voltage | 500 V _{DC} |
| CMR @ 50/60 Hz | 150 dB |
| NMR @ 50/60 Hz | 100 dB |
| Bandwidth | 4 Hz |
| Conversion rate | 10 samples/sec. |
| Input impedance | 2 M Ω |
| Digital output | 2 channels open collector to 30 V |
| sink current | 30 mA max. load |
| power dissipation | 300 mW |
| Digital input | 1 channel |
| logic level 0 | +1 V max. |
| logic level 1 | +3.5 to +30 V |
| pull up current | 0.5 mA |
| Event counter | |
| Max. input frequency | 50 Hz |
| Min. pulse width | 1 msec |
| Watchdog timer | Yes |
| Power supply | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 1.2 W |

Table A-2 ADAM-4011 Range Accuracy for Thermocouple

| Input Range Code (Hex) | Input Range | Typical Accuracy | Maximum Error | Units |
|------------------------|------------------------------|------------------|---------------|-------|
| 0E | J thermocouple 0 to 760 C | -0.5 | -0.75 | C |
| 0F | K thermocouple 0 to 1000 C | -0.5 | -0.75 | C |
| 10 | T thermocouple -100 to 400 C | -0.5 | -0.75 | C |
| 11 | E thermocouple 0 to 1000 C | -0.5 | -0.75 | C |
| 12 | R thermocouple 500 to 1750 C | -0.6 | -1.5 | C |
| 13 | S thermocouple 500 to 1750 C | -0.6 | -1.5 | C |
| 14 | B thermocouple 500 to 1800 C | -1.2 | -2.0 | C |

Figure A-1 ADAM-4011 Function Diagram



A.2 ADAM-4011D Thermocouple Input Module with LED Display

Table A-3 ADAM-4011D Specifications

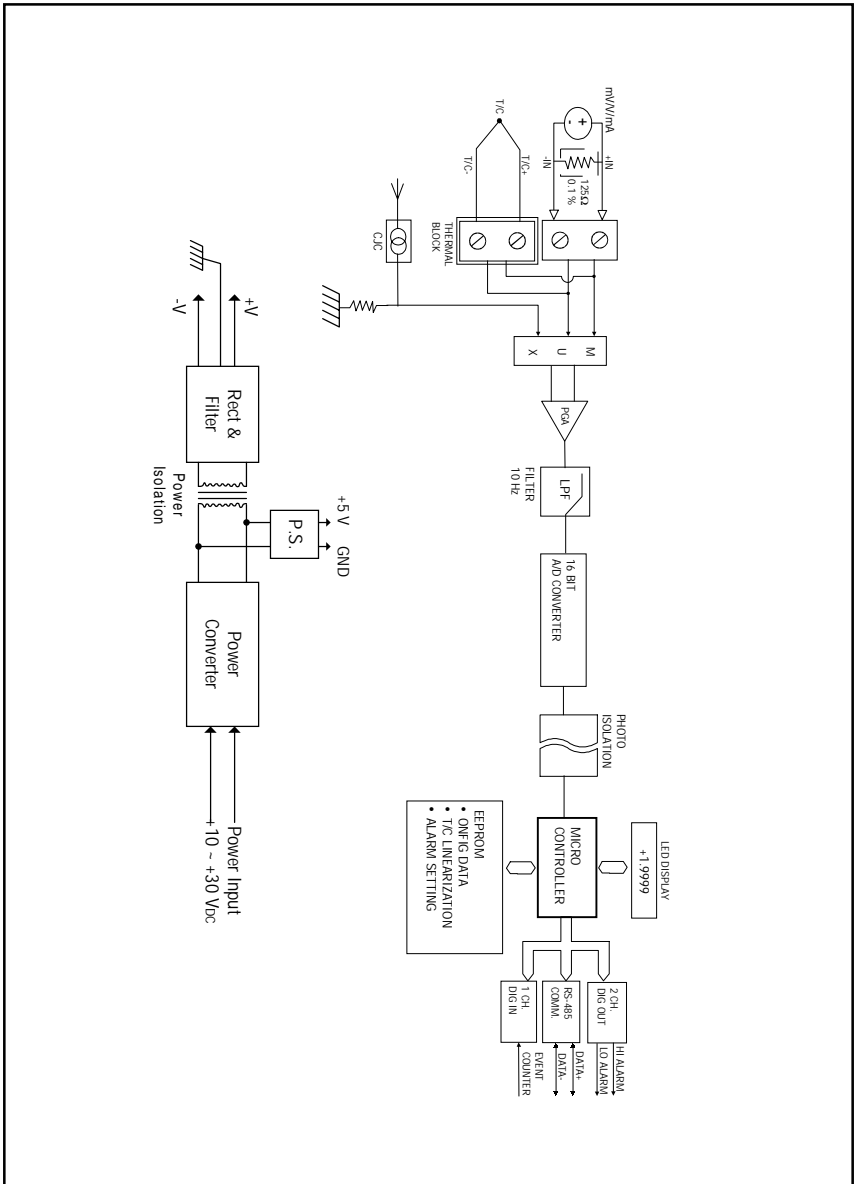
| | |
|--------------------------------|--|
| Input range | J, K, T, E, R, S and B Thermocouple ±15 mV, ±50 mV, ±100 mV, ±500mV, ±1 V, ±2.5 V, and ±20mA |
| Output | RS-485 (2-Wire) |
| speed (in bps) | 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| maximum distance | 4000 ft. (1200 m.) |
| Accuracy | ±0.05% or better |
| Zero drift | ±0.3 μ V / °C |
| Span drift | ±25 ppm / °C |
| Isolation-rated voltage | 3000 V _{DC} |
| CMR @ 50/60 Hz | 150 dB |
| NMR @ 50/60 Hz | 100 dB |
| Bandwidth | 4 Hz |
| Conversion rate | 10 samples/sec. |
| Input impedance | 2 M Ω |
| LED indicator | 4½ digit readout |
| Digital output | 2 channels open collector to 30 V |
| sink current | 30 mA max. load |
| power dissipation | 300 mW |
| Digital input | 1 channel |
| logic level 0 | +1 V max. |
| logic level 1 | +3.5 to +30 V |
| pull up current | 0.5 mA |
| Event counter | |
| Max. input frequency | 50 Hz |
| Min. pulse width | 1 msec |
| Watchdog timer | Yes |
| Power supply | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 1.4 W |

Technical specifications

Table A-4 ADAM-4011D Range Accuracy for Thermocouple

| Input Range Code (Hex) | Input Range | Typical Accuracy | Maximum Error | Units |
|------------------------|-------------------------------|------------------|---------------|-------|
| 0E | J thermocouple 0 to 760 °C | -0.5 | -0.75 | °C |
| 0F | K thermocouple 0 to 1370 °C | -0.5 | -0.75 | °C |
| 10 | T thermocouple -100 to 400 °C | -0.5 | -0.75 | °C |
| 11 | E thermocouple 0 to 1000 °C | -0.5 | -0.75 | °C |
| 12 | R thermocouple 500 to 1750 °C | -0.6 | -1.5 | °C |
| 13 | S thermocouple 500 to 1750 °C | -0.6 | -1.5 | °C |
| 14 | B thermocouple 500 to 1800 °C | -1.2 | -2.0 | °C |

Figure A-2 ADAM-4011D Function Diagram

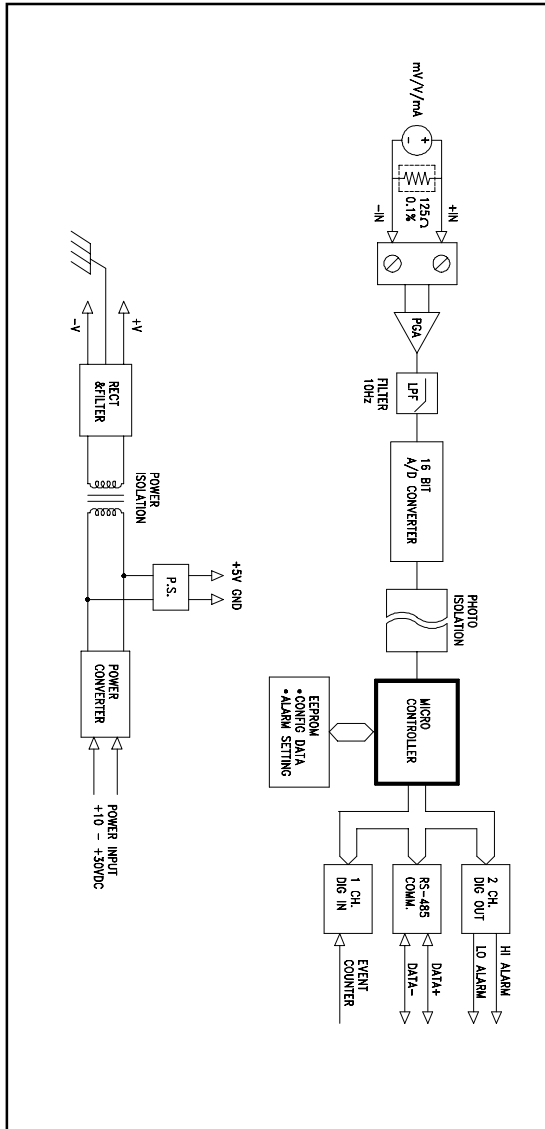


A.3 ADAM-4012 Analog Input Module

Table A-5 *ADAM-4012 Specifications*

| | |
|--------------------------------|---|
| Input range | ± 150 mV, ± 500 mV, ± 1 V, ± 5 V, ± 10 V and ± 20 mA |
| Output | RS-485 (2-wire) |
| speed (in bps) | 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| maximum distance | 4000 ft. (1200 m.) |
| Accuracy | $\pm 0.05\%$ or better |
| Zero drift | ± 6 μ V / $^{\circ}$ C |
| Span drift | ± 25 ppm / $^{\circ}$ C |
| Isolation-rated voltage | 3000 V _{DC} |
| CMR @ 50/60 Hz | 150 dB |
| NMR @ 50/60 Hz | 100 dB |
| Bandwidth | 4 Hz |
| Conversion rate | 10 samples/sec. |
| Input impedance | 20 M Ω |
| Digital output | 2 channels open collector to 30 V |
| sink current | 30 mA max. load |
| power dissipation | 300 mW |
| Digital input | 1 channel |
| logic level 0 | +1 V max. |
| logic level 1 | +3.5 to +30 V |
| pull up current | 0.5 mA |
| Event counter | |
| Max. input frequency | 50 Hz |
| Min. pulse width | 1 msec |
| Watchdog timer | Yes |
| Power supply | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 1.2 W |

Figure A-3 ADAM-4012 Function Diagram

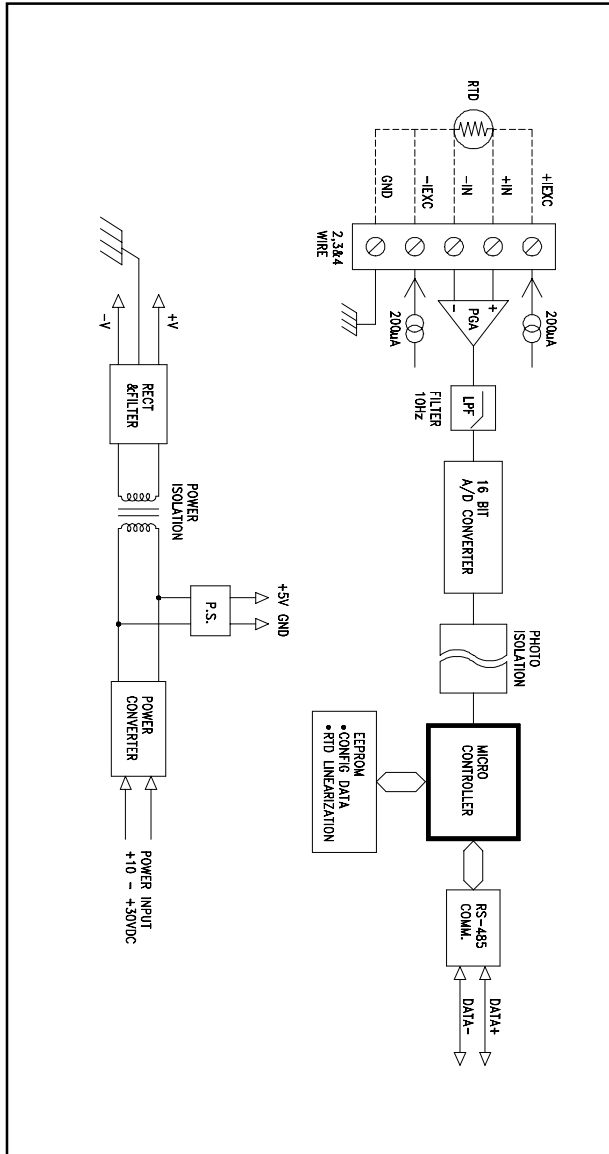


A.4 ADAM-4013 RTD Input Module

Table A-6 *ADAM-4013 Specifications*

| | |
|--------------------------------|--|
| Input range | Pt and Ni RTD |
| Output | RS-485 (2-Wire) |
| speed (in bps) | 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| maximum distance | 4000 ft. (1200 m.) |
| Accuracy | ±0.05% or better |
| Zero drift | ±0.01 °C/ °C |
| Span drift | ±0.01 °C/ °C |
| Input connections | 2, 3, or 4 wires |
| Isolation-rated voltage | 3000 V _{DC} |
| CMR @ 50/60 Hz | 150 dB |
| NMR @ 50/60 Hz | 100 dB |
| Bandwidth | 4 Hz |
| Conversion rate | 10 samples/sec. |
| Input impedance | 2 MΩ |
| Watchdog timer | Yes |
| Power supply | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 0.7 W |

Figure A-4 ADAM-4013 Function Diagram

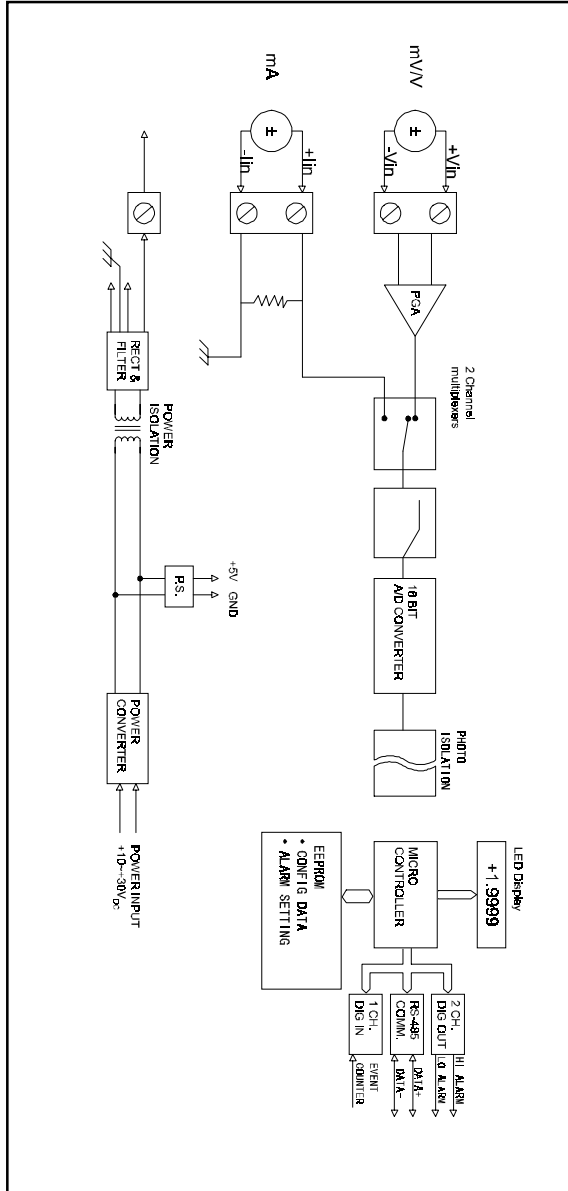


A.5 ADAM-4014D Analog Input Module with LED Display

Table A-7 ADAM-4014D Specifications

| | |
|----------------------------|---|
| Input range | ± 150 mV, ± 500 mV, ± 1 V, ± 5 V, ± 10 V and ± 20 mA |
| Output | RS-485 (2-wire) |
| Speed (bps) | 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| Maximum distance | 4000 ft. (1200 m) |
| Isolation voltage | 500V _{DC} |
| Sampling rate | 10 samples/sec |
| Bandwidth | 4 Hz |
| Accuracy | $\pm 0.05\%$ or better |
| Zero drift | ± 6 μ V/ $^{\circ}$ C |
| Span drift | ± 25 ppm / $^{\circ}$ C |
| CMR @ 50/60 Hz | 150 dB |
| NMR @ 50/60 Hz | 100 dB |
| Isolated loop power | +15 VDC @ 30 mA |
| Input impedance | 40 K Ω |
| LED indicator | 4½ -digit readout |
| Digital input | 1 channel |
| logic level 0 | +1 V max. |
| logic level 1 | +3.5 to +30 V |
| pull up current | 0.5 mA, 10 K resistor to +5 V |
| Event counter | |
| Max. input frequency | 50 Hz |
| Min. pulse width | 0.5 msec |
| Digital output | 2 channels open collector to 30 V |
| sink current | 30 mA max. load |
| power dissipation | 300 mW |
| Watchdog timer | Yes |
| Power requirements | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 1.8 W |

Figure A-5 ADAM-4014D Function Diagram

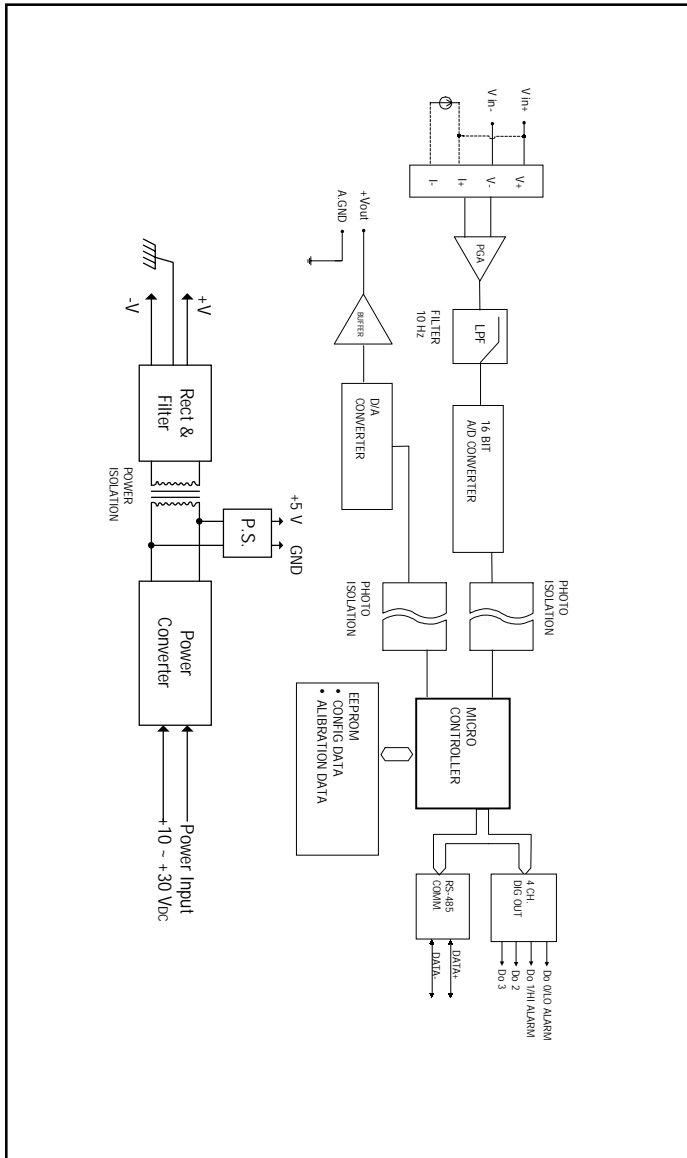


A.6 ADAM-4016 Strain Gauge Input Module

Table A-8 ADAM-4016 Specifications

| | |
|----------------------------|--|
| Input range | ± 15 mV, ± 50 mV, ± 100 mV, ± 500 mV and ± 20 mA |
| Output | RS-485 (2-wire) |
| Speed (bps) | 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| Maximum distance | 4000 ft. (1200 m) |
| Isolation voltage | 3000 V _{DC} |
| Sampling rate | 10 samples/sec |
| Bandwidth | 4 Hz |
| Accuracy | $\pm 0.05\%$ or better |
| Zero drift | ± 6 μ V / °C |
| Span drift | ± 25 ppm / °C |
| CMR @ 50/60 Hz | 150 dB |
| NMR @ 50/60 Hz | 100 dB |
| Analog output range | 0–10 V |
| Drive current | 30 mA |
| Isolation voltage | 3000 V _{DC} |
| Accuracy | 0.05% of FSR |
| Drift | ± 50 ppm / °C |
| Input impedance | 2 M Ω |
| Digital output | 4 channels open collector to 30 V |
| sink current | 30 mA max. load |
| power dissipation | 300 mW |
| Watchdog timer | Yes |
| Power requirements | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 2.2 W |

Figure A-6 ADAM-4016 Function Diagram

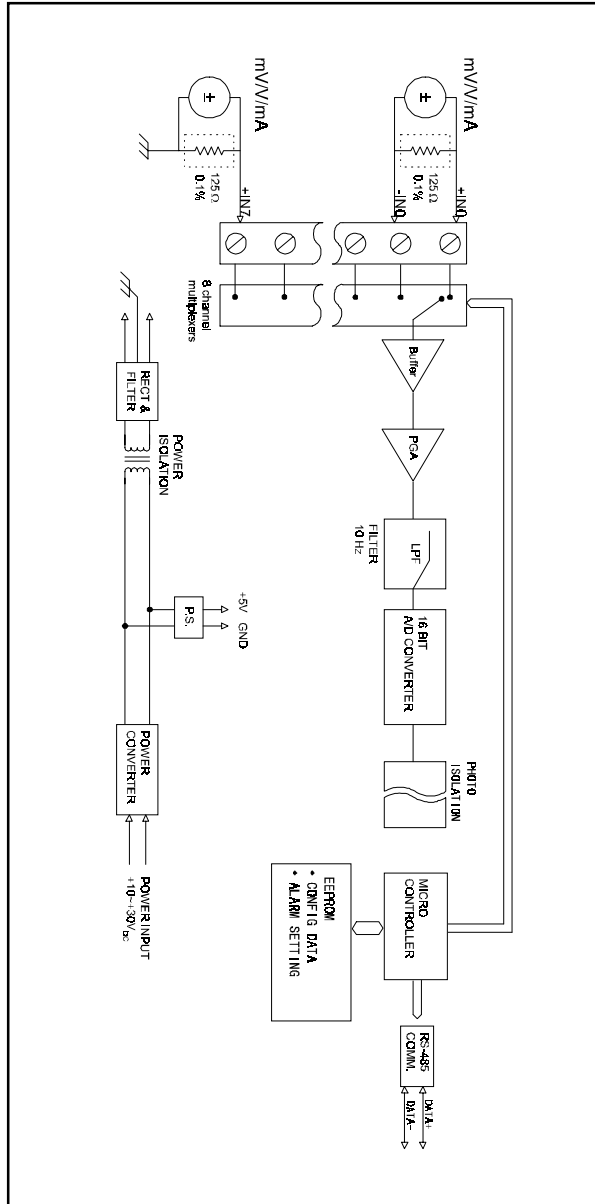


A.7 ADAM-4017, 4017+ 8-Channel Analog Input Module

Table A-9 ADAM-4017 Specifications

| | |
|--|---|
| Analog input channels | Six differential Two single-ended |
| Input type | mV, V, and mA |
| Input range | ± 150 mV, ± 500 mV, ± 1 V, ± 5 V, ± 10 V and ± 20 mA |
| Output speed (bps) maximum distance | RS-485 (2-wire) 1200, 2400, 4800, 9600, 19.2K, 38.4K 4000 ft. (1200 m) |
| Isolation voltage | 3000 V _{DC} |
| Sampling rate | 10 samples/sec(total) |
| Bandwidth | 13.1 Hz |
| Accuracy | $\pm 0.1\%$ or better |
| Zero drift | ± 6 μ V/ $^{\circ}$ C |
| Span drift | ± 25 ppm / $^{\circ}$ C |
| CMR @ 50/60 Hz | 92 dB |
| Input impedance | 2 M Ω |
| Watchdog timer | Yes |
| Power requirements | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 1.2 W |

Figure A-7 ADAM-4017 Function Diagram



A.8 ADAM-4018, 4018+ 8-channel Analog Input Module

Table A-10 *ADAM-4018 Specifications*

| | |
|-----------------------|---|
| Analog input channels | Eight differential for ADAM-4018 |
| Input type | mV, V, and mA |
| Input range | J, K, T, E, R, S and B Thermocouple ± 15 mV, ± 50 mV, ± 100 mV, ± 500 mV, ± 1 V, ± 2.5 V, and ± 20 mA |
| Output speed (bps) | RS-485 (2-wire) 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| maximum distance | 4000 ft. (1200 m) |
| Isolation voltage | 3000 V _{DC} |
| Sampling rate | 10 samples/sec(total) |
| Bandwidth | 13.1 Hz |
| Accuracy | $\pm 0.1\%$ or better |
| Zero drift | $\pm 0.3 \mu\text{V} / ^\circ\text{C}$ |
| Span drift | ± 25 ppm / $^\circ\text{C}$ |
| CMR @ 50/60 Hz | 92 dB |
| Input impedance | 1.8 M Ω |
| Watchdog timer | Yes |
| Power requirements | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 0.8 W |

Figure A-8 *ADAM-4018 Function Diagram*

Table A-11 ADAM-4018/4018+ Range Accuracy for Thermocouple

| Input Range Code (Hex) | Input Range | Typical Accuracy | Maximum Error | Units |
|------------------------|-------------------------------|------------------|---------------|-------|
| 0E | J thermocouple 0 to 760 °C | -1.0 | -1.5 | °C |
| 0F | K thermocouple 0 to 1000 °C | -1.0 | -1.5 | °C |
| 10 | T thermocouple -100 to 400 °C | -1.0 | -1.5 | °C |
| 11 | E thermocouple 0 to 1000 °C | -1.0 | -1.5 | °C |
| 12 | R thermocouple 500 to 1750 °C | -1.2 | -2.5 | °C |
| 13 | S thermocouple 500 to 1750 °C | -1.2 | -2.5 | °C |
| 14 | B thermocouple 500 to 1800 °C | -2.0 | -3.0 | °C |

NOTE: Because the CJC sensor of ADAM-4018/4018M is located in the side of channel 0 to 4, the measurement will have the difference ± 1 °C between channel 0 ~ 4 and channel 5 ~ 7.

A.9 ADAM-4018M 8-channel Analog Input Data Logger

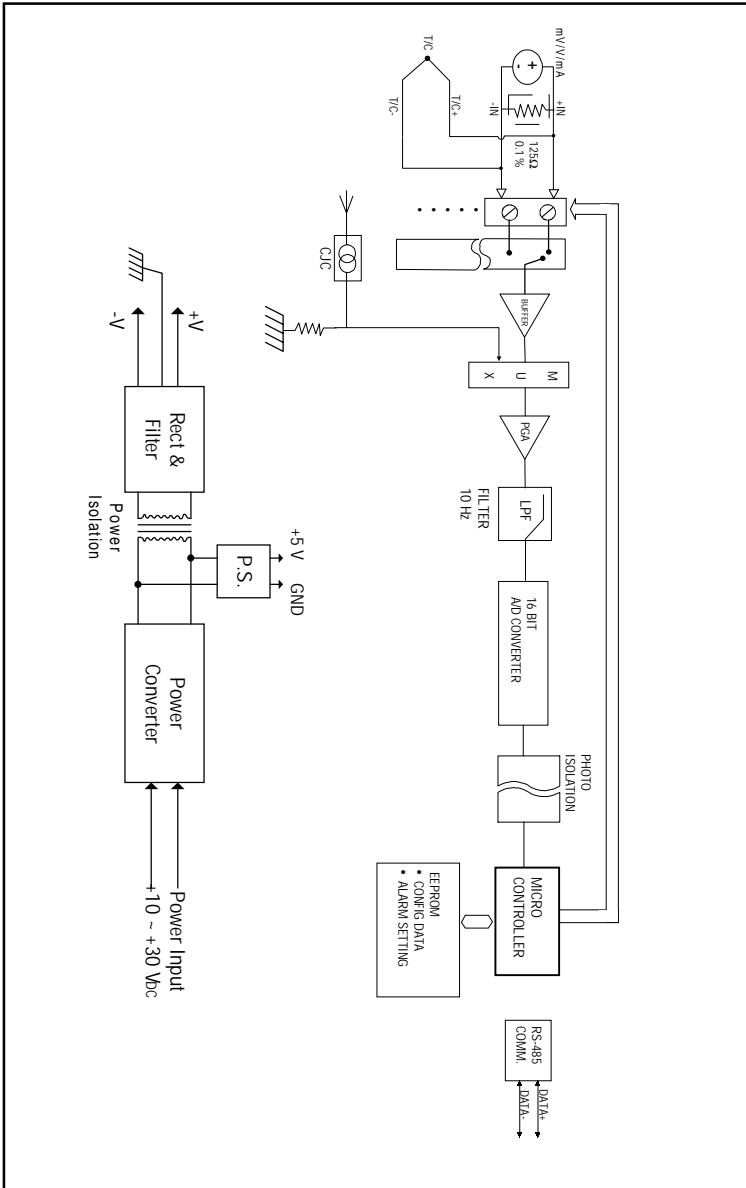


Table A-12 ADAM-4018M Specifications

| | |
|---|---|
| Analog input channels | Six differential Two single-ended |
| Input type | mV, V, and mA |
| Input range | J, K, T, E, R, S and B Thermocouple ±15 mV, ±50 mV, ±100 mV, ±500mV, and ±20mA |
| Output speed (bps) maximum distance | RS-485 (2-wire) 1200, 2400, 4800, 9600, 19.2K, 38.4K 4000 ft. (1200 m) |
| Isolation voltage | 500 V _{DC} |
| Sampling rate | 10 samples/sec(total) |
| Bandwidth | 13.1 Hz |
| Accuracy | ±0.1% or better |
| Zero drift | ±0.3 μV / °C |
| Span drift | ±25 ppm / °C |
| CMR @ 50/60 Hz | 92 dB min |
| Input impedance | 1.8 MΩ |
| Storage capacity Standard log Event log Mixed log | 128 KB Flash memory 38,000 samples (total) 16,300 samples (total) 16,300 data samples and 9,300 event samples (total) |
| Storage type | Write to end of memory, or circular memory |
| Logging mode | Standard log, Event log or Mixed log |
| Sampling interval | 2 sec. to 18 hours |
| Measurement duration | 330 minutes to 20 years |
| Watchdog timer | Yes |
| Power requirements | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 1.8 W |

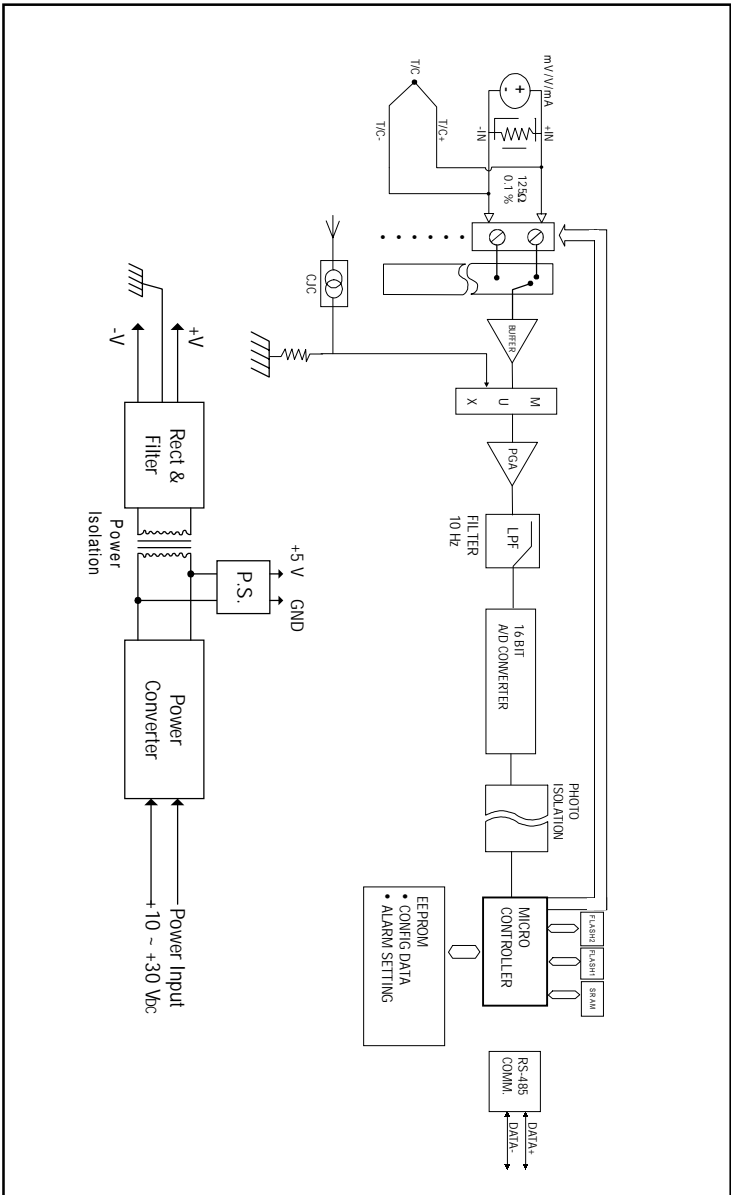
A.10 ADAM-4021 Analog Output Module

Table A-13 ADAM-4018M Range Accuracy for Thermocouple

| Input Range Code (Hex) | Input Range | Typical Accuracy | Maximum Error | Units |
|------------------------|------------------------------|------------------|---------------|-------|
| 0E | J thermocouple 0 to 760 C | -1.0 | -1.5 | C |
| 0F | K thermocouple 0 to 1000 C | -1.0 | -1.5 | C |
| 10 | T thermocouple -100 to 400 C | -1.0 | -1.5 | C |
| 11 | E thermocouple 0 to 1000 C | -1.0 | -1.5 | C |
| 12 | R thermocouple 500 to 1750 C | -1.2 | -2.5 | C |
| 13 | S thermocouple 500 to 1750 C | -1.2 | -2.5 | C |
| 14 | B thermocouple 500 to 1800 C | -2.0 | -3.0 | C |

NOTE: Because the CJC sensor of ADAM-4018/4018M is located in the side of channel 0 to 4, the measurement will have the difference ± 1 °C between channel 0 ~ 4 and channel 5 ~ 7.

Figure A-9 ADAM-4018M Function Diagram

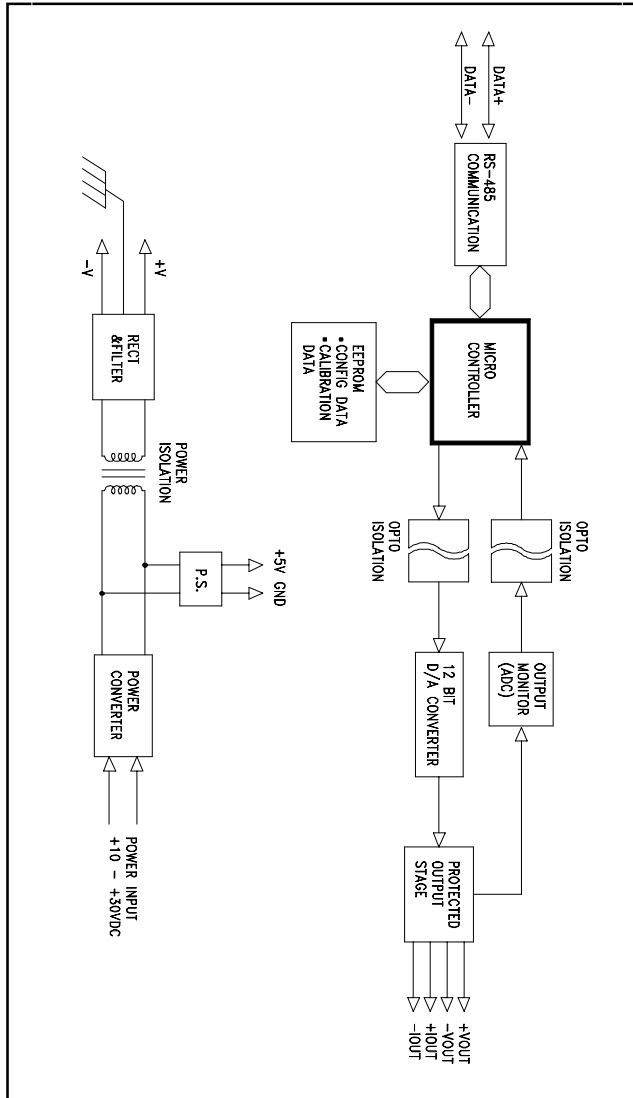


Technical specifications

Table A-14 ADAM-4021 Specifications

| | |
|-------------------------------------|--|
| Output range | mA, V |
| Input | RS-485 (2-wire) |
| speed (bps) | 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| maximum distance | 4000 ft. (1200 m) |
| Accuracy | ±0.1% of FSR for current output ±0.2% of FSR for voltage output |
| Readback accuracy | ±0.1% of FSR |
| Zero drift | |
| voltage output | ±30 μ V/ °C |
| current output | ±0.2 μ A/ °C |
| Span temperature coefficient | ±25 ppm/ °C |
| Isolation rated voltage | 3000 V _{DC} |
| Programmable output slope | 0.125 to 128.0 mA/s 0.0625 to 64 V/s |
| Current load resistor | 0 to 500 Ω (source) |
| Bandwidth | 100 samples per second |
| Output impedance | 0.5 Ω |
| Watchdog timer | Yes |
| Power supply | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 1.4 W |

Figure A-10 ADAM-4021 Function Diagram

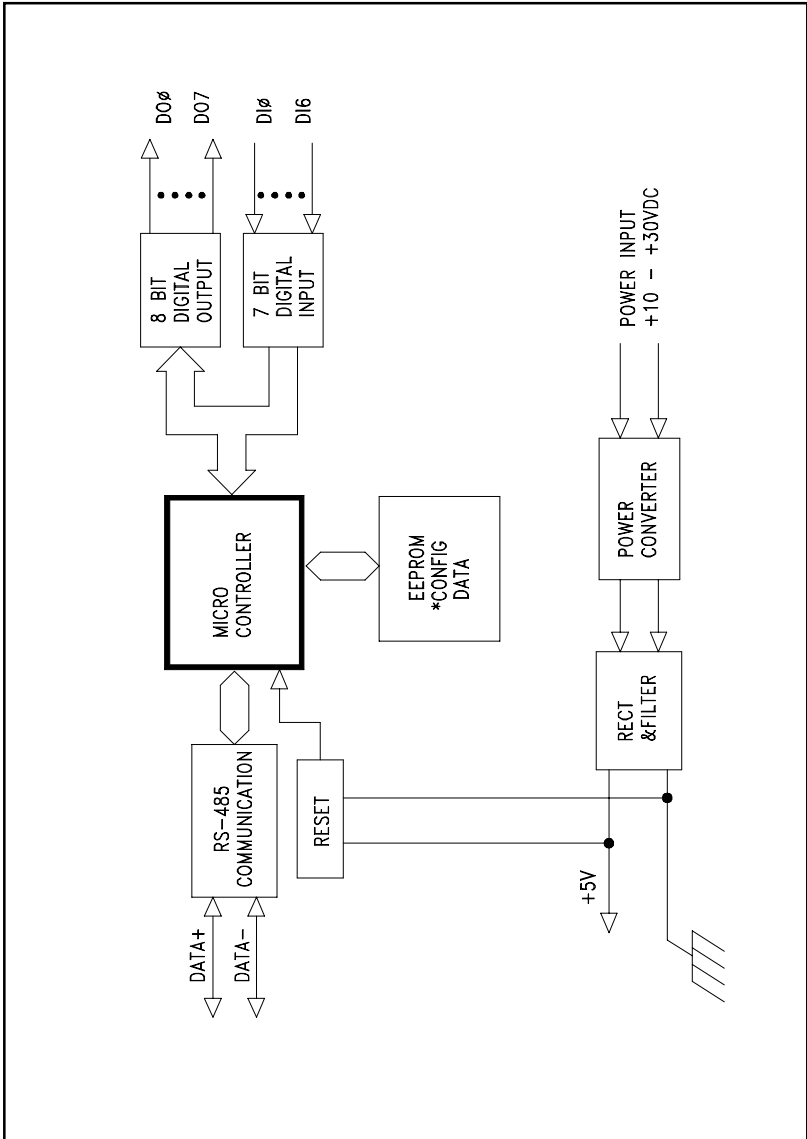


A.11 ADAM-4050 Digital I/O Module

Table A-15 *ADAM-4050 Specifications*

| | |
|---|--|
| I/ O Channels | 7 inputs 8 outputs |
| Input/Output speed (bps) maximum distance | RS-485 (2-wire) 1200, 2400, 4800, 9600, 19.2K, 38.4K 4000 ft. (1200 m) |
| Digital Output sink-current power dissipation | 8-channel open collector to 30 V 30 mA 300 mW |
| Digital Input logic level 0 logic level 1 Pull-up current | 7-channel +1 V max. +3.5 to +30 V 0.5 mA, 10K resistor to +5 V |
| Watchdog timer | Yes |
| Power supply | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 0.4 W |

Figure A-11 ADAM-4050 Function Diagram

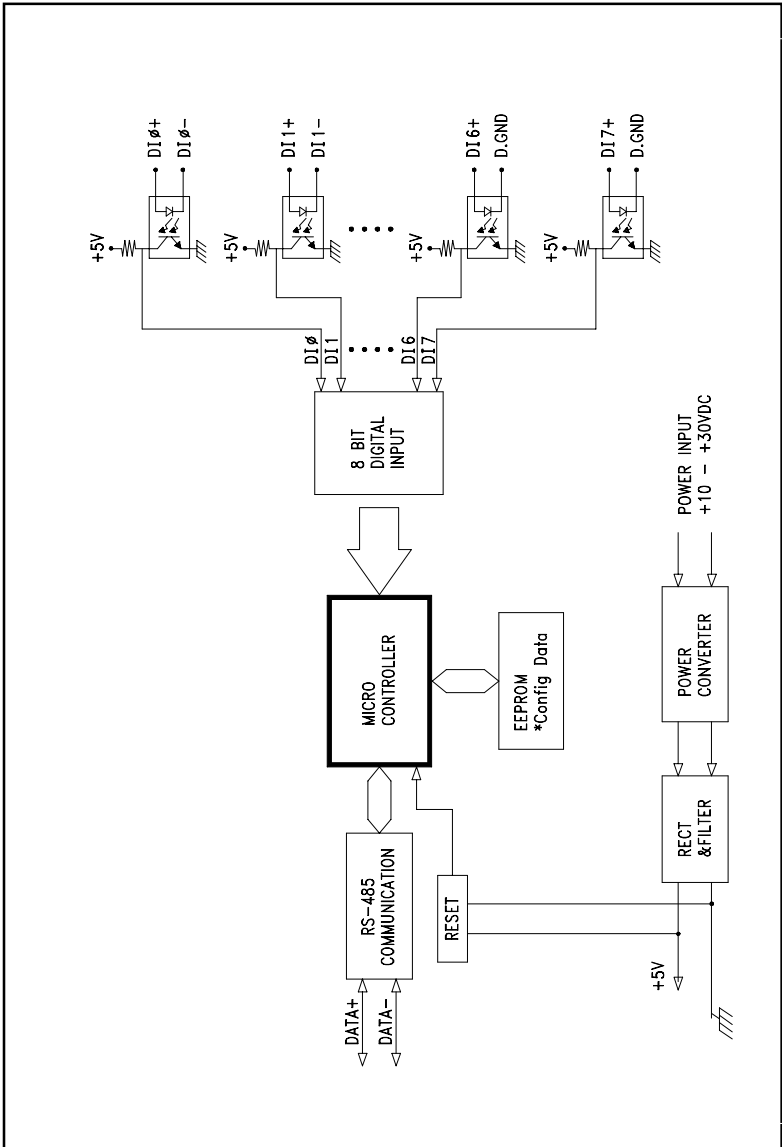


A.12 ADAM-4052 Isolated Digital Input Module

Table A-16 ADAM-40521/4052 Specifications

| | |
|--|--|
| I/ O channels | 16 isolated input for ADAM-4051 8 inputs (6 fully independent isolated channels, and 2 isolated channels with common ground) for ADAM-4052 |
| Input/Output speed (bps) maximum distance | RS-485 (2-wire) 1200, 2400, 4800, 9600, 19.2K, 38.4K 4000 ft. (1200 m) |
| Digital input | ADAM-4051 |
| Dry Contact: | |
| Logic level 0: | Close to GND |
| Logic level 1: | Open |
| Wet Contact: | |
| Logic level 0: | +3V max. |
| Logic level 1: | +10 to 50V |
| ADAM-4051/4052 | |
| Logic level 0: | +1V max. |
| Logic level 1: | +3.5 to +30 V |
| Input resistance | 3 k Ω @ 0.5 W |
| Isolation voltage | 5000 V _{RMS} |
| Watchdog timer | Yes |
| Power supply | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 0.4 W |

Figure A-12 ADAM-4052 Function Diagram

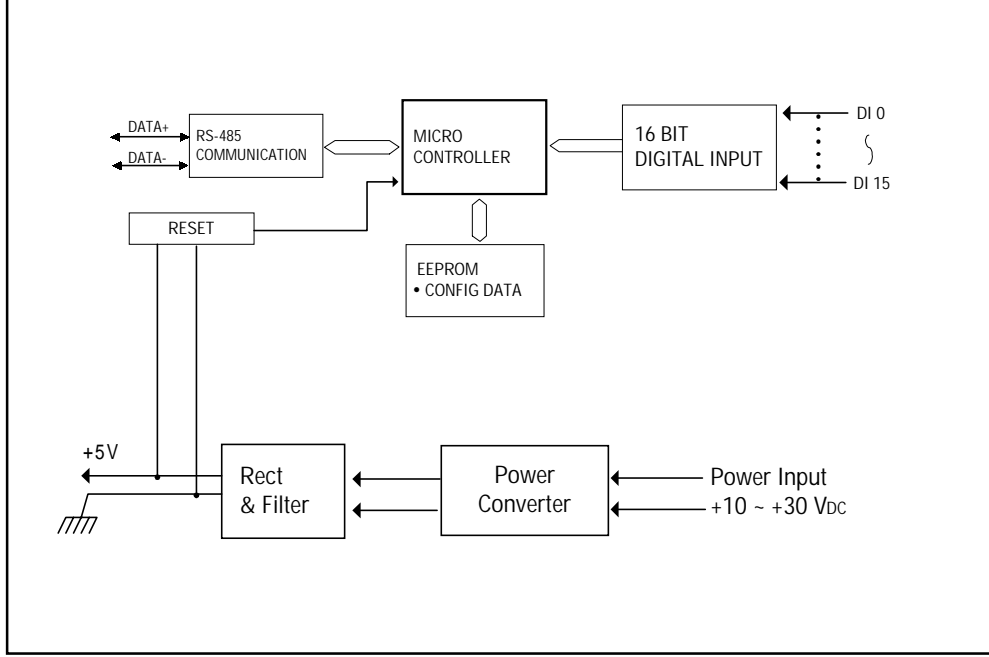


A.13 ADAM-4053 16-channel Digital Input Module

Table A-17 *ADAM-4053 Specifications*

| | |
|--|--|
| I/ O channels | 16 inputs |
| Input/Output speed (bps) | RS-485 (2-wire) 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| maximum distance | 4000 ft. (1200 m) |
| Digital Input | Dry Contact: logic level 0: close to GND logic level 1: open Wet Contact: logic level 0: +2V max logic level 1: +4 V to +30 V |
| Effective distance (dry contact only) | 500 m max. |
| Watchdog timer | Yes |
| Power supply | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 1.0 W |

Figure A-13 ADAM-4053 Function Diagram

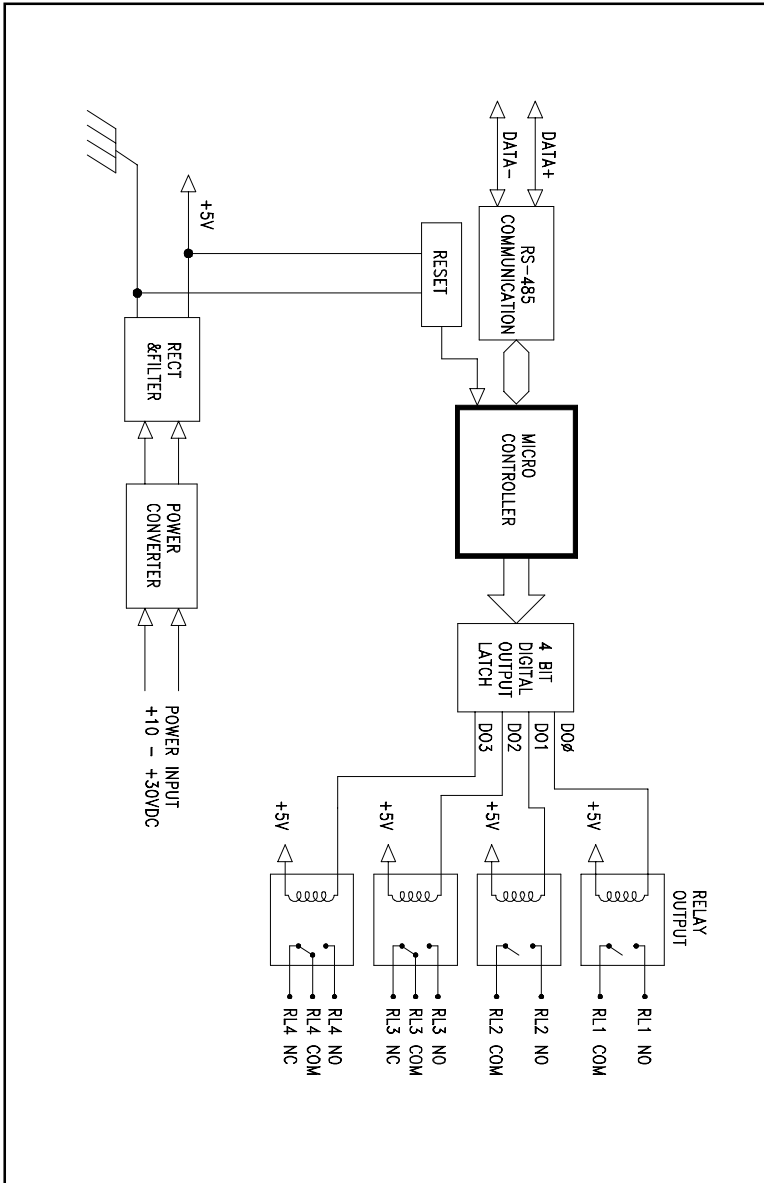


A.14 ADAM-4060 Relay Output Module

Table A-18 *ADAM-4060 Specifications*

| | |
|---------------------------------|--|
| Digital output | 4-channel relay, 2 form A, 2 form C |
| Input | RS-485 (2-wire) |
| speed (bps) | 1200, 2400, 4800, 9600, 19.2K, 38.4K |
| maximum distance | 4000 ft. (1200 m) |
| Contact rating | AC: 0.6 A/125 V; 0.3 A/250 V DC: 2 A/30 V; 0.6 A/ 110 V |
| Breakdown voltage | 500 V _{AC} (50/60 Hz) |
| Relay on time (typical) | 3 msec |
| Relay off time (typical) | 1 msec |
| Total switching time | 10 msec |
| Insulation resistance | 1000 M Ω minimum at 500 V _{DC} |
| Watchdog timer | Yes |
| Power supply | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 0.8 W |

Figure A-14 ADAM-4060 Function Diagram

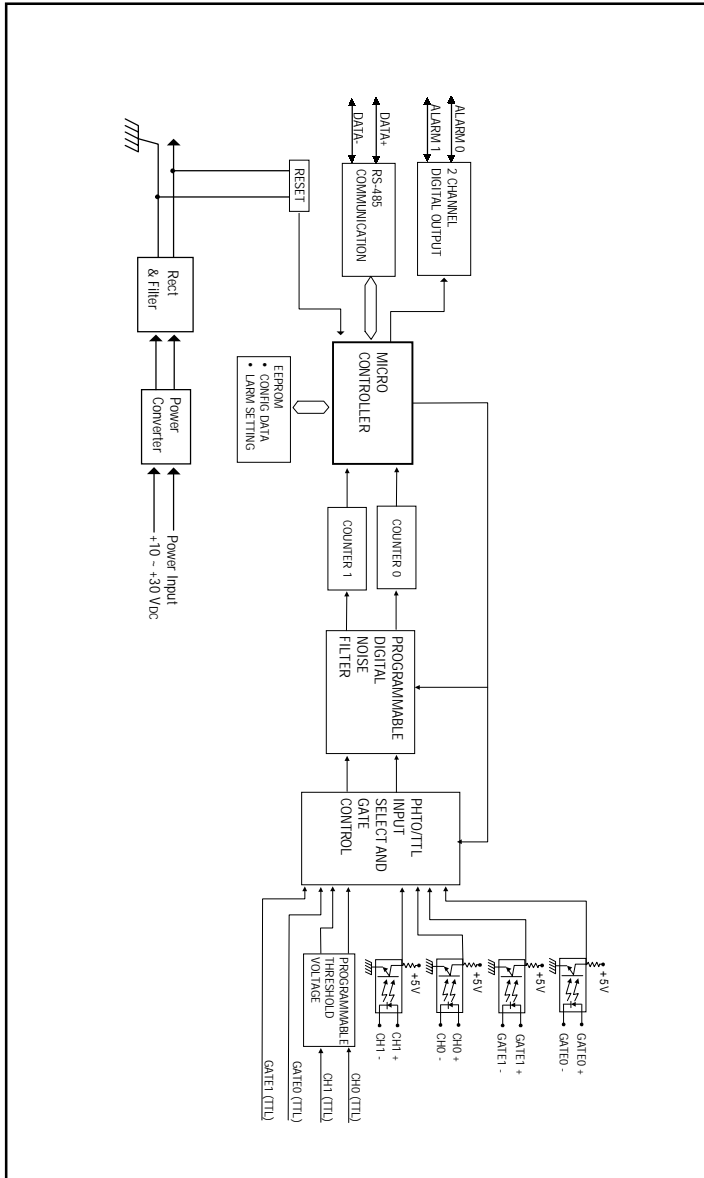


A.15 ADAM-4080 Counter/Frequency Input Module

Table A-19 ADAM-4080 Specifications

| | |
|--|--|
| Input channels | Two independent 32-bit counters |
| Input frequency | 50 kHz max. |
| Input mode | Isolated or non-isolated |
| Isolation input level | |
| Logic level 0 | +1 V max |
| Logical level 1 | +3.5 V to +30 V |
| Isolation voltage | 2500 V _{RMS} |
| Non-isolation input level | Programmable threshold |
| Logic level 0 | 0 to +5 V (default = 0.8 V) |
| Logic level 1 | 0 to +5 V (default = 2.4 V) |
| Input pulse width | >10 µsec |
| Maximum count | 4,294,967,295 (32 bits) |
| Programmable digital noise filter | 2 µsec to 65 msec |
| Alarming | Alarm comparators on each counter |
| Preset type | Absolute or relative |
| Frequency measurement | |
| Range | 5 Hz to 50 KHz |
| Programmable built-in gate time | 1.0/0.1 sec |
| Digital output channels | 2 channels open collector to 30 V |
| Sink current | 30 mA max. load |
| Power dissipation | 300 mW |
| Watchdog timer | Yes |
| Power requirements | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 2.0 W |

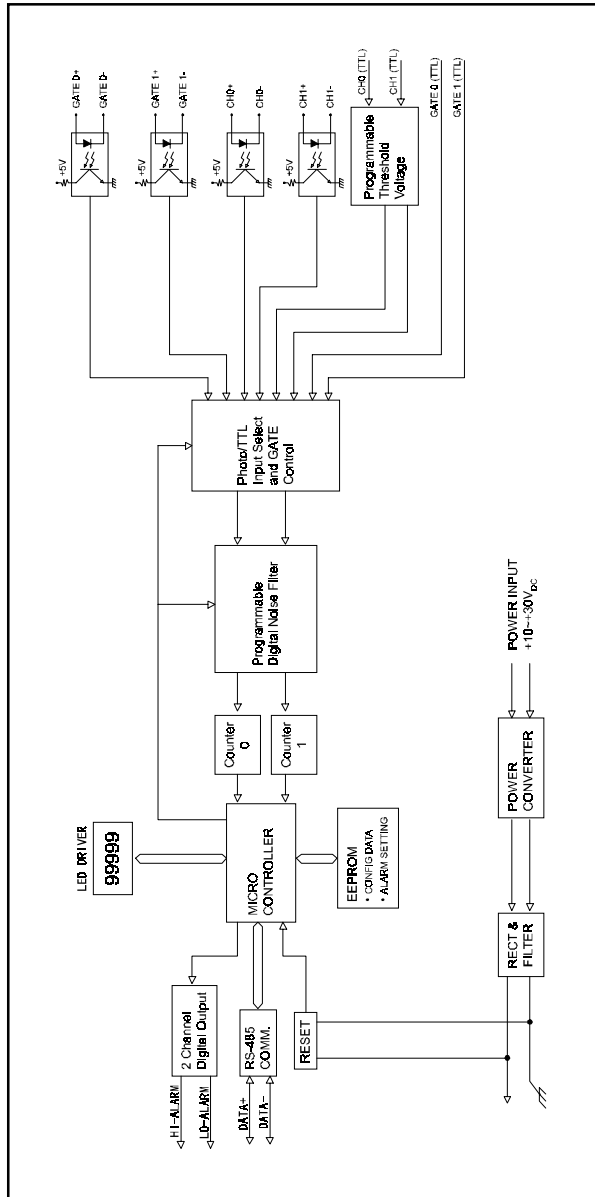
Figure A-15 ADAM-4080 Function Diagram



A.16 ADAM-4080D Counter/Frequency Input Module with LED Display**Table A-20** *ADAM-4080D Specifications*

| | |
|--|--|
| Input channels | Two independent 32-bit counters |
| Input frequency | 50 kHz max. |
| Input mode | Isolated or non-isolated |
| Isolation input level | |
| Logic level 0 | +1 V max |
| Logical level 1 | +3.5 V to +30 V |
| Isolation voltage | 2500 V _{RMS} |
| Non-isolation input level | Programmable threshold |
| Logic level 0 | 0 to +5 V (default = 0.8 V) |
| Logic level 1 | 0 to +5 V (default = 2.4 V) |
| Input pulse width | >10 μ sec |
| Maximum count | 4,294,967,295 (32 bits) |
| Programmable digital noise filter | 2 μ sec to 65 msec |
| Alarming | High and Low comparators on counter 1 |
| Frequency Measurement | |
| Range | 5 Hz to 50 KHz |
| Programmable built-in gate time | 1.0/0.1 sec |
| Display | LED indicator 5-digit readout; CH 0 or CH 1 (programmable) |
| Digital Output Channels | 2 channels Open collector to 30 V |
| Sink current | 30 mA max. load |
| Power dissipation | 300 mW |
| Watchdog timer | Yes |
| Power requirements | +10 to +30 V _{DC} (non-regulated) |
| Power consumption | 2.0 W |

Figure A-16 ADAM-4080D Function Diagram



Data Formats and I/O Ranges

B

B.1 Analog Input Formats

The ADAM analog input modules can be configured to transmit data to the host in one of the following data formats:

- Engineering Units
- Percent of FSR
- Twos complement hexadecimal
- Ohms

B.1.1 Engineering Units

Data can be represented in engineering units by assigning bits 0 and 1 of the data format/checksum/integration time parameter the value 00.

This format presents data in natural units such as degrees, volts, millivolts and milliamps. The engineering 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.

| Input Range | Resolution |
|------------------------------------|--|
| - 15 mV, - 50 mV | 1 μ V (three decimal places) |
| - 100 mV, - 150 mV, - 500 mV | 10 μ V (two decimal places) |
| - 1 V, - 2.5 V, - 5 V | 100 μ V (four decimal places) |
| - 10 V | 1 mV (three decimal places) |
| - 20 mA | 1 μ A (three decimal places) |
| Type J and T thermocouple | 0.01 $^{\circ}$ C (two decimal places) |
| Type K, E, R, S and B thermocouple | 0.01 $^{\circ}$ C (one decimal place) |

Data is grouped into 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:

Example 1

The input value is -2.65 and the corresponding analog input module is configured for a range of ± 5 V. The response to the Analog Data In command is:

-2.6500 (cr)

Example 2

The input value is 305.5°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 (cr)

Example 3

The input value is +5.653 V. The analog input module is configured for a ± 5 V range. When the engineering units format is used, the ADAM Series analog input modules are configured so that they automatically provide an overrange capability. The response to the Analog Data In command in this case is:

+5.6530 (cr)

B.1.2 Percent of FSR

This mode is used by setting bits 0 and 1 of the data format/checksum / integration time parameter to 01. The format used in Percent of FSR consists of a Plus (+) or minus (-) sign followed by five decimal digits including a decimal point. The maximum resolution possible is 0.01%. The decimal point is fixed.

Data is given as the ratio of the input signal to the value of the full-scale range.

Example 1

The input value is +2.0 V. The input module is configured for a range of ± 5 V. The response to the Analog Data In command is as follows:

+040.00 (cr)

The full calibrated voltage range ranges from -100% to 100% as voltage input ranges are always bipolar. A ± 5 V input would range from -5 V (-100%) to 5 V (100%).

In this example the input is represented by +40% of the full-scale range which equals $(+40/100) \times 5 \text{ V} = +2.0 \text{ V}$ the actual input value.

Example 2

The input value is 652.5°C. A type K thermocouple (0°C to 1000°C) is configured in the analog input module. The response to the Analog Data In command is:

+065.25 (cr)

The result shows that the value of the input (652.5°C) is 65.25% of the value of the calibrated full-scale range (1000°C).

Thermocouple input ranges are always assumed to be bipolar with zero being the point of symmetry. This holds true regardless of the specified range of operation. For example, when using a type J thermocouple (0°C to 760°C) 760°C corresponds to +100% and 0°C corresponds to 0%. Even if 0°C lies outside the specified range of operation for the thermocouple, zero will remain the point of symmetry. For instance, a type B thermocouple is specified for operation from +500°C to +1800°C. In this case +1800°C corresponds to +100% and 500°C corresponds to +27.77%.

The percentage is related to the full span of the configured range. If for instance a nickel RTD is specified for -80°C to +100°C then the lower value of -80°C equals 0% of span and the upper value of +100°C equals 100% of span.

When in the FSR mode, if a value exceeds the uppermost value of the input range, an overrange feature is automatically invoked by the ADAM analog input modules. Take, for instance, an analog module which is configured for a ± 5 V range but one of the values read is + 5.5V. The resulting value would then be 110%.

The readings must fall within the input range to be guaranteed of accuracy. Although they are typically linear readings which fall between the $\pm 100\%$ and $\pm 115\%$ limits are not accurate, but still generally linear. Readings beyond these limits are neither accurate nor linear.

B.1.3 Twos complement hexadecimal

Easily transferred to integer format the Twos Complement Hexadecimal format represents the data in ASCII hexadecimal form providing rapid communication, high resolution and easy conversion to computer-compatible integer format.

To indicate twos complement hexadecimal bits 0 and 1 of the data format/checksum/integration time parameter must be set to 10. This format displays data in the form of a 4-character hexadecimal string. This string represents a 16-bit twos complement binary value. Positive full scale is denoted as 7FFF (+32,767) while negative full scale is represented by the value 8000 (-32,768). The resolution is one least significant bit (LSB) of 16 bits.

Example

The input value is -1.234 V. An analog input module is configured for a ± 5 V range. The value returned is:

E069 (cr)

This value is equivalent to the signed integer -8087.

Input ranges with voltage and milliamp values are used with the full calibrated voltage range from 8000 to 7FFF. For instance, an ADAM-4011 module is given a ± 5 V input range. In this case -5 V is represented as 8000h and +5 V is denoted as 7FFFh.

When thermocouple input ranges are used, an input range which is bipolar

Data Formats and I/O Ranges

and symmetric about zero is assumed. The following table provides several examples.

| Thermocouple Type | Temperature Range (Degrees) | Temperature Range (Hex) |
|-------------------|-----------------------------|-------------------------|
| J | 0 C to 760 C | 0000h - 7FFFh |
| T | -100 C to 400 C | E000h - 7FFFh |
| R | 500 C to 1750 C | 2492h - 7FFFh |

The given span of the specified range is used to determine the RTD input ranges for twos complement hexadecimal values. As an example, if the nickel RTD is specified for the range -80°C to +100°C, the respective values in hexadecimal notation would be 8000h to 7FFFh.

B.1.4 Ohms

To indicate ohms, set bits 0 and 1 of the data format/checksum/integration time parameter to 11; this data format is only valid for ADAM-4013 analog input modules.

The ohms format allows you to read the value of the RTD resistance in ohms. It consists of a “+” or “-” sign followed by five decimal digits and a decimal point. The resolution (position of the decimal point) of Platinum-Nickel RTDs is 10 m Ω . (two decimal places).

For example, for a 100 Ω . platinum RTD specified for -100°C to +100°C, +100°C corresponds to +138.50 Ω . and -100°C corresponds to +060.60 Ω .

B.2 Analog Input Ranges

| Range Code (hex) | Input Range Description | Data Formats | +F.S. | Zero | -F.S. | Displayed Resolution |
|------------------|-------------------------|-------------------|---------|---------|---------|----------------------|
| 00 | – 15 mV | Engineering Unite | +15.000 | –00.000 | -15.000 | 1 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 01 | – 50 mV | Engineering Unite | +50.000 | –00.000 | -50.000 | 1 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 02 | – 100 mV | Engineering Unite | +100.00 | –000.00 | -100.00 | 10 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 03 | – 500 mV | Engineering Unite | +500.00 | –000.00 | -500.00 | 10 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 04 | – 1 V | Engineering Unite | +100.00 | –0.0000 | -1.0000 | 100.00 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 05 | – 2.5 V | Engineering Unite | +2.5000 | –0.0000 | -2.5000 | 100.00 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 06 | – 20 mV | Engineering Unite | +20.000 | –00.000 | -20.000 | 1 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 07 | not used | | | | | |
| | | | | | | |
| | | | | | | |

| Range Code (hex) | Input Range Description | Data Formats | +F.S. | Zero | -F.S. | Displayed Resolution |
|------------------|-------------------------|-------------------|---------|---------|---------|----------------------|
| 08 | – 10 mV | Engineering Unite | +10.000 | –00.000 | -10.000 | 1 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 09 | – 5 V | Engineering Unite | +5.0000 | –0.0000 | -5.0000 | 100.00 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 0A | – 1 V | Engineering Unite | +1.0000 | –0.0000 | -1.000 | 100.00 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 0B | – 500 mV | Engineering Unite | +500.00 | –000.00 | -500.00 | 10 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 0C | – 150 mV | Engineering Unite | +150.00 | –000.00 | -150.00 | 10 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |
| 0D | – 20 mV | Engineering Unite | +20.000 | –00.000 | -20.000 | 1 v |
| | | % of FSR | +100.00 | –000.00 | -100.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 8000 | 1 LSB ¹ |

| Range Code (hex) | Input Range Description | Data Formats | Maximum Specified Signal | Minimum Specified Signal | Displayed Resolution |
|------------------|-------------------------------------|-------------------|--------------------------|--------------------------|----------------------|
| 0E | Type J Thermocouple 0 C to 760 C | Engineering Unite | +760.000 | +000.00 | 0.01 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |

Data Formats and I/O Ranges

| Range Code (hex) | Input Range Description | Data Formats | Maximum Specified Signal | Minimum Specified Signal | Displayed Resolution |
|------------------|--|-------------------|--------------------------|--------------------------|----------------------|
| 0F | Type K Thermocouple 0 C to 1000 C | Engineering Unite | +1000.0 | +0000.0 | 0.1 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |
| 10 | Type T Thermocouple -100 C to 400 C | Engineering Unite | +400.00 | -100.00 | 0.01 C |
| | | % of FSR | +100.00 | -0.25.00 | 0.01% |
| | | Twos Complement | 7FFF | E000 | 1 LSB ¹ |
| 11 | Type E Thermocouple 0 C to 1000 C | Engineering Unite | +1000.0 | +0000.0 | 0.1 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |
| 12 | Type R Thermocouple 500 C to 1750 C | Engineering Unite | +1750.0 | +0500.0 | 0.1 C |
| | | % of FSR | +100.00 | +028.57 | 0.01% |
| | | Twos Complement | 7FFF | 2492 | 1 LSB ¹ |
| 13 | Type R Thermocouple 500 C to 1750 C | Engineering Unite | +1750.0 | +0500.0 | 0.1 C |
| | | % of FSR | +100.00 | +028.57 | 0.01% |
| | | Twos Complement | 7FFF | 2492 | 1 LSB ¹ |
| 14 | Type B Thermocouple 500 C to 1800 C | Engineering Unite | +1800.0 | +0500.0 | 0.1 C |
| | | % of FSR | +100.00 | +027.77 | 0.01% |
| | | Twos Complement | 7FFF | 2381 | 1 LSB ¹ |
| 20 | 100.00 W Platinum RTD a = .00385 -100 C to 100 C | Engineering Unite | +100.00 | -100.000 | 0.1 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 8000 | 1 LSB ¹ |
| | | Ohms | +138.50 | +060.60 | 10 mW |
| 21 | 100.00 W Platinum RTD a = .00385 0 C to 100 C | Engineering Unite | +100.000 | +000.00 | 0.1 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |
| | | Ohms | +138.50 | +100.00 | 10 mW |

| Range Code (hex) | Input Range Description | Data Formats | Maximum Specified Signal | Minimum Specified Signal | Displayed Resolution |
|------------------|--|-------------------|--------------------------|--------------------------|----------------------|
| 22 | 100.00 W Platinum RTD a = .00385 0 C to 200 C | Engineering Unite | +200.00 | +000.00 | 0.01 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |
| | | Ohms | +175.84 | +100.00 | 10 mW |
| 23 | 100.00 W Platinum RTD a = .00385 0 C to 600 C | Engineering Unite | +600.00 | +000.00 | 0.01 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |
| | | Ohms | +313.59 | +100.00 | 10 mW |
| 24 | 100.00 W Platinum RTD a = .00392 -100 C to 100 C | Engineering Unite | +00.00 | -100.00 | 0.01 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 8000 | 1 LSB ¹ |
| | | Ohms | +139.16 | +060.60 | 10 mW |
| 25 | 100.00 W Platinum RTD9285 0 C to 100 C | Engineering Unite | +100.00 | +000.00 | 0.01 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |
| | | Ohms | +139.16 | +100.00 | 10 mW |
| 26 | 100.00 W Platinum RTD a = .00392 0 C to 200 C | Engineering Unite | +200.00 | +000.00 | 0.01 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |
| | | Ohms | +177.13 | +100.00 | 10 mW |
| 27 | 100.00 W Platinum RTD a = .00392 0 C to 600 C | Engineering Unite | +600.00 | +000.00 | 0.01 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |
| | | Ohms | +317.28 | +100.00 | 10 mW |
| 28 | 120 W Nickel RTD -80 C to 100 C | Engineering Unite | +100.000 | -80.00 | 0.01 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 8000 | 1 LSB ¹ |
| | | Ohms | +200.64 | +066.60 | 10 mW |

Data Formats and I/O Ranges

| Range Code (hex) | Input Range Description | Data Formats | Maximum Specified Signal | Minimum Specified Signal | Displayed Resolution |
|------------------|--|-------------------|--------------------------|--------------------------|----------------------|
| 29 | 100.00 W Nickel RTD a = .00392 0 C to 100 C | Engineering Unite | +100.00 | +000.00 | 0.01 C |
| | | % of FSR | +100.00 | +000.00 | 0.01% |
| | | Twos Complement | 7FFF | 0000 | 1 LSB ¹ |
| | | Ohms | +200.64 | +120.00 | 10 mW |

NOTE: ¹ Resolution is one LSB of 16 bits

B.3 Analog Output Formats

You can configure ADAM analog output modules to receive data from the host in one of the following data formats:

- Engineering Units
- Percent of FSR
- Twos complement hexadecimal

Data for ADAM modules can be used in any one of the following data formats:

B.3.1 Engineering Units

This format is chosen by setting bits 0 and 1 of the data format/slew rate/checksum parameter to 00.

Data is presented in natural units such as milliamps. The Engineering Units format is readily parsed by most 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 μ A.

Example: An analog output module (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 #0104.762(cr).

B.3.2 Percent of Span

The Percent of Span format is employed by setting bits 0 and 1 of the data format/slew rate/checksum parameter to 01.

This format consists of a “+” or “-” sign, three decimal digits, a decimal point and two decimal digits. Data is presented as the value sent of the output signal relative to the span of the output (percent of span).

The maximum resolution is 0.2% of span (5 μ A for the 0 to 20 mA output range) and the desired output value is 10 mA the format of the Analog Data Out command is as follows:

#01+050.00(cr)

indicating that the output is at 50% of span. The decimal point is fixed.

B.3.3 Hexadecimal

This format is selected by setting bits 0 and 1 of the data format/slew rate/checksum parameter to 10.

ASCII's condensed hexadecimal representation of data allows and provides high resolution, quick communication and easy conversion to computer-compatible integer format. The format consists of a 3-character hexadecimal string representing a 12-bit binary value. The resolution for the 0 to 20 mA output range equals .025% of span, which is 5 μ A. The corresponding value for 000 is 0 mA; likewise, the value FFF corresponds to 20 mA.

B.4 Analog Output Ranges

| Range Code (hex) | Input Range Description | Data Formats | Maximum Specified Signal | Minimum Specified Signal | Output Resolution |
|------------------|-------------------------|--------------------|--------------------------|--------------------------|-------------------|
| 30 | 0 to 20 mA | Engineering Unite | 20.000 | 00.000 | 5 A |
| | | % of FSR | +100.00 | +000.00 | 5 A |
| | | Hexadecimal Binary | FFF | 000 | 5 A |
| 31 | 4 to 20 mA | Engineering Unite | 20.000 | 04.000 | 5 A |
| | | % of FSR | +100.00 | +000.00 | 5 A |
| | | Hexadecimal Binary | FFF | 000 | 5 A |
| 32 | 0 to 10 V | Engineering Unite | 20.000 | 00.000 | 2.442 mA |
| | | % of FSR | +100.00 | +000.00 | 2.442 mA |
| | | Hexadecimal Binary | FFF | 000 | 2.442 mA |

Technical Diagrams

C

C.1 ADAM Dimensions

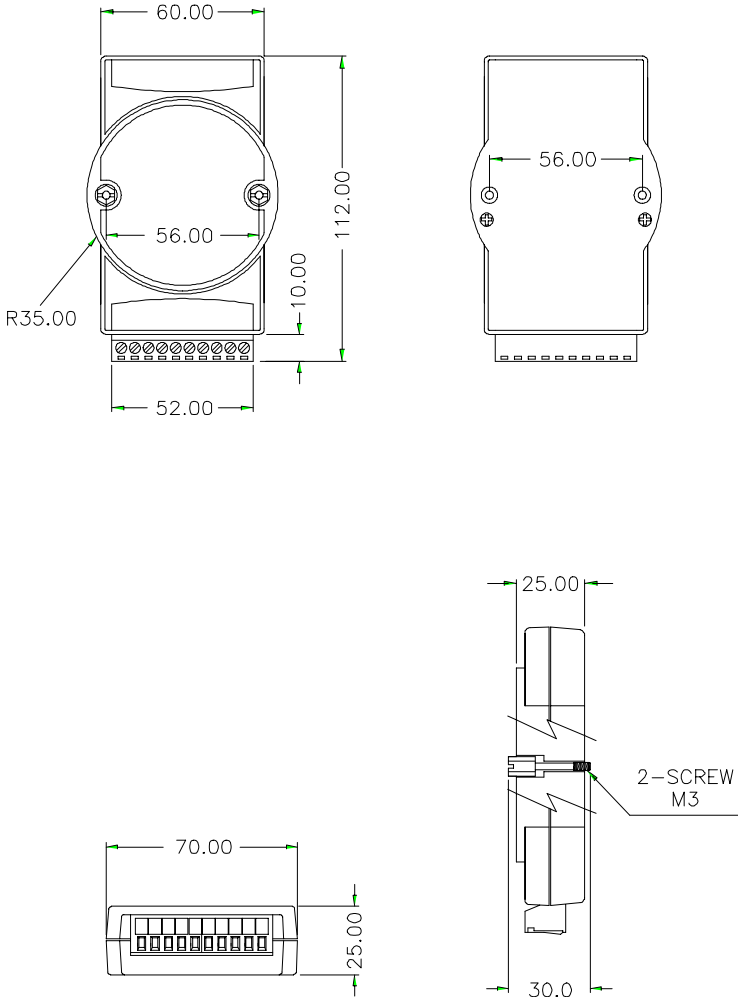


Figure C-1 ADAM Modules Dimensions

C.2 Installation

C.2.1 DIN-Rail Mounting

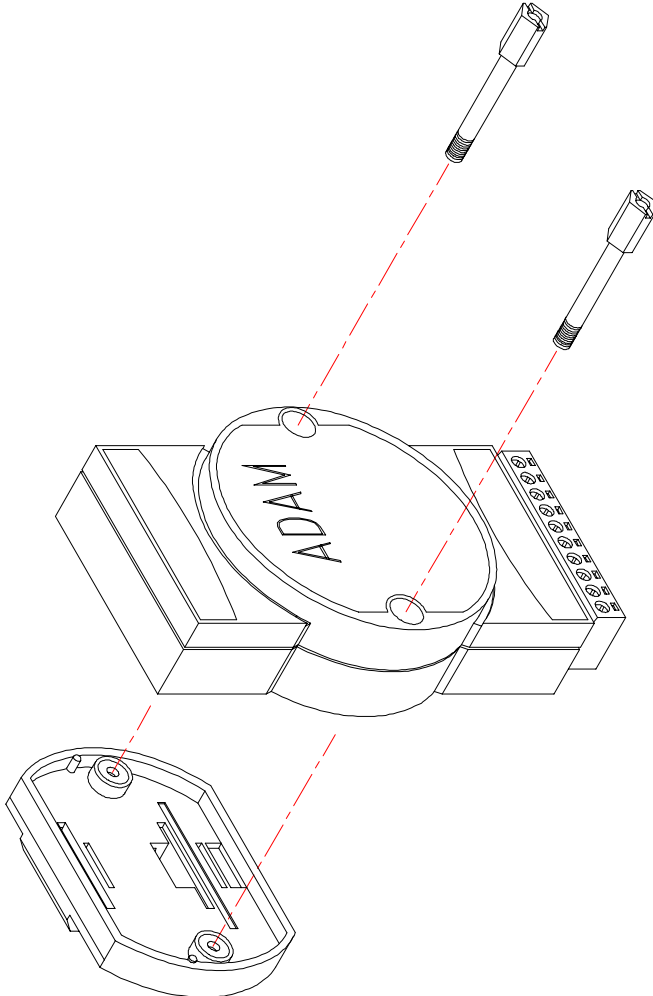


Figure C-2 *DIN-Rail Adapter*

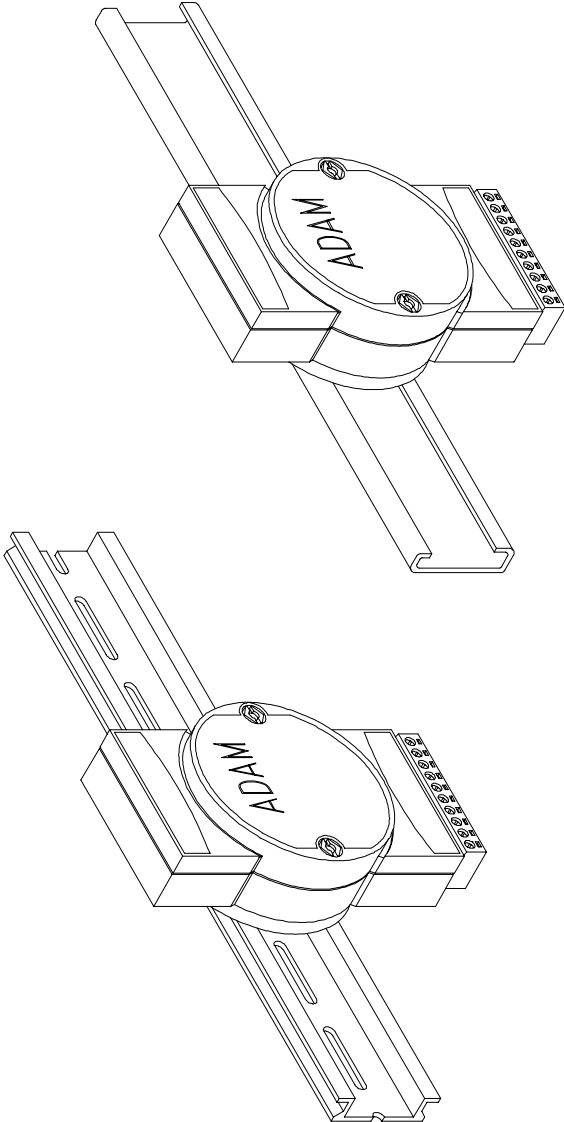


Figure C-3 *DIN-Rail Mounting*

C.2.2 Panel Mounting

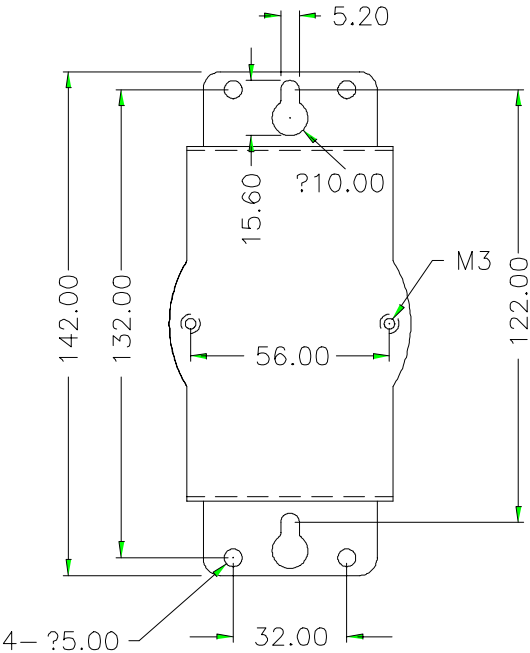


Figure C-4 Panel Mounting Bracket Dimensions

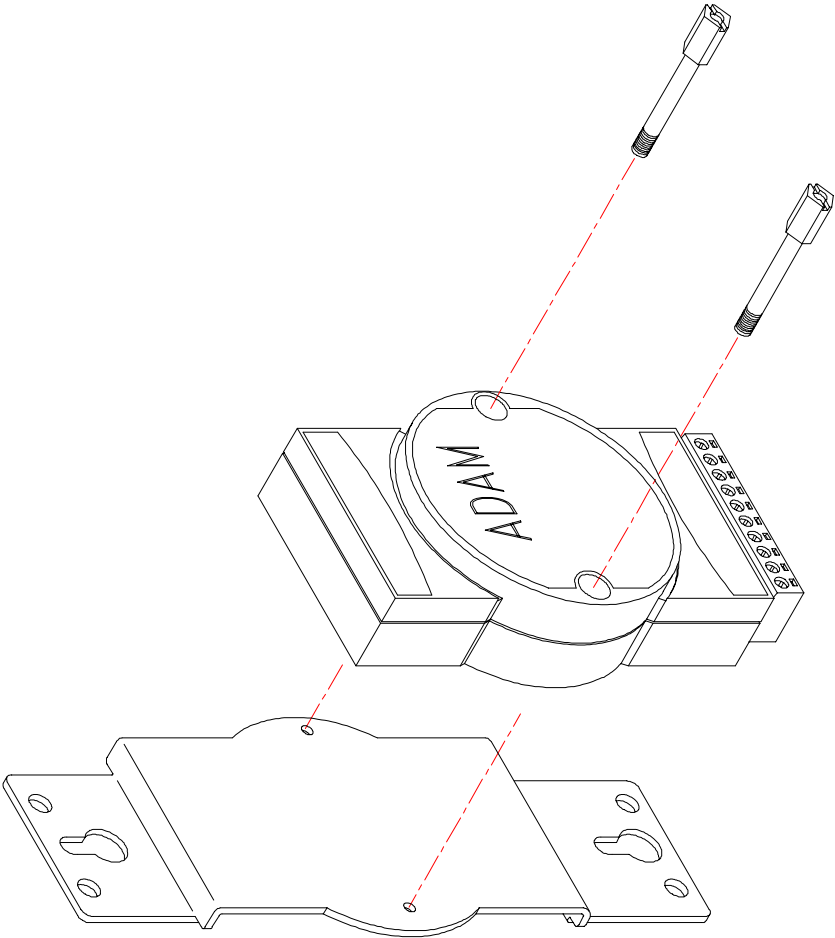


Figure C-5 Panel Mounting

C.2.3 Piggyback Stack

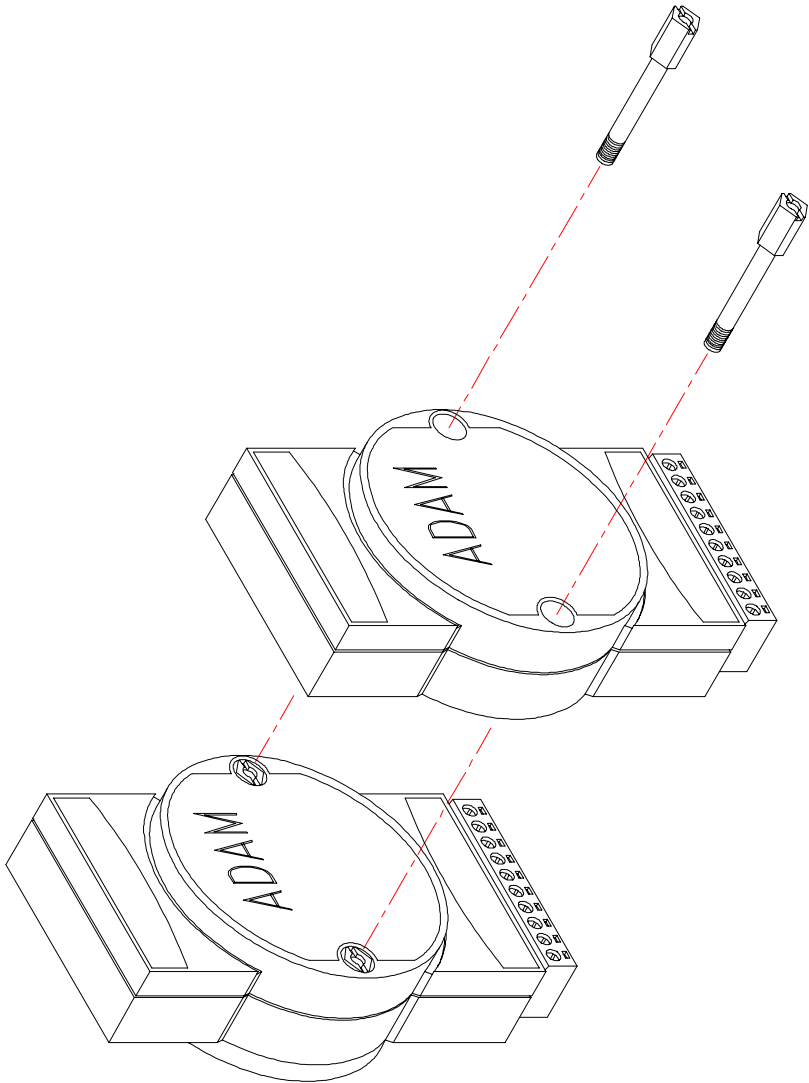


Figure C-6 Piggyback Stack

Utility Software

D

D.1 ADAM-4000 Utility Software

Together with the ADAM modules you will find a utility disk containing utility software with the following capabilities:

- Module configuration
- Module calibration
- Data Input and Output
- Alarm settings
- Autoscan of connected modules
- Terminal emulation

The following text will give you a brief instruction how to use the program.

Main menu

The main screen consists of a menu bar at the top side of the screen and a status field which displays information about the connected modules. When you first start the program, it will automatically scan for any attached modules and display their data. The status field lists module characteristics, module configuration parameters and in or output values.

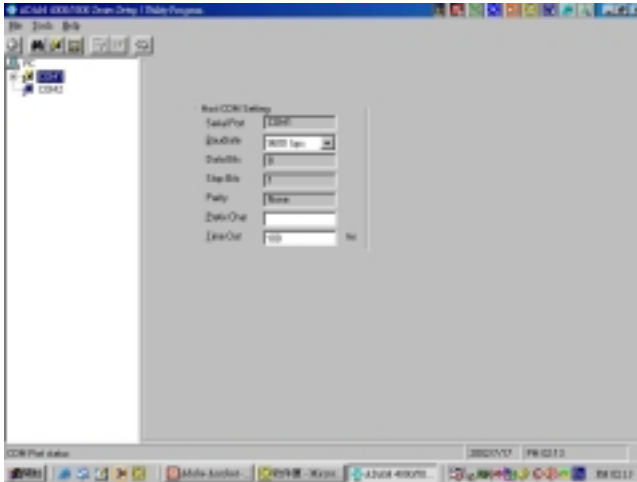


Figure D-1 Main screen

Normally you will use the Search command to scan the network. Highlight the Search command on the menu bar and press <Enter> (or simply press the “s” key). The “Search Installed Modules” window will then appear to prompt you to enter the range it should scan. Input a value 0 and 256 decimal.

NOTICE: When changing configuration, calibration or alarm parameters, you should always make sure that a window appears notifying you that the target module has confirmed the changes.

An asterisk sign “” before the modules address indicates that the module is in the INIT* state*

Setup

Select Setup from the top bar and a selection bar will appear in the status field. First, move the selection bar over the module you wish to configure and select it by pressing <Enter>. A configuration screen will appear with the settings available for its module type and the current values of its inputs. An example is shown in Figure D-2 for an ADAM-4011 module.



Figure D-2 Setup options

Here there are three different options, Configure, Edit-data and Alarm/Counter. To change the basic settings, select Configure and press <Enter>. (To return to the previous screen without making any changes, press <ESC>. This works for most screens in the program.)

Highlight the parameter you wish to change and press <Enter>. A window will appear with the configuration options for that parameter. Highlight the proper value and hit <Enter>. For some parameters, such as alarm high and low limit values, you will need to type in a specific value after selecting the parameter.

The Checksum and Baud rate options need special attention since they can only be changed when an ADAM module is in the INIT* state. To place a module in INIT state, its INIT terminal should be connected to its GND terminal (see Baud rate and Checksum in Chapter 2). When the ADAM module is not in INIT mode, an error message will appear. When it is in INIT mode, a window to change the Checksum or an option window showing you the valid baud rates will appear, depending on your choice.

After you have made the changes for a block of parameters, press <ESC>. You will be asked if you are satisfied with the changes you have made or not. Answer y to keep the changes you have made, n to escape without changing the values.

A similar procedure applies for alarm settings. Note that only the ADAM-4011 and ADAM-4012 analog input modules provide alarming and event counting.

If you wish to set the values of the module's outputs, select from the configuration screen menu and press <Enter>. Next highlight the output channel and toggle its value with the spacebar. Note that the digital outputs cannot be used when alarming functions are activated.

After you have made all necessary changes to the module configuration, you must issue the Run command to send the data to the Modules and let the changes take effect. If you select Quit instead of Run, the changes you have just made will not take effect and you will leave the program.

Calibration

Press <Enter> on the Calibrate option on the top bar and a selection bar appears in the status field. Move the selection bar over the module you wish to configure and select it by pressing <Enter>. Only analog input and output modules can be calibrated. If the module is an analog input module, you will be able to choose, for example, Zero Calibration. The screen will then look like Figure D-3.

To learn what steps should be taken to calibrate both input and output modules, refer to Chapter 5, Calibration.



Figure D-3 Zero Calibration

File

This option allows you to update the status field and can give you a hardcopy of all the connected modules that are shown on the screen. You can also print this information.

Terminal

This option allows you to directly send and receive commands on the RS-485 line. It has two options: Command Test and Terminal Emulation.

With Command Test you send commands one at a time by typing them into the top blank and pressing <Enter>. The response appears in the bottom blank. To send the command again, simply press <Enter> again.

Terminal Emulation is a full screen version of Command Test. (See Figure D-4, below.) Previous commands and their responses stay on the screen for you to refer to. If you want to repeatedly send a command, press <F10> and a dialogue box will appear into which you can enter the command. Press <Enter> to send the command. To stop the repeating command, press any key.

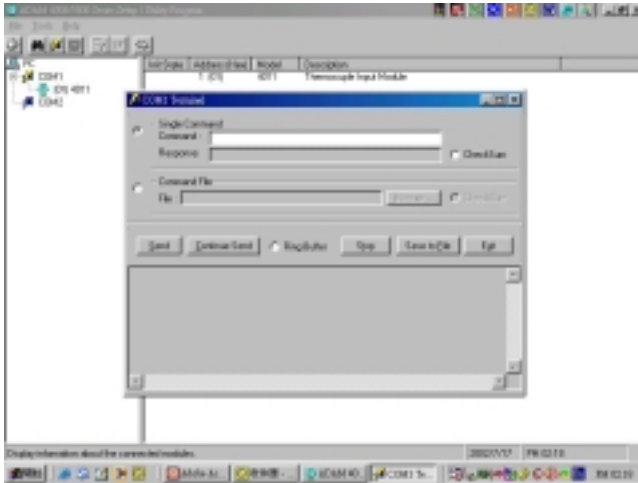


Figure D-4 Terminal Emulation

A box on the right hand side of the screen shows the communication parameters for the serial line such as the baud rate and number of stop bits.

Quit

Choosing the Quit option ends the ADAM utility program.

D.2 ADAM-4018M Utility Software

The ADAM-4018M is designed to add memory to the ADAM-4018. To simplify matters, it uses the same software utility as the ADAM-4018, but with a few additional memory function settings:

- Configuration setting
- Alarm setting
- Operation setting
- Data reading

The following text provides a brief introduction on software usage.

Configuration setting

The configuration setting allows you to set the sampling interval, standalone mode, data logger mode and channel enable/disable status. The sampling interval can be set from 2 to 65,535 seconds. To operate the ADAM-4018M in the field, you must power on the memory module by setting the standalone mode as "YES". Otherwise, the data will not be recorded.



There are three data logging modes.

1. **Standard Mode:** All eight channels serve as the standard logger to record normal data according to the sampling interval.
2. **Event Mode:** All eight channels serve as the event logger in cases when the data recording value either exceeds the High Alarm limit or goes below the Low Alarm limit.
3. **Mixed Mode:** Channels 0-3 serve as the standard logger, while channels 4-7 serve as the event logger.

NOTE For channel setting, "E" means Enable the channel to record data, "D" means to Disable.

Alarm setting

This screen allows users to set the high/low alarm limit. Its range is from -65,535 to +65,535.



Operation setting

This screen allows users to "START/STOP" the storage function of the memory module and to select the option to transfer ADAM-4018M memory data to the host.



Data reading

This screen allows users to read data stored in the memory, after the data has been transferred to the host.

```

MS-DOS Prompt
nDMS-4818 Memory Setup Program
Configuration setting Alarm setting Operation setting Data Reading

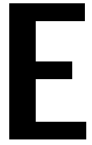
PAGE : 0
Record 0:CH=1,Data=19.6618,time_interval=1304231959 seconds
Record 1:CH=2,Data=19.8940,time_interval=1304231959 seconds
Record 2:CH=6,Data=270.2000,time_interval=1304231959 seconds
Record 3:CH=5,Data=187.1100,time_interval=1304231959 seconds
Record 4:CH=4,Data=0021.0000,time_interval=1304231959 seconds
Record 5:CH=9,Data=0.9599,time_interval=1304231959 seconds
Record 6:CH=6,Data=16.9.0000,time_interval=1304231959 seconds
Record 7:CH=2,Data=69.4100,time_interval=1304231959 seconds
Record 8:CH=1,Data=0.6024,time_interval=1304231959 seconds
Record 9:CH=1,Data=0.6167,time_interval=1304231959 seconds
Record 10:CH=6,Data=21.9250,time_interval=1304231959 seconds
Record 11:CH=5,Data=234.6000,time_interval=1304231959 seconds

Tue Feb 11 12:20:42 1996
Tue Feb 11 12:29:06 1996
<F5> show previous page <ESC> exit
<F6> show next page <TAB> key in page number
  
```

NOTE 1. When standalone mode (in Memory Configuration Settings) is set to "NO" and you want to read data from the data logger, you **MUST** set the operation mode to "STOP" before you read data.

NOTE 2. In standalone mode, if an LED light begins to blink once per second, 15 seconds after power is turned on, this means that the memory is not working. Turn the power off, then back on, and check the memory.

RS-485 Network



EIA RS-485 is industry's most widely used bidirectional, balanced transmission line standard. It is specifically developed for industrial multi-drop systems that should be able to transmit and receive data at high rates or over long distances.

The specifications of the EIA RS-485 protocol are as follows:

- Max line length per segment: 1200 meters (4000 feet)
- Throughput of 10 Mbaud and beyond
- Differential transmission (balanced lines) with high resistance against noise
- Maximum 32 nodes per segment
- Bi-directional master-slave communication over a single set of twisted pair cables
- Parallel connected nodes, true multi-drop

ADAM modules are fully isolated and use just a single set of twisted pair wires to send and receive! Since the nodes are connected in parallel they can be freely disconnected from the host without affecting the functioning of the remaining nodes. In industry shielded twisted pair is preferable due to the high noise ratio of the environment.

When nodes communicate through the network, no sending conflicts can occur since a simple command/response sequence is used. There is always one initiator (with no address) and many slaves (with address). In this case the master is a personal computer that is connected with its serial, RS-232, port to an ADAM RS-232/RS-485 converter. The slaves are the ADAM I/O modules. When modules are not transmitting data, they are in listen mode. The host computer initiates a command/response sequence with one of the modules. Commands normally contain the address of the module the host wants to communicate with. The module with the matching address carries out the command and sends its response to the host.

E.1 Basic Network Layout

Multi-drop RS-485 implies that there are two main wires in an segment. The connected modules tap from these two lines with so called drop cables. Thus all connections are parallel and connecting or disconnecting of a node doesn't affect the network as a whole. Since ADAM modules use the RS-485 standard, and use an ASCII-based commands set, they can connect and communicate with all ASCII-based computers and terminals. The basic layouts that can be used for an RS-485 network are:

Daisychain

The last module of a segment is a repeater. It is directly connected to the main-wires thereby ending the first segment and starting the next segment. Up to 32 addressable modules can be daisychained . This limitation is a physical one. When using more modules per segment the IC driver current rapidly decreases, causing communication errors. Totally the network can hold up to 256 addressable modules. The limitations for this number is the two number hexadecimal address code that knows 256 combinations. The ADAM converter, ADAM repeaters and the host computer are non addressable units and therefore are not included in these numbers.

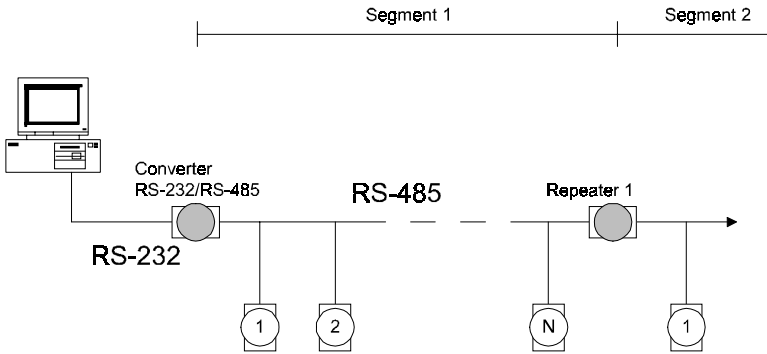


Figure E-1 *Daisy chaining*

Star Layout

In this scheme the repeaters are connected to drop-down cables from the main wires of the first segment. A tree structure is the result. This scheme is not recommended when using long lines since it will cause a serious amount of signal distortion due to a signal reflection in a several line endings.

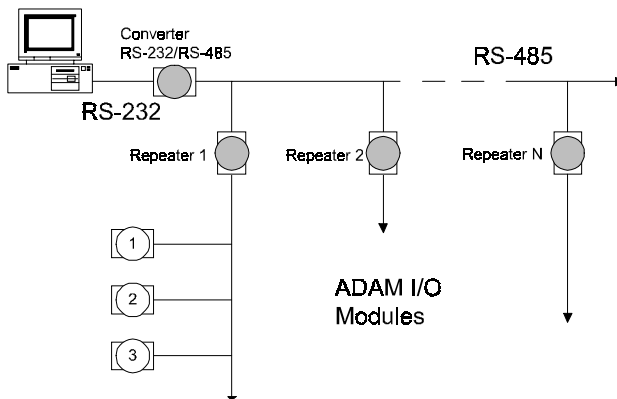


Figure E-2 *Star structure*

Random

This is a combination of daisychain and hierarchical structure

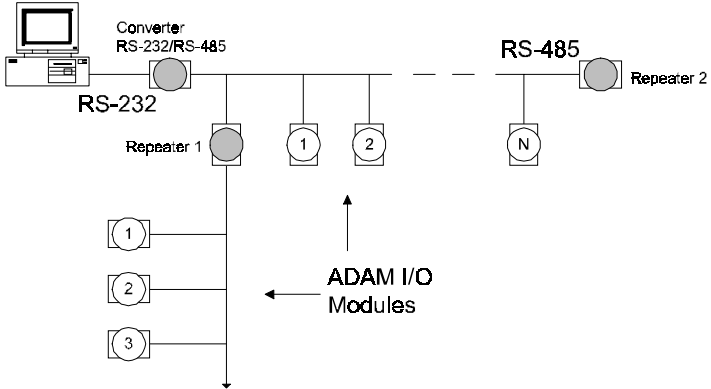


Figure E-3 *Random structure*

E. 2 Line Termination

Each discontinuity in impedance causes reflections and distortion.

When an impedance discontinuity occurs in the transmission line the immediate effect is signal reflection. This will lead to signal distortion. Specially at line ends this mismatch causes problems. To eliminate this discontinuity terminate the line with a resistor.

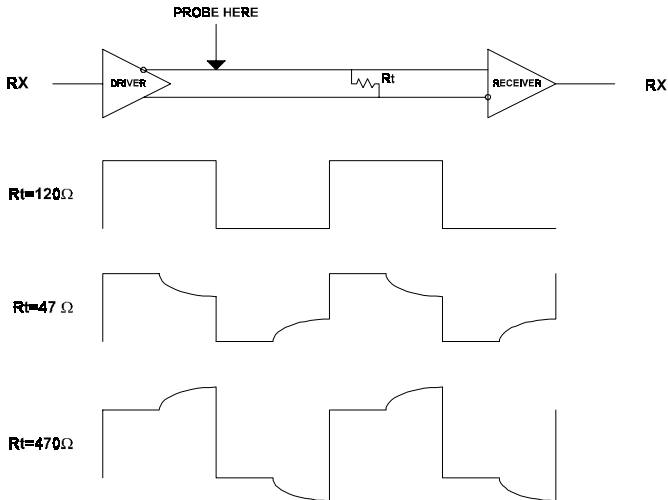


Figure E-4 *Signal Distortion*

The value of the resistor should be as close as possible to the characteristic impedance of the line. Although receiver devices add some resistance to the whole of the transmission line, normally it is sufficient to the resistor impedance should equal the characteristic impedance of the line.

Example:

Each input of the receivers has a nominal input impedance of 18 k Ω feeding into a diode transistor-resistor biasing network that is equivalent to an 18 k Ω input resistor tied to a common mode voltage of 2.4 V. It is this configuration which provides the large common range of the receiver required for RS-485 systems! (See Figure E-5 below).

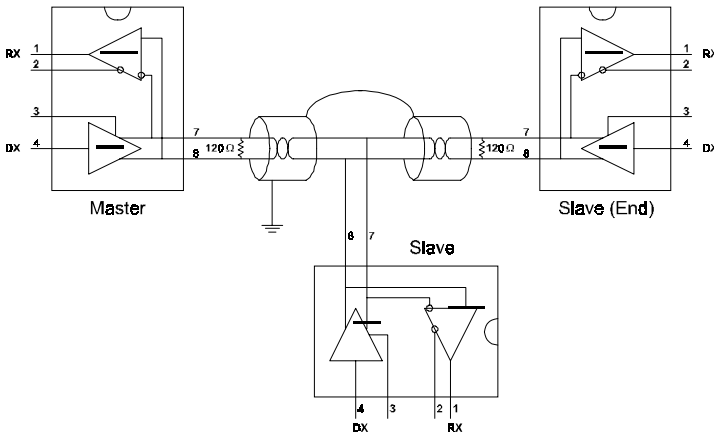


Figure E-5 Termination resistor locations

Because each input is biased to 2.4 V, the nominal common mode voltage of balanced RS-485 systems, the 18 k Ω on the input can be taken as being in series across the input of each individual receiver.

If thirty of these receivers are put closely together at the end of the transmission line, they will tend to react as thirty 36k Ω resistors in parallel with the termination resistor. The overall effective resistance will need to be close to the characteristics of the line.

The effective parallel receiver resistance R_P will therefore be equal to:

$$R_p = 36 \times 10^3 / 30 = 1200 \text{ } \Omega$$

While the termination resistor R_T will equal:

$$R_T = R_O / [1 - R_O/R_p]$$

Thus for a line with a characteristic impedance of $100 \text{ } \Omega$ resistor, the termination resistor R_T should be:

$$R_T = [1 - 100/1200] = 110 \text{ } \Omega$$

Since this value lies within 10% of the line characteristic impedance. Thus as already stated above the line termination resistor R_T will normally equal the characteristic impedance Z_O .

The star connection causes a multitude of these discontinuities since there are several transmission lines and is therefore not recommended.

NOTICE: *The recommended wiring method that causes a minimum amount of reflection is daisy chaining where all receivers tap from one transmission line and needs to be terminated only twice.*

E.3 RS-485 Data Flow Control

The RS-485 standard uses a single pair of wires to send and receive data. This line sharing requires some method to control the direction of the data flow. RTS (Request To Sent) and CTS (Clear To Sent) are the most commonly used method.

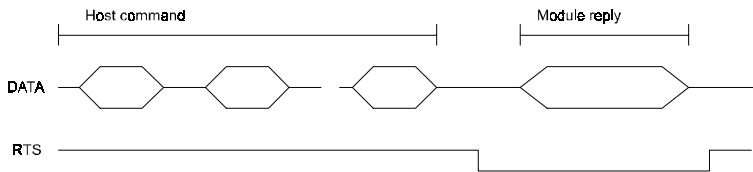


Figure E-6 *RS-485 data flow control with RTS*

Intelligent RS-485 Control

ADAM-4510 and ADAM-4520 are both equipped with an I/O circuit which can automatically sense the direction of the data flow. No handshaking with the host (like RTS, Request to Send) is necessary to receive data and forward it in the correct direction. You can use any software written for half-duplex RS-232 with an ADAM network without modification. The RS-485 control is completely transparent to the user.

How to use the Checksum feature



A checksum helps you to detect errors in commands from the host to the modules, and in responses from the modules to the host. The feature adds two extra checksum characters to the command or response string, which does reduce the throughput.

F.1 Checksum Enable/Disable

To enable configuration of a module's checksum feature, its INIT* terminal should be shorted to its GND terminal, after which the module should be rebooted. The checksum feature is enabled by setting bit 6 of the data format/checksum parameter to 1. To disable the checksum, set the parameter to 0. Remember that when using the checksum feature, it should always be enabled for all connected devices including the host computer.

The checksum is represented by a 2-character ASCII hexadecimal format and is transmitted just prior to the carriage return. The checksum equals the modulo-256 (100h) sum of all the ASCII values in the command preceding the checksum. If the checksum in a command is missing or incorrect the module will not respond.

Example 1

The following is an example of an Analog Data In command and response when the checksum is enabled:

Command: #0588(CR)

Response: +3.56719D(CR)

The input value at the module at address 05h is +3.5671 V. (The date format is engineering units.) The command checksum (88h) is the sum of the ASCII values of the following characters: #, 0, and 5. The response checksum (9Dh) is the sum of the ASCII values of the following characters: ">" "+" "3" "." "5" "6" "7" and "1".

Example 2

This example explains how to calculate the checksum value of a Read High alarm limit command string:

Case 1. (If the Checksum feature is **disabled**)

Command: \$07RH(cr)

Response: !07+2.0500(cr) when the command is valid.

Case 2. (If the Checksum feature is **enabled**)

Command: \$07RH25(cr)

Response: !07+2.0500D8(cr)

where:

25 represents the checksum of this command, and

D8 represents the checksum of the response.

The checksum of the command string is derived as shown below:

$$25h = (24h + 30h + 37h + 52h + 48h) \text{ MOD } 100h$$

The hexadecimal ASCII codes for \$, 0, 7, R, H are 24h, 30h, 37h, 52h and 48h respectively. The sum of these ASCII codes is 125h. The module-256(100h) sum of 125h is 25h.

How to use the Checksum feature

Table F-1 *Printable ASCII Characters*

| HEX | ASCII |
|-----|-------|
| 21 | ! |
| 22 | " |
| 23 | # |
| 24 | \$ |
| 25 | % |
| 26 | & |
| 27 | ' |
| 28 | (|
| 29 |) |
| 2A | * |
| 2B | + |
| 2C | , |
| 2D | - |
| 2E | . |
| 2F | / |
| 30 | 0 |
| 31 | 1 |
| 32 | 2 |
| 33 | 3 |
| 34 | 4 |
| 35 | 5 |
| 36 | 6 |
| 37 | 7 |
| 38 | 8 |
| 39 | 9 |
| 3A | : |
| 3B | ; |
| 3C | < |
| 3D | = |
| 3E | > |
| 3F | ? |

| HEX | ASCII |
|-----|-------|
| 40 | @ |
| 41 | A |
| 42 | B |
| 43 | C |
| 44 | D |
| 45 | E |
| 46 | F |
| 47 | G |
| 48 | H |
| 49 | I |
| 4A | J |
| 4B | K |
| 4C | L |
| 4D | M |
| 4E | N |
| 4F | O |
| 50 | P |
| 51 | Q |
| 52 | R |
| 53 | S |
| 54 | T |
| 55 | U |
| 56 | V |
| 57 | W |
| 58 | X |
| 59 | Y |
| 5A | Z |
| 5B | [|
| 5C | \ |
| 5D |] |
| 5E | ^ |
| 5F | _ |

| HEX | ASCII |
|-----|-------|
| 60 | ' |
| 61 | a |
| 62 | b |
| 63 | c |
| 64 | d |
| 65 | e |
| 66 | f |
| 67 | g |
| 68 | h |
| 69 | i |
| 6A | j |
| 6B | k |
| 6C | l |
| 6D | m |
| 6E | n |
| 6F | o |
| 70 | p |
| 71 | q |
| 72 | r |
| 73 | s |
| 74 | t |
| 75 | u |
| 76 | v |
| 77 | w |
| 78 | x |
| 79 | y |
| 7A | z |
| 7B | { |
| 7C | |
| 7D | } |
| 7E | ~ |