ADAM 4100

Industrial Grade Data Acquisition Modules
User's Manual

Copyright Notice

CE Notification

The ADAM-4100 series developed by Advantech Co., Ltd. has passed the CE test for environmental specifications. 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.

FCC Class A

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

Edition 1.9
May 2011

Table of Contents

1 Introduction

Overview

The ADAM-4100 series modules are compact, versatile sensor-to-computer interface units designed specifically for reliable operation in harsh environments. Their built-in microprocessors, encased in rugged industrial grade plastic, independently provide intelligent signal conditioning, analog I/O, digital I/O, LED data display and located mode to show their address. Especially, the located mode is friendly design for use to read module address directly.

Modular Industrial Design

The ADAM-4100 series are designed to endure the more adverse circumstances and to hold the more robust design. User can make use of them under special circumstances to fit more widespread application.

Ready for the Industrial Environment

- **Environment Monitoring Low Operating Temperature**
  ADAM-4100 series can support broad Operating Temperature range from -40^0C to $+85^0$C

- **High Noise Immunity**
  In order to get over the noise resulted from special environmental, the ADAM-4100 provide the more protection to counteract these effects like as **1KV Surge Input, 3KV EFT and 8KV ESD Protection**

- **Broad power input range**
  The ADAM-4100 series modules accept any unregulated power source between +10 and +48 V_{DC}. They are not only widened the input range but also protected from accidental power supply reversals and can be safely connected or disconnected without disturbing a running network.

- **The new features for individual I/O modules**
  Support 200Vdc Hi Common Mode Voltage (ADAM-4117)
  Support Uni-polar and Bi-polar input (ADAM-4117)
  Support +/- 15V Input Range (ADAM-4117)
  Support Filter Auto-tuning or Filter-out 50Hz/60Hz (ADAM-4117/4118)
  Digital Filter Function (ADAM-4150)
  DI channels allow to be used as 3 KHz counter (ADAM-4150)
  DO channels support pulse output function (ADAM-4150 / 4168)

ADAM-4100 Module with LED Display

ADAM-4100 series own a series of LED display on the face. They let you monitor the status and also allow reading the address of ADAM-
4100. Actually, they have original two operating mode (Initial mode and Normal mode). They also own a new mode "address mode". It is friendly interface to read the modular address directly through these LEDs.

Firmware online update

ADAM-4100 series have a friendly and convenient design for user to update firmware online. It can save a lot of time and money to update firmware procedure.

Dual Watchdog Timer Inside

A watchdog timer supervisory function will automatically reset the ADAM-4100 series modules if required, which reduces the need for maintenance. It includes the system and communication watchdog.

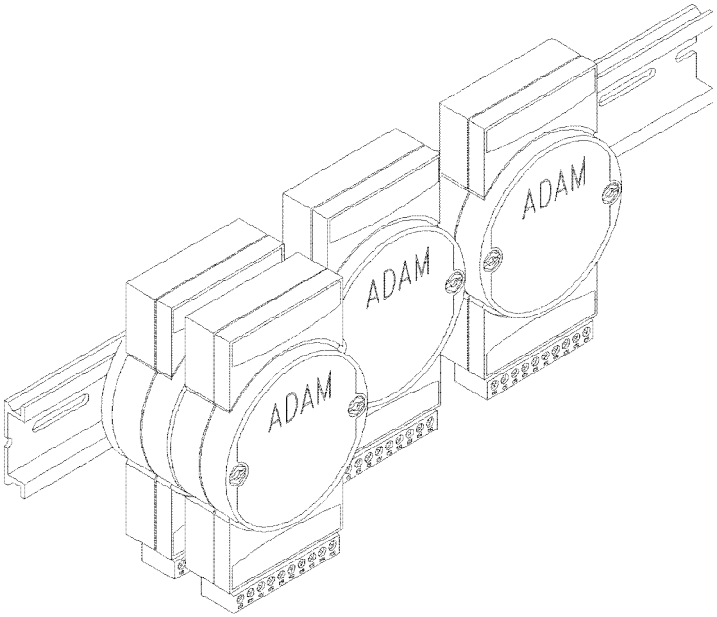
Dual Communication Protocol Support

To satisfy both current ADAM users and Modbus users, these ADAM-4100 Modules support both the ADAM protocol and Modbus/RTU protocol. You can select the communication mode you want through the Windows Utility Software. If users apply the ADAM protocol, the ASCII command/response will remain the same as usual. In RTU mode, data is sent as two four-bit, hexadecimal characters, providing for higher throughput than in ASCII mode for the same baud rate. The ADAM-4100 Series is a complete I/O solution, featuring Modbus Network Support, with a robust and intelligent design. It is the easiest to use, and a cost-effective choice for your system I/O needs.

RS-485

The ADAM-4100 series of modules use the EIA RS-485 communication protocol, the industry's most widely used bi-directional, balanced transmission line standard. The EIA RS-485 was specifically developed for industrial applications.

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.

2

Installation Guideline

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

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

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 a RS-232C or RS-485 port.
- Power supply for the ADAM modules (+10 to +48 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} or +48 V_{DC}, unregulated power. Operation is guaranteed when using any power supply between +10 and +48 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 +48 V_{DC}. All power supply specifications are referenced at module

connector. When modules are powered remotely, the effects of DC voltage drops must be considered.

All modules use on-board switching regulators to sustain good efficiency over the 10 to 48 V input range; therefore, we can assume that the actual drawn current is inversely proportional to the DC voltage.

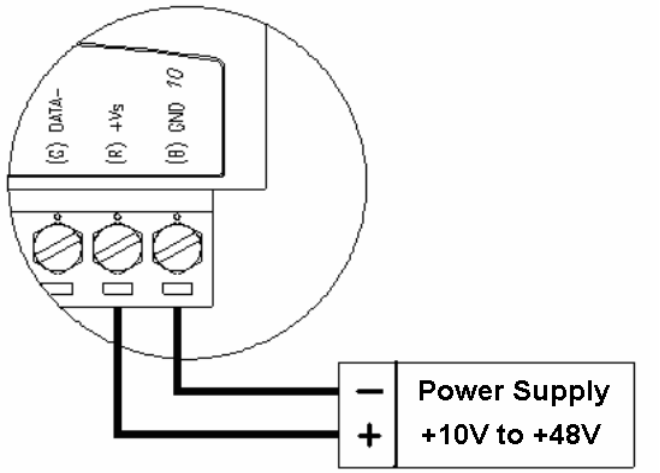


Figure 2-1 *Power Supply Connections*

We advise the following standard colors (as indicated on the modules) for each power line:

+Vs	(R)	Red
GND	(B)	Black

Communication Wiring

We recommend the use of shielded-twisted-pair cable in the ADAM network for reducing interference purpose, but the cable has to comply with the EIA RS-485 standard. Furthermore, only one set of twisted-pair cable is required for transmitting Data. We advise the following standard colors (as indicated on the modules) for each the communication line:

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 communicate through the ADAM command set. (See Appendix A, Utility Software)

ADAM Communication Speed

In ADAM series, the baud rate can be configured from 1200 bps to 115.2 Kbps. However, the baud rate 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 only has a RS-232 port, an ADAM Isolated RS-232/RS-485 Converter is required. Since this module is not addressable by the host, the baud rate must be reset 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 more than 32 ADAM modules are connected, a repeater should be implemented. In a network, up to eight Repeater modules can be connected allowing connection up to 255 ADAM modules. As with the Converter module, the Repeater module is not addressable by the host and the baud rate must be reset 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 if the baud rate is set correctly beforehand.

Default Factory Settings

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

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

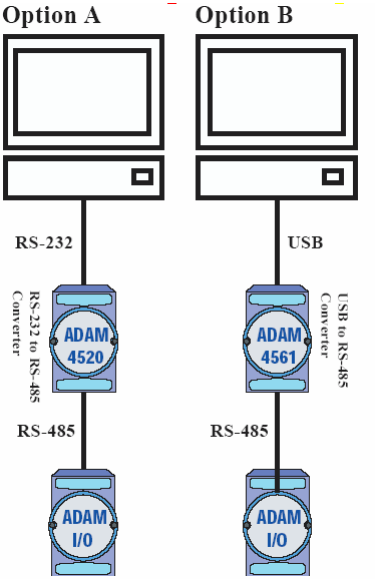


Figure 2-2 ***Basic Hook-up of ADAM Module to Host Switches***

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.

Grounding Protection

Grounding is one of the most important issues for our system. Just like Frame Ground of the computer, this signal offers a reference point of the electronic circuit inside the computer. If we want to communicate with this computer, both Signal Ground and Frame Ground should be

connected to make a reference point of each other's electronic circuit. Generally speaking, it is necessary to install an individual grounding bar for each system, such as computer networks, power systems, telecommunication networks, etc. Those individual grounding bars not only provide the individual reference point, but also make the earth a ground.

ADAM-4100 series allow choosing the ground setting. User can use right side ground scrap to connect. Following picture is illustration.

Configuration with the ADAM Utility Software

The easiest way to configure the ADAM module is by using the ADAM utility software. It is a user friendly structured menu program that will guide you through every step of the configuration. (See Appendix A, Utility Software)

Changing the protocol from ADAM ASCII to Modbus

Some ADAM-4100 modules support both ADAM ASCII and Modbus protocols, and the factory default setting of these modules is ADAM ASCII protocol. If you would like to configure the modules to Modbus protocol, please refer to Appendix G which describes how to change the protocol in ADAM utility.

Configuration with the ADAM command set

ADAM modules can also be configured by issuing direct commands through a terminal emulation program that is part of the ADAM utility software. Please refer to Chapter 4 to know more details.

2.3 Baud rate and Checksum

Adam modules contain EEPROMs to store configuration information and calibration constants. The EEPROM replaces the conventional array of switches and pots that are originally used for specifying baud rate, input and output range… etc.

Since there is no visual indication of a module's configuration status, it is impossible to know the baud rate, address and other settings just by looking at it. 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*. Booting the module while connecting the INIT* switch forces the configuration into a known state called the INIT* state.

INIT* state defaults:
Baud rate: 9600
Address: 00h
Checksum: disabled

Forcing the module in INIT* state does not change any parameters in the module's EEPROM. When the module is in the INIT* state, 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 settings 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 turning the switch to “initial” as following (See Figure 2-3).

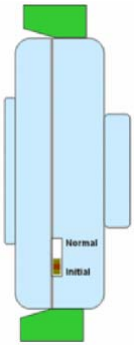


Figure 2-3 Initial mode

- 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 OFF to the ADAM Module.
- Turn the switch to “Normal” 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 is an example of how ADAM modules are connected in a multiple module network:

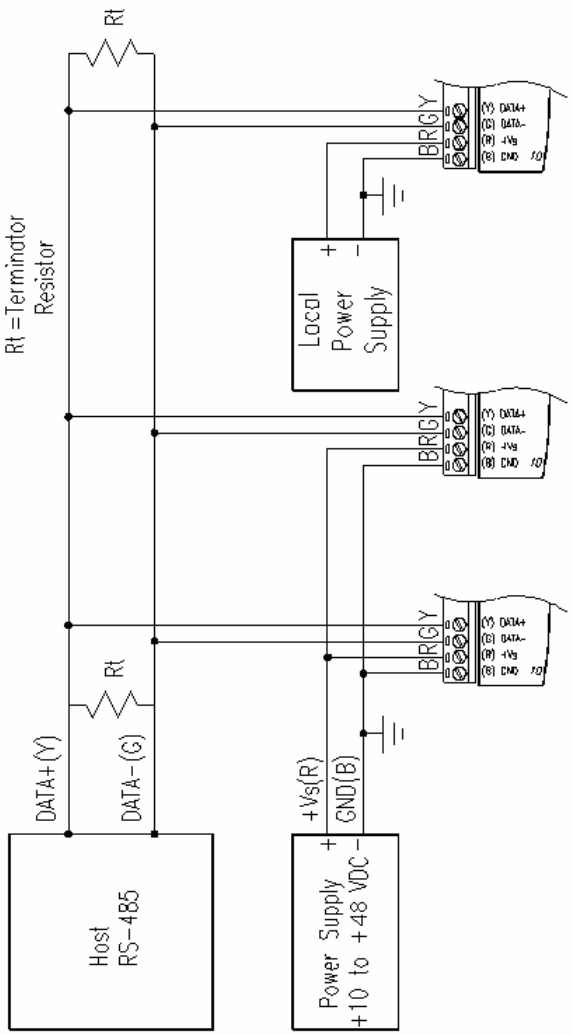
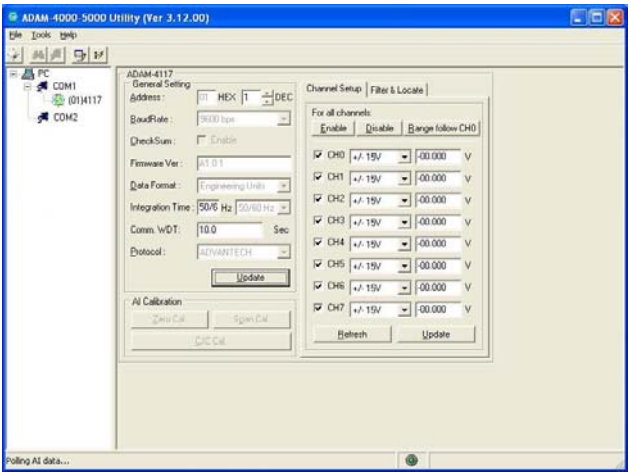


Figure 2-4 ***Multi-module Connection***

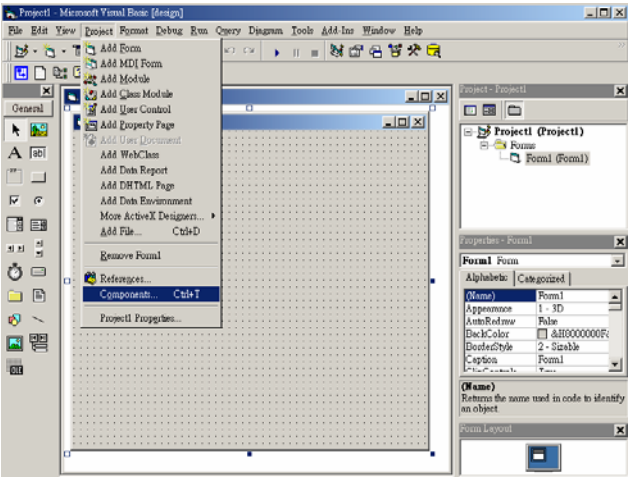
2.5 Programming Example

The following example is a simple program written in Visual Basic 6.0 that demonstrates how to get temperature reading which is stored in the address of 01H from ADAM-4117 module.

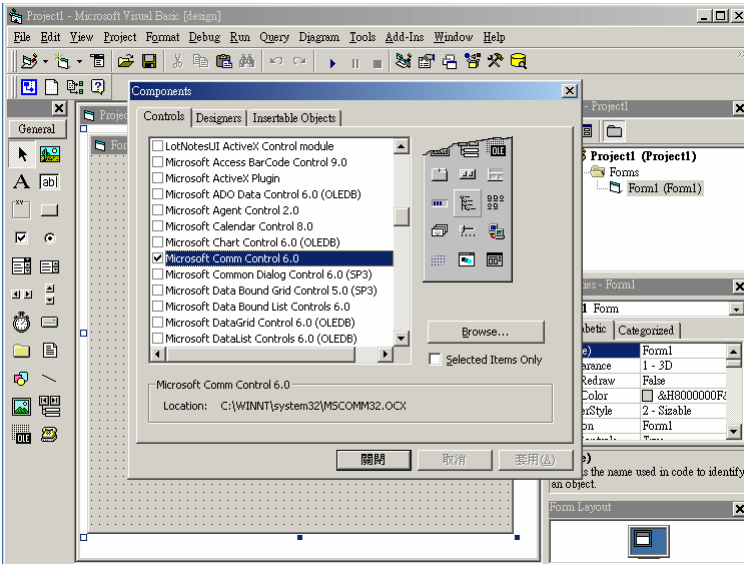
Step 1. Using ADAM Utility to check the settings as the following below:
"Address = 01H", "Baud rate = 9600" and "Checksum = Disabled".



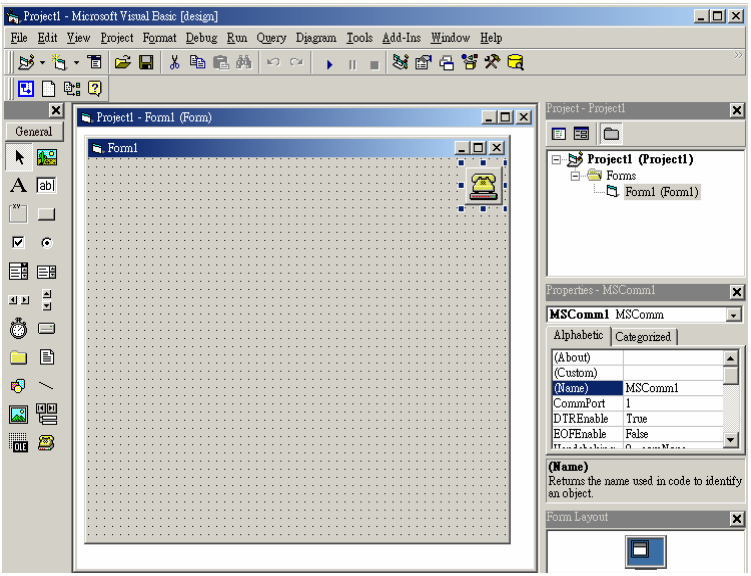
Step 2. Run VB 6.0 and add a control via "Project\Component".



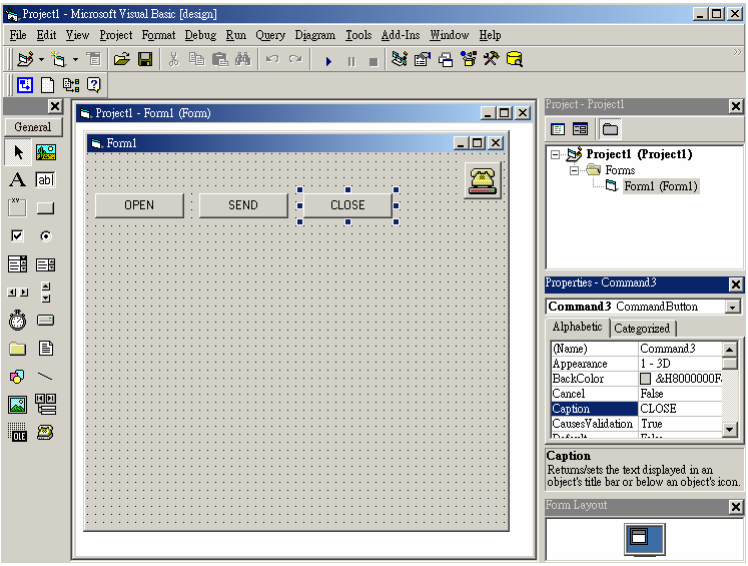
Step 3. Select “Microsoft Comm Control”



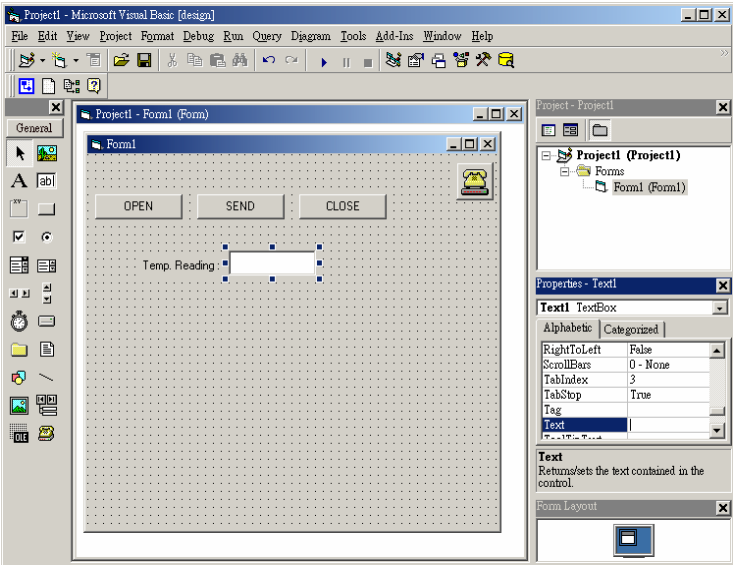
Step 4. Add the Comm Control on the form.



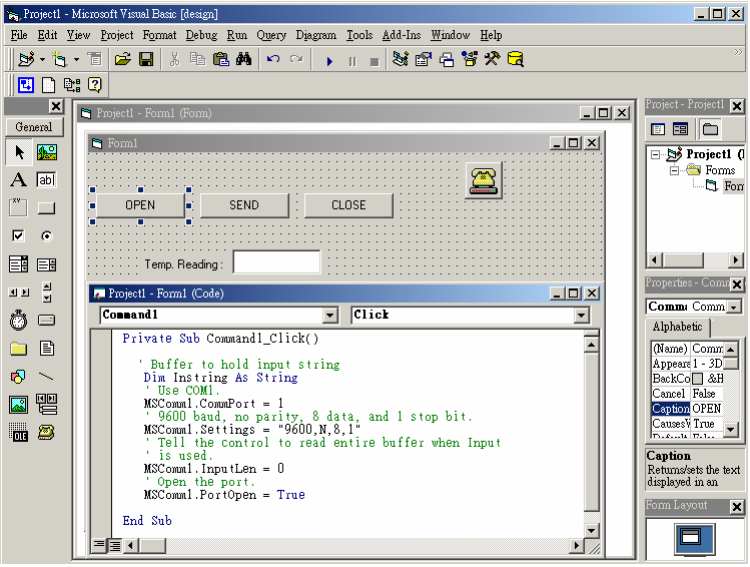
Step 5. Add three Command Buttons on the form as shown below



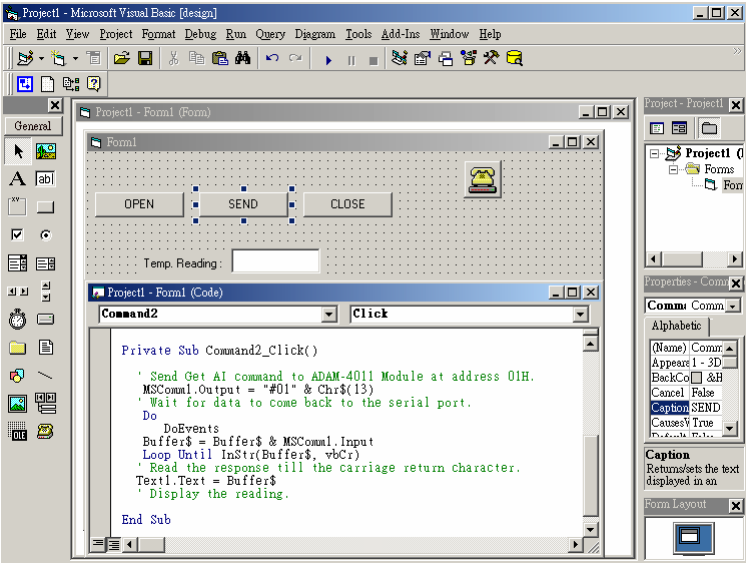
Step 6. Add one Label and one Text on the form as shown below.



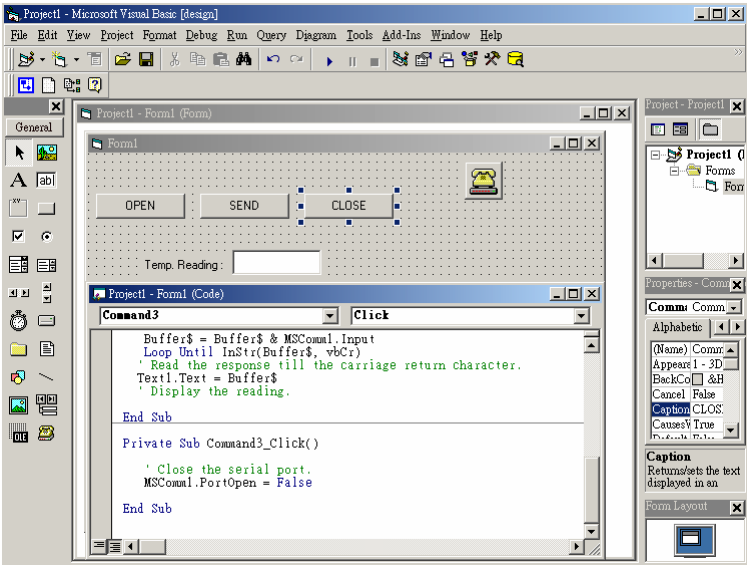
Step 7. Click OPEN Button and type in the following codes. The source codes are listed at the end of this section.



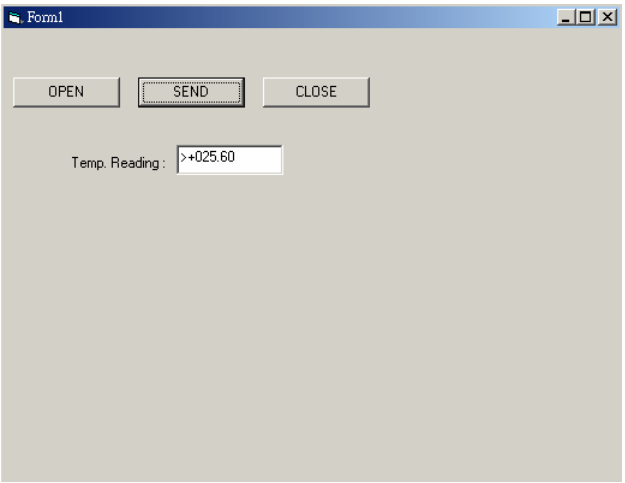
Step 8. Click SEND Button and type in the following codes. The source codes are listed at the end of this section.



Step 9. Click CLOSE Button and type in the following codes. The source codes are listed at the end of this section.



Step 10. Run the Project → Click OPEN to open COM1 → Click SEND to send the Get Temperature Reading Command. Now, you will find the reading the same as the displayed format shown below.



Program Source Codes:

❶ OPEN Command Button:

```
Private Sub Command1_Click()
  ' Buffer to hold input string
   Dim Instring As String
   ' Use COM1.
   MSComm1.CommPort = 1
   ' 9600 baud, no parity, 8 data, and 1 stop bit.
   MSComm1.Settings = "9600,N,8,1"
   ' Tell the control to read entire buffer when Input
   ' is used.
   MSComm1.InputLen = 0
   ' Open the port.
  MSComm1.PortOpen = True
  End Sub
```

❶ SEND Command Button:

```
Private Sub Command2_Click()
  ' Send Get AI command to ADAM-4011 Module at address 01H.
   MSComm1.Output = "#01" & Chr$(13)
  ' Wait for data to come back to the serial port.
   Do
     DoEvents
   Buffer$ = Buffer$ & MSComm1.Input
   Loop Until InStr(Buffer$, vbCr)
  ' Read the response till the carriage return character.
  Text1.Text = Buffer$
  ' Display the reading.
  End Sub
```

❶ CLOSE Command Button

```
Private Sub Command3_Click()
   ' Close the serial port.
  MSComm1.PortOpen = False
  End Sub
```

3

I/O Modules

3.1 The common specification of ADAM-4100 I/O series

Communication

- Speeds: support from 1200 to 115K bps
- Max. communication distance: 4000 feet (1.2 km)
- Communication error checking with checksum
- Asynchronous data format:
  Advantech protocol: 1 start bit, 8 data bits, 1 stop bit, no parity
  Modbus protocol: 1 start bit, 8 data bits, 1 or 2 stop bit, parity check (none, odd, even)
- Up to 256 multi-drop modules per serial port
- Online module insertion and removal
- Transient suppression on RS-485 communication lines
- Reset default setting
- Power and communication LED indicators

Environmental

- Operating Temperature -40 ~ 85° C (-40 ~ 185° F)
- EMI Meets FCC Class A and CE
- Storage Temperature -40 ~ 85° C (-40 ~ 185° F)
- Humidity 5 ~ 95%, non-condensing
- EFT : 3kV
- ESD : 8KV

Power Requirements

- Unregulated +10 ~ +48 VDC
- Surge Protection: 1kV

Mechanical

- Case ABS & PC with captive mounting hardware
- Plug-in screw terminal block accepts wire size :#14-22 AWG, stripped length: 5 mm

3.2 ADAM-4117 8-channel Analog Input Module

The ADAM-4117 is a 16-bit, 8-channel analog input module that provides programmable input ranges for all channels. This module is an extremely cost-effective solution for industrial measuring and monitoring applications. It is not only able to endure under the harsh environment, but also hold a more robust design. The detailed specification and enhancements are described as the following.

ADAM-4117

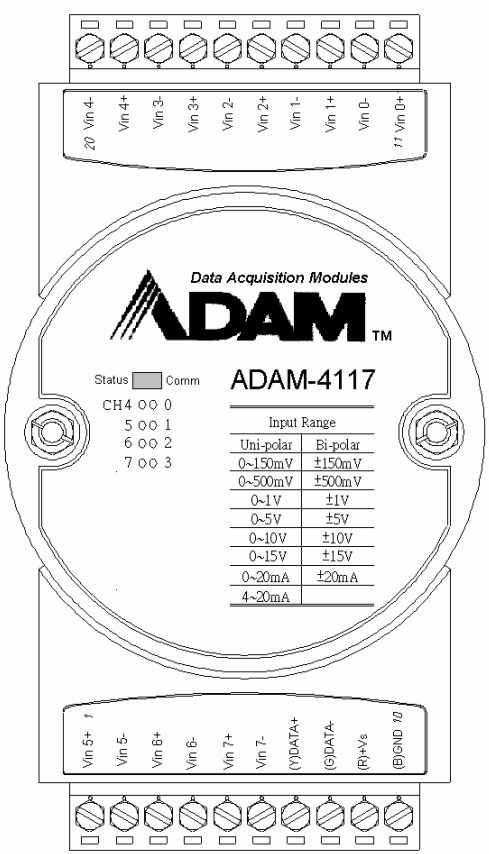


Figure 3-1 ADAM-4117 8-channel Analog Input Module

Specification

Analog Input

- Effective Resolution: 16-bit
- Channel: eight differential and independent configuration channels
- High common mode :200Vdc
- ASCII command and Modbus protocol
- lnput Type: mV, V(support Uni-polar and Bi-polar), mA
- lnput Range: 0~150mV, 0~500mV, 0~1V, 0~5V, 0~10V, 0~15V
  ±150 mV, ±500 mV, ±1V, ±5 V, ±10 V, ±15V, 0~20mA,
  ±20 mA, 4~20mA
- Isolation Voltage 3000 VDC
- Fault and Over voltage: Protection up to ±60V
- Sampling Rate: 10/100 samples/sec(selected by Utility)
- Input Impedance 20 MΩ
- Accuracy : Voltage mode : ±0.1% or better
  Current mode : ±0.2% or better
- Zero Drift ±6μV/° C
- Span Drift ±25 ppm/° C
- CMR @ 50/60 Hz 92 dB min.
- Built-in Dual Watchdog Timer
- Built-in TVS/ESD Protection
- Power Consumption 1.2 W @ 24 VDC

Jumper setting

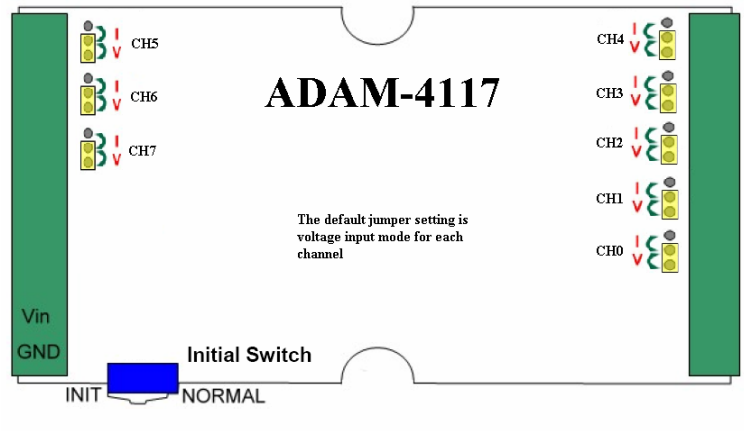
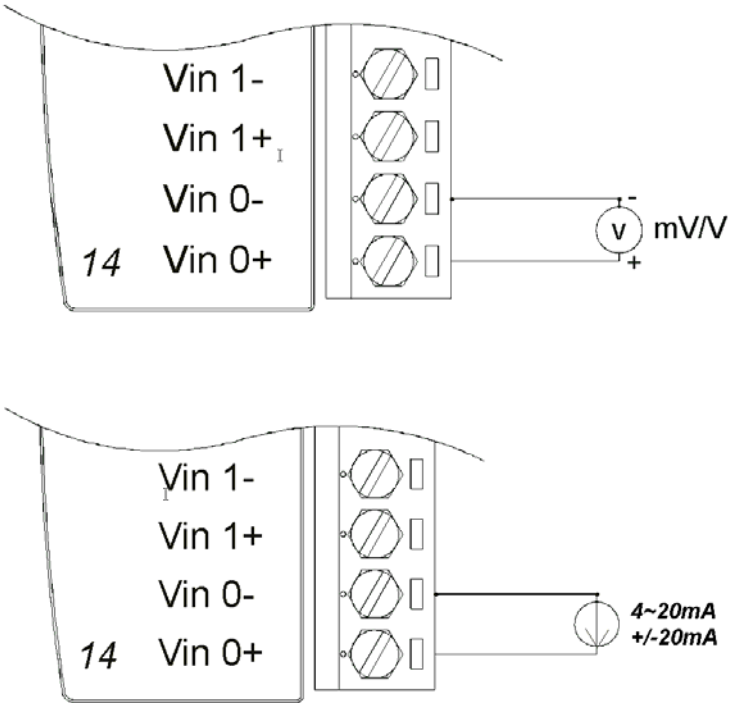


Figure 3-2 ADAM-4117 Jumper setting



***There is a resister built into ADAM-4117 for the current input mode**

Figure 3-3 ADAM-4117 wiring application

3.3 ADAM-4118 8-channel Thermocouple Input Module

The ADAM-4118 is a 16-bit, 8-channel Thermocouple input module that provides programmable input ranges on all channels. It accepts various Thermocouple inputs (Type J, K, T, E, R, S, B) and provides data to the host computer in engineering units (°C) In order to satisfy various temperature requirements in one module, each analog channel is allowed to configure each individual range for several applications.

ADAM-4118

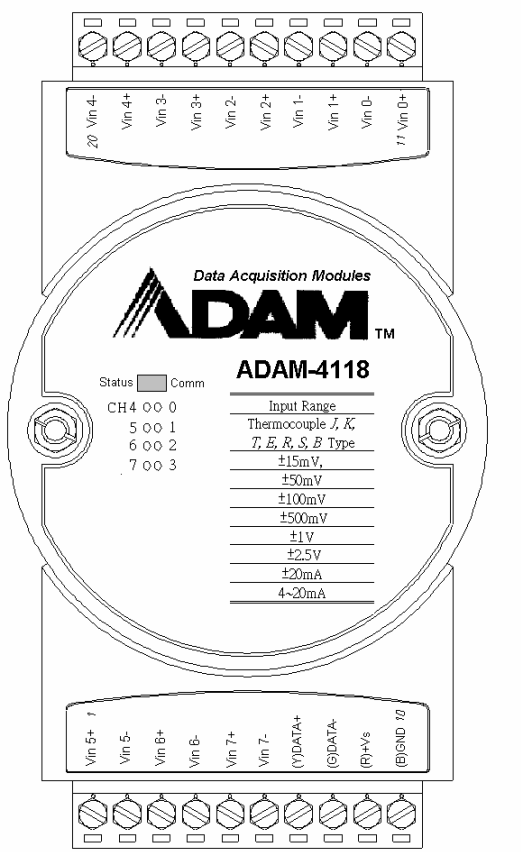


Figure 3-4 ADAM-4018 8-channel Thermocouple Input Module

Specification

Analog Input

- Effective Resolution: 16-bit
- Channel: 8 differential
- ASCII command and Modbus protocol
- Input type & range:

Thermocouple

J	0 ~ 760 °C
K	0 ~ 1370 °C
T	-100 ~ 400 °C
E	0 ~ 1000 °C
R	500 ~ 1750 °C
S	500 ~ 1750 °C
B	500 ~ 1800 °C

Voltage mode

±15mV, ±50mV, ±100mV, ±500mV, ±1V, ±2.5V

Current mode

±20 mA, +4~20 mA

- Isolation Voltage 3000 V_{DC}
- Fault and Over voltage: Protection up to ±60V
- Sampling Rate: 100 samples/sec (Max).
- Input Impedance 20 MΩ
- Accuracy of voltage mode : ±0.1% or better
- Accuracy of Current mode and high speed mode: ±0.2% or better
- Zero Drift: ±6μV/° C
- Span Drift: ±25 ppm/° C
- CMR @ 50/60 Hz 92 dB min.
- Built-in Dual Watchdog Timer
- Built-in TVS/ESD Protection
- Power Consumption 0.5W @ 24 VDC

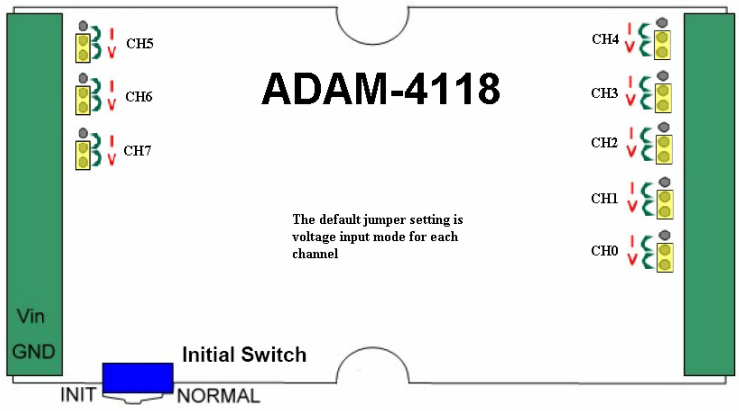
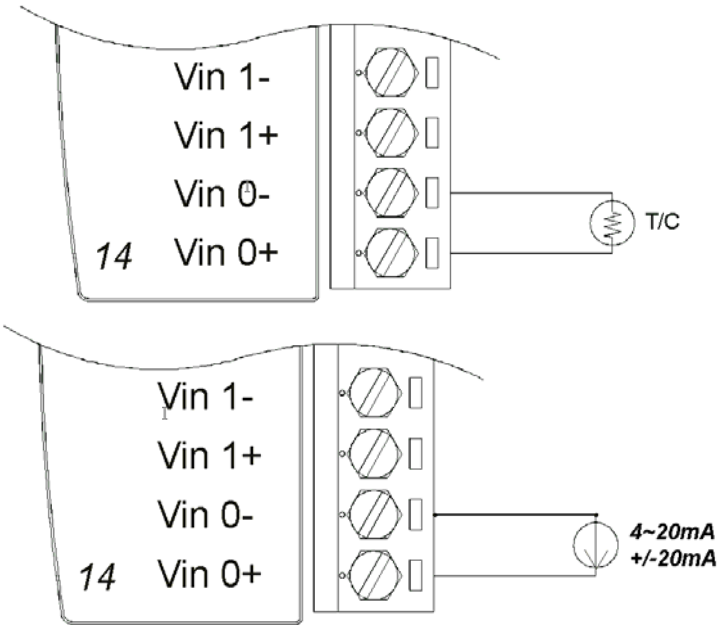


Figure 3-5 ADAM-4118 jumper setting



There is a resister built into ADAM-4118 for the current input mode

Figure 3-6 ADAM-4118 wiring application

NOTE: **1.** Because the CJC sensor of ADAM-4118 is located in the side of channel 0 to 4, the measurement will have the difference **± 1 ^{0}C** between channel 0 ~ 4 and channel 5 ~ 7.

2. The *ADAM-4118 Input Range Accuracy for Thermocouple* is showed as below

ADAM-4118 Input Range Accuracy for Thermocouple

Input Range	Typical Accuracy	Maximum Error	Units
J thermocouple 0 to 760 °C	±1.0	±1.5	°C
K thermocouple 0 to 1370 °C	±1.0	±1.5	°C
T thermocouple -100 to 400 °C	±1.0	±1.5	°C
E thermocouple 0 to 1000 °C	±1.0	±1.5	°C
R thermocouple 500 to 1750 °C	±1.2	±2.5	°C
S thermocouple 500 to 1750 °C	±1.2	±2.5	°C
B thermocouple 500 to 1800 °C	±2.0	±3.0	°C

3.4 ADAM-4150 Digital I/O Module

The ADAM-4150 features a seven digital input 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 can be applied to controlling heaters, pumps and power equipment. The host computer can use the module's digital inputs to determine the limit status or safety switches or remote digital signals. Furthermore, **these DI channels can be used as 3 KHz counter. Aside from its intelligent DI functions, the Digital Output channels also support 1 KHz pulse output function.**

ADAM-4150 has two situations, Normal & Communication. When the situation is at Normal, the LED of Status will be on. If the situation is at Communication, both the LEDs of Status & Comm will be on.

Here is the meaning of LED color for Status and Comm.

Situation	Status	Comm
Normal	**On (Red)**	**-**
Communication	**On (Red)**	**On (Green)**

ADAM-4150

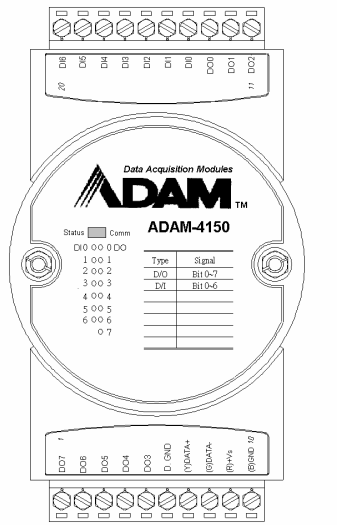


Figure 3-7 ADAM-4150 Digital I/O Module

Specification

- Channels:
  - 7 input channels
  - 8 output channels

- Digital Input:
  Dry contact:
  - Logic level 0: Close to GND.
  - Logic level 1: Open

  Wet contact:
  - Logic level 0: +3V max.
  - Logic level 1: +10V to +30V

  Isolation voltage: $3000V_{DC}$
  Support 3 KHz counter
  Support Digital Filter Function

- Digital Output:
  - Open drain to 40 V, 0.8A max.
    Maximum power dissipation 1 W load
    Ron Maximum: 150m ohm
  - Support 1 KHz pulse output
  - Support communication fail safety value

- Power Consumption:
  0.4W (Typical); 0.7W (Max)

- Built-in Dual Watchdog Timer

Notice: 1. Digital Filter Function is working on Counter mode and it can set the minimum width of low and high signal to filter unwanted noise.
2. Communication fail safety value is to force the DO channels to safety status when communication is in time-out and over pre-defined period.

Application Wiring

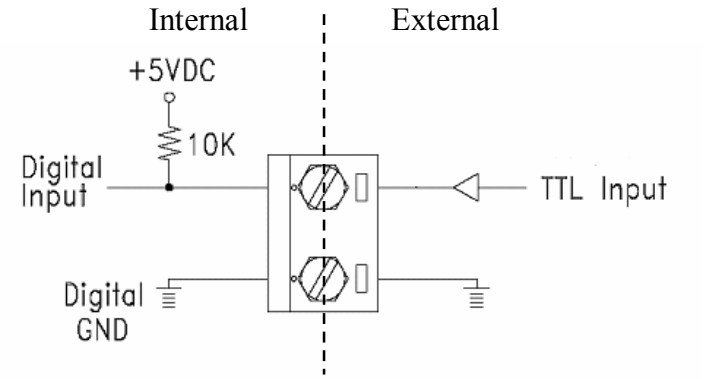


Figure 3-8 TTL Digital Input (ADAM-4150)

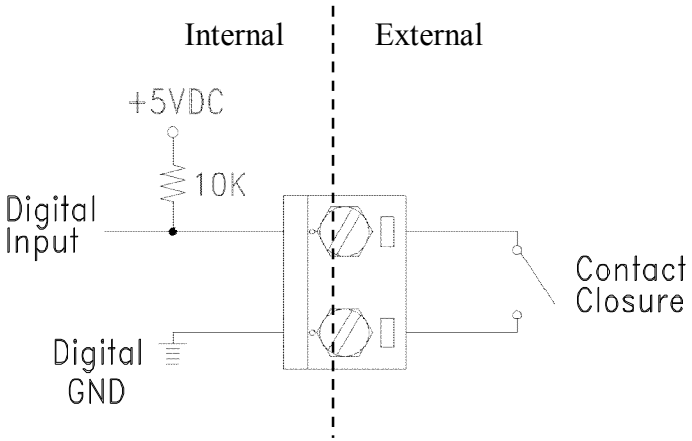


Figure 3-9 Contact Closure Input (ADAM-4150)

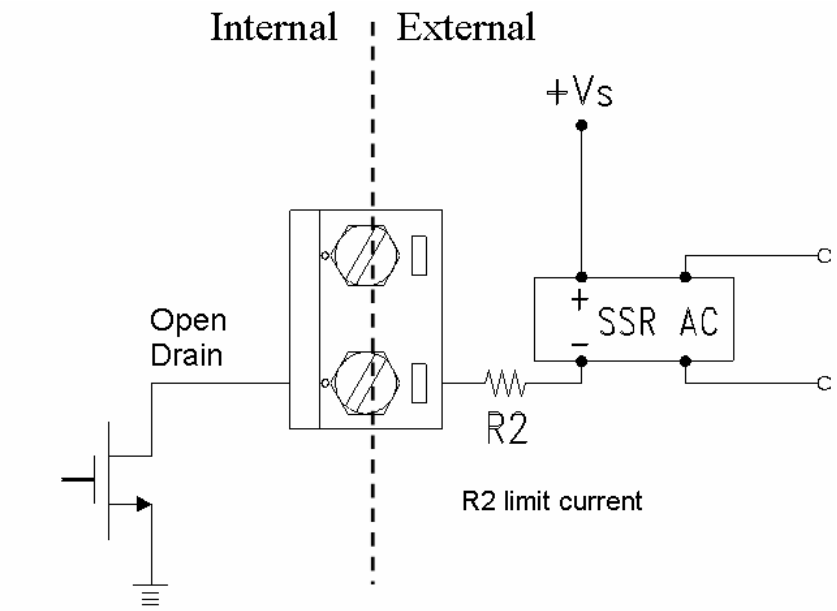


Figure 3-10 Digital Output used with SSR (ADAM-4150)

3.5 ADAM-4168 Relay Output Module

The ADAM-4168 Relay Output Module provides eight form A channels and is excellent for ON/OFF control or low-power switching applications. **Furthermore, all of the Digital Output channels can also support 100 Hz pulse output function.**

ADAM-4168

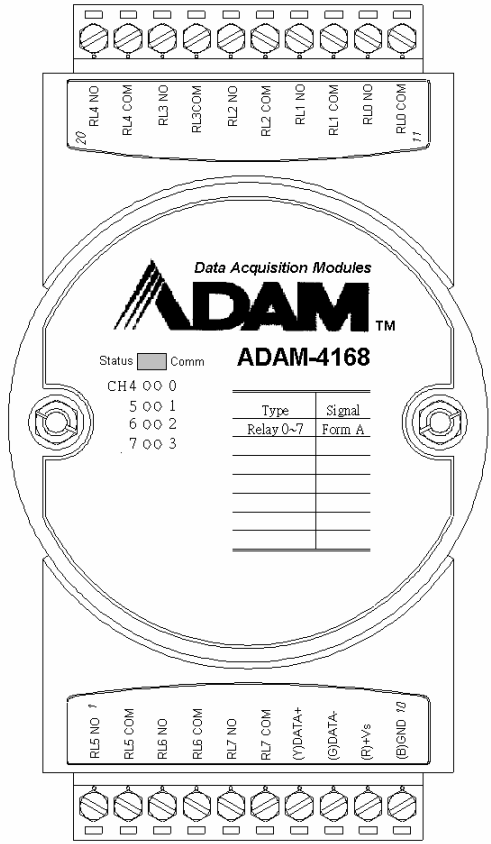


Figure 3-11 ADAM-4168 8-channel Relay Output Module

Specification

- **Number of Output Channel:** 8 Form A
- **Contact Rating: AC:0.5 A@120 V; 0.25A@240V**
  DC:1A@30V; 0.3A@110V
- **Breakdown Voltage :** 750 V_{AC} (50/60 Hz)
- **Insulation Resistance:** 1000MΩ minimum @500V_{DC}
- **Power Consumption:** 0.4W(typical) 1.8W(Max)
- **Relay response Time(typical):** ON :3 ms Off: 1 ms
- **Total switching time:** 10 ms
- **Support 100 Hz pulse output**
- **Support communication fail safety value**
- **Built-in Dual Watchdog Timer**

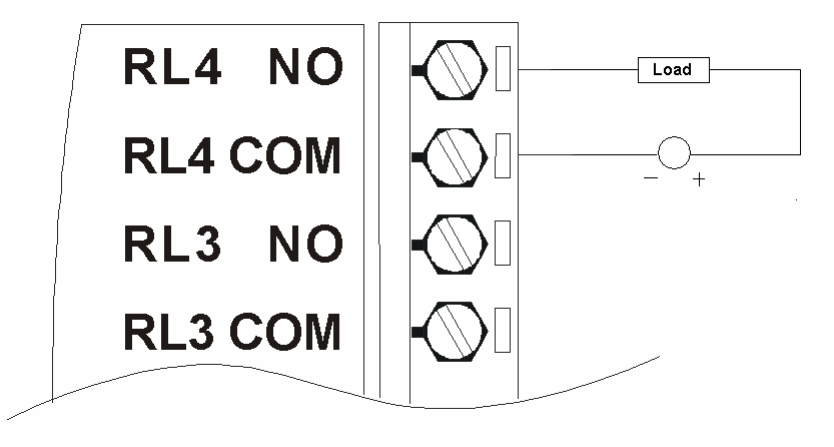


Figure 3-12 Form A relay output (ADAM-4168)

4

Command Set

4.1 Introduction

In order to avoid communication conflicts among devices trying to send data simultaneously, all the traffics are coordinated by the host computer. This action is initiated by the host computer using a command/response protocol.

When the modules are not transmitting, they are in listening mode. The host issues a command to a module with a specified address and waits for the module's response. If there is no response, a timeout aborts the sequence and returns the control to the host.

Changing ADAM's configuration might require the module to perform auto calibration before changes may take effect. This is the case when the range is modified especially. The module has to perform all stages of auto calibration which is also performed during the start up. When the calibration process is underway, the module does not respond to any other commands. The command set in the following pages includes the exact delays that might occur as 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 by the address. Depending on the command, an optional data segment may follows by a command string. Furthermore, an optional two-character checksum may be appended to the total string. Every command is terminated by a carriage return (cr).

ALL COMMANDS SHOULD BE ISSUED IN UPPERCASE CHARACTERS!

Before the command set is given, we provide an I/O module commands search table to help you find the commands that you wish to use. The command set is divided into the following three categories:

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

Each Category starts with a command summary of a particular type of module.

Although commands in different subsections sometimes share the same format, the effect they have on a certain module can be completely different from others. For example, the configuration command %AANNTTCCFF affects analog input modules and analog output modules differently. The full command set for every module is listed below.

4.3 Analog I/O Module Commands Search Table

ADAM-4117 Command Table

Command Syntax	Command Description	Section
%AANNTTCCFF	Set the address, input range, baud rate, data format, checksum status and/or integration time	4.4.1
#AAN	Return the input value from channels number N	4.4.2
#AA	Return the input values from all channels of the specified analog input module	4.4.3
$AA0	Calibrate the analog input module to correct for gain errors	4.4.4
$AA1	Calibrate the analog input module to correct for offset errors	4.4.5
$AA2	Return the configuration parameters	4.4.6
$AA5VV	Enables/disables multiplexing simultaneously	4.4.7
$AA6	Read the status (Enable/disable) of all Channels for Multiplexing	4.4.8
$AAF	Return the firmware version code	4.4.9
$AAM	Return the module name	4.4.10
$AA7CiRrr	Input range setting individually	4.4.11
$AA8Ci	Get the input range of the assignment channel	4.4.12
$AAXnnnn	Set communication WDT value	4.4.13
$AAY	Get communication WDT setting value	4.4.14
$AAMC	Get Sample Rate of the Auto-Filter	4.4.15
#AAMKmm	Set Software Filter Enable/Disable	4.4.16
$AAMD	Read Software Filter channel Enable/Disable	4.4.17
#AAFQm	Locate the module	4.4.18

ADAM-4118 Command Table

Command Syntax	Command Description	Section
%AANNTTCCFF	Set the address, input range, baud rate, data format, checksum status and/or integration time	4.4.1
#AAN	Return the input value from channels number N	4.4.2
#AA	Return the input values from all channels of the specified analog input module	4.4.3
$AA0	Calibrate the analog input module to correct for gain errors	4.4.4
$AA1	Calibrate the analog input module to correct for offset errors	4.4.5
$AA2	Return the configuration parameters	4.4.6
$AA5VV	Enables/disables multiplexing simultaneously	4.4.7
$AA6	Read the status (Enable/disable) of all Channels for Multiplexing	4.4.8
$AAF	Return the firmware version code	4.4.9
$AAM	Return the module name	4.4.10
$AA7CiRrr	Input range setting individually	4.4.11
$AA8Ci	Get the input range of the assignment channel	4.4.12
$AAXnnnn	Set communication WDT value	4.4.13
$AAY	Get communication WDT setting value	4.4.14
$AAMC	Get Sample Rate of the Auto-Filter	4.4.15
#AAMKmm	Set Software Filter Enable/Disable	4.4.16
$AAMD	Read Software Filter channel Enable/Disable	4.4.17
#AAFQm	Locate the module	4.4.18
$AA3	Returns the value of the CJC sensor	4.4.19
$AA9SNNNN	CJC Offset Calibration	4.4.20

4.4 Analog I/O Module Command set

4.4.1

%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 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)

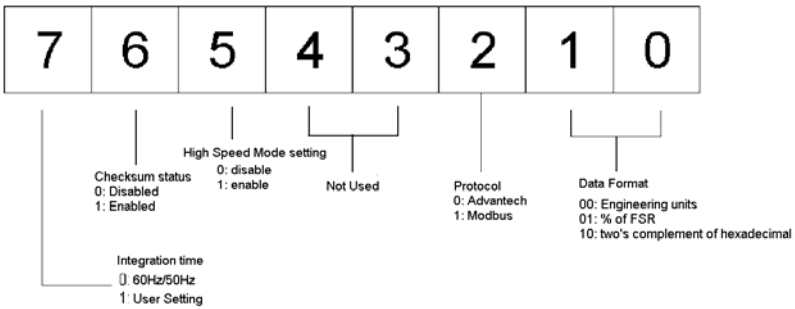


Figure 4-1 Data format for 8-bit parameter

(Continue %AANNTTCCFF)

Response

!AA(cr) if the command is valid.
?AA(cr) if an invalid parameter was entered or 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 or communication error, or even if the specified address does not exist.

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

? is a delimiter character indicating that 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 is configured to have address 23h to a new address of 24h, an input range ±2.5 V, baud rate 9600, and integration time 50 ms (60 Hz), engineering unit's data format and no checksum checking or generation.

The response indicates that the command was received.

Wait for 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. (Please refer to Baud rate and Checksum configuration in Chapter 2, for the correct procedure)*

Table 4-1 Input Range Codes (Type Code)

ADAM-4117

Input Range Code (Hex)	Input Range
07	4~20 mA
08	± 10 V
09	± 5 V
0A	± 1 V
0B	± 500 mV
0C	± 150 mV
0D	± 20 mA
15	± 15 V
48	0 ~ 10 V
49	0 ~ 5 V
4A	0 ~ 1 V
4B	0 ~ 500 mV
4C	0 ~ 150 mV
4D	0 ~ 20 mA
55	0 ~ 15 V

ADAM-4118

Input Range Code (Hex)	Input Range
00	± 15 mV
01	± 50 mV
02	± 100 mV
03	± 500 mV
04	± 1 V
05	± 2.5 V
06	± 20 mA
07	4~20 mA
0E	Type J Thermocouple 0^0 to 760^0 C
0F	Type K Thermocouple 0^0 to 1370^0 C
10	Type T Thermocouple -100^0 to 400^0 C
11	Type E Thermocouple 0^0 to 1000^0 C
12	Type R Thermocouple 500^0 to 1750^0 C
13	Type S Thermocouple 500^0 to 1750^0 C
14	Type B Thermocouple 500^0 to 1800^0 C

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
09	57.6 kbps
0A	115.2 kbps
0B	230.4 kbps

4.4.2

#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 #AAN(cr)

is a delimiter character.

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

N identifies the channel you want to read. The value can range from 0 to 7

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

Response >(data)(cr)

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

> is a delimiter character.

(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.

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

Example command: #120(cr)
response: >+1.4567(cr)

The command requests the analog input module at address 12h to return the input value of channel 0.

The analog input module responds an input value, +1.4567 volts, of channel 0.

#AA

Name Analo

4.4.3

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

> is a delimiter character.

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

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

Example command: #33(cr)

response: >+5.8222(cr)

The command accesses 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 accesses 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.

(Continue #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 two's complement)

	Two's complement	% of Span	Engineering units
under	0000	-0000	-0000
over	FFFF	+9999	+9999

NOTICE: *When modules measure **Thermocouple** input values that are outside of their configured range, they will send data that implies out of bound inputs. The next table shows the return values of the modules; however, it depends on the configured data format and input value which may falls under or exceeds the configured range.*

An "input out of bounds" warning occurs only when modules are configured for Thermocouple. Furthermore, the current and voltage measurement will return to the actual measured input if the readings fall outside of the configured range!

In the next example, the target module is configured that it has an input range of T/C type J (Input range: 0 - 760° C) and 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.

4.4.4

$AA0

Name	Span Calibration
Description	Calibrates an analog input module to correct gain errors.
Syntax	$AA0(cr) $ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module 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 or communication error, or even if the specified address does not exist. ! is a delimiter character indicating that a valid command was received. ? is a delimiter character indicating that 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 calibrating input signal should be connected to the analog input module before and during the calibration.

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

4.4.5

$AA1

Name Offset Calibration.

Description Calibrates an analog input module to correct for offset errors.

Syntax $AA1(cr)

$ is a delimiter character.

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

1 represents the offset calibration command.

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

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

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

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

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

? is a delimiter character indicating that 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 calibrating input signal should be connected to the analog input module before and during the calibration.

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

4.4.6

$AA2

Name	Configuration Status
Description	The command requests the return of the configuration data from the analog input module at address AA.
Syntax	$AA2(cr) $ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module. 2 is the Configuration Status command. (cr) is the terminating character, carriage return (0Dh).
Response	!AATTCCFF(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 even if the specified address does not exist. ! is a delimiter character indicating that a valid command was received. ? is a delimiter character indicating that the command was invalid. AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module. TT represents the type of code that determines the input range. CC represents the baud rate code. FF is a hexadecimal number that equals to an 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. (cr) is the terminating character, carriage return (0Dh). (Also see the %AANNTTCCFF configuration command)

(Continue $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 are in use.

4.4.7

$AA5VV

Name	Enable/disable channels for multiplexing
Description	Enables/disables multiplexing simultaneously for separating channels of a specified input module
Syntax	$AA5VV(cr) $ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of analog input module. 5 is the enable/disable channels command. VV are the two hexadecimal values which are interpreted as two binary words (4-bit) by the module. The first word represents the status of channel 4-7, and 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).
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 or communication error, or even if the specified address does not exist. ! is a delimiter character indicating that a valid command was received. ? is a delimiter character indicating that 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: $00581(cr) response: !00(cr) Hexadecimal 8 equals to binary 1000, which enables channel 7 and disables channels 4, 5, and 6. Hexadecimal 1 equals to binary 0001, which enables channel 0 and disables channel 1,2, and 3.

4.4.8
$AA6

Name Read Channel Status

Description Read the status (Enable/disable) of all Channels for Multiplexing

Syntax $AA6(cr)

AA (range 00-FF) represents the 2-character hexadecimal address where stores the channel status you want to read. 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 or communication error, or even if the specified address does not exist.

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

? is a delimiter character indicating that the command was invalid.

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

VV are two hexadecimal values which are interpreted as two binary words (4-bit). The first word represents the status of channel 4-7, and 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 with all its multiplex channels enabled (FF equals 1111 and 1111).

4.4.9

$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	$AAF (cr) $ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address that you want to access. 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 or communication error, or even if the specified address does not exist. ! is a delimiter character indicating that a valid command was received. AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module. (Version) is the version code of the module's firmware at address AA. (cr) is the terminating character, carriage return (ODh).

4.4.10

$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 that you want to access.

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

! is a delimiter character indicating that 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).

4.4.11

$AA7CiRrr

Name Single Channel Range Configuration

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

Syntax $AA7CiRrr(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address for configuration.

7 represents the range configuration command.

Ci represent the specified input channel you want to configure.

Rrr represent the type and range you want to set. (Please refer to Table 4-1 to check range code)

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

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

? is a delimiter character indicating that 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: $027C5R21(cr)

response: !02(cr)

The command configures the range of channel 5 in the analog input module at address 02 as Pt100 (IEC) 0~100℃.

4.4.12

$AA8Ci

Name Read Single Channel Range Configuration

Description This command reads 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 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.

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

? is a delimiter character indicating that the command was invalid.

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

Ci represents the specified input channel you read.

Rrr represent the type and range setting in the specified channel. (Please refer to Table 4-1 to check range code)

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

Example command: $028C5(cr)

response: !02C5R21(cr)

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

4.4.13

$AAXnnnn

Name	Watchdog Timer Setting
Description	This command set the Watchdog Timer communication cycle.
Syntax	$AAXnnnn(cr) $ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address which is to be read. X represents the setting of WDT command. nnnn (range 0000~9999) represents the specified value of communication cycle. (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 or communication error, or even if the specified address does not exist. ! is a delimiter character indicating that a valid command was received. ? is a delimiter character indicating that 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 sets the WDT cycle as 1234 from address 02 of the input module.

NOTICE: *If the value of "nnnn" is 0000, the communication WDT function will be disabled.*

4.4.14

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

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

? is a delimiter character indicating that the command was invalid.

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

nnnn (range 0000~9999) represents the specified value of communication cycle.

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

Example command: $02Y(cr)

response: !020030(cr)

The command read the WDT cycle as 0030 from address 02 of the input module.

4.4.15

#AAMC

Name	Get Sample Rate of the Auto-Filter
Description	This command read the Sample Rate of Auto-Filter.
Syntax	#AAMC(cr) # is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address which is to be read. MC represents the reading Sample Rate of Auto-Filter command. (cr) is the terminating character, carriage return (0Dh).
Response	!AAmmm(cr) if the command was valid. ?AA(cr) if an invalid operation was entered. There is no response if the module detects a syntax or communication error, or even if the specified address does not exist. ! is a delimiter character indicating that a valid command was received. ? is a delimiter character indicating that the command was invalid. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module. mmm represents the specified value of AI sample rate (cr) represents terminating character, carriage return (0Dh).
Example	command: $02MC(cr) response: !02016(cr) The command reads the sample rate of 016 from address 02 of the input module.

4.4.16

#AAMKmm

Name Set Software Filter Enable/Disable

Description This command configures the Software Filter status (Enable/disable) of all Channels for Multiplexing

Syntax #AAMKmm(cr)

is a delimiter character.

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

MK represents the configuration Software Filter status of (Enable/disable) of all Channels for multiplexing command.

mm are the two hexadecimal values which are interpreted as two binary words (4-bit). The first word represents the status of channel 4-7, and 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).

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

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

? is a delimiter character indicating that 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: $01MK23(cr)
response: !01(cr)
Hexadecimal 2 equals to binary 0010, which enables software filter of channel 5 and disables channels 4, 6, and 7.

Hexadecimal 3 equals to binary 0011, which enables software filter channel 0 and 1, disables channel 2, and 3.

4.4.17

$AAMD

Name	Read Software Filter channel Enable/Disable
Description	This command reads the Software Filter status (Enable/disable) of all Channels for Multiplexing
Syntax	#AAMD(cr) \# is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address which is to be read. MD represents the reading Software Filter status of (Enable/disable) of all Channels for multiplexing command. (cr) is the terminating character, carriage return (0Dh).
Response	!AAmm (cr) if the command was valid. ?AA(cr) if an invalid operation was entered. There is no response if the module detects a syntax or communication error, or even if the specified address does not exist. ! is a delimiter character indicating that a valid command was received. ? is a delimiter character indicating that the command was invalid. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module. mm are the two hexadecimal values which are interpreted as two binary words (4-bit). The first word represents the status of channel 4-7, and 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) represents terminating character, carriage return (0Dh).
Example	command: $01MD(cr) response: !0132(cr) Hexadecimal 3 equals to binary 0011, which enables software filter channels 4, 5 and disables channels 6, 7. Hexadecimal 2 equals to binary 0010, which enables software filter channel 1 and disables channels 0, 2, 3.

4.4.18

$AAFQm

Name	Locate the module
Description	This command light the LED to help user to know which model is Located
Syntax	#AAFQm(cr) # is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address which is to be read. FQ represents the lighting of the LED for locating a specific model. m: 1 LED Turn ON(10 seconds) 0 Clear Locate (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 or communication error, or even if the specified address does not exist. > is a delimiter character indicating that a valid command was received. ? is a delimiter character indicating that 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: $01FQ1(cr) response: >01 (cr) The command at address 01 of the module lights up the “Status LED” for 10 seconds.

4.4.19

$AA3

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

4.4.20

$AA9SNNNN

Name CJC Offset Calibration

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

Syntax $AA9SNNNN(number of counts)(cr).

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which contains the CJC Status you wish to retrieve.

9 is CJC Status command.

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

NNNN(number of counts) a four character hexadecimal "count" value. Each count equals approximately to 0.009° C. The value can range from 0000 to FFFF.

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

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

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

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

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

? is a delimiter character indicating that 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: $079+0042(cr)
response: !07(cr)

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

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

4.5 Digital I/O Module Commands Search Table

ADAM-4150 Command Table

Command Syntax	Command Description	Section
%AANNTTCCFF	Set the address, input range, baud rate, data format, checksum status and/or integration time	4.6.1
$AA2	Return the configuration parameters	4.6.2
$AA6	Returns the values of the digital I/O channels	4.6.3
#AABB(data)	Writes specified values to either a single channel or all channels simultaneously	4.6.4
$AAF	Return the firmware version code	4.4.9
$AAM	Return the module name	4.4.10
$AAX0TTTTDD	Write Safety value	4.6.5
$AAX1	Read safety value	4.6.6
$AAX2	Read safety flag	4.6.7
$AACIIIIIIIIIIIIOOOOOOOOOOOOOOOO	Set DI/O status	4.6.8
$AAC	Read DI/O Status	4.6.9
$AACICjII	Set DI Status	4.6.10
$AACICj	Read DI Status	4.6.11
$AACOCjOO	Set DO Status	4.6.12
$AACOCj(cr)	Read DO Status	4.6.13
$AA0CjLLLLLLLLHHHHHHHH	Set DI filter input width	4.6.14
$AA0Cj	Read DI filter input width	4.6.15
$AA9n(lw)(hw)(ld)(hd)	Set pulse output input width	4.6.16

$AA9n	Read pulse output Status	4.6.17
#AAN(cr)	Read counter or frequency value	4.6.18
#aaERFFccvvvvvvvv(cr)	Set DO pulse output counts	4.6.19
$aaERFFcc(cr)	Read DO current pulse output counts	4.6.20
@AACACj(cr)	Clear Latch Alarm	4.6.21
$AA5NS(cr)	Start/Stop counter	4.6.22
$AA5N(cr)	Read Counter Start/Stop status	4.6.23
$AA6N(cr)	Clear counter	4.6.24
#AAFQm	Locate the module	4.4.18

ADAM-4168 Command Table

Command Syntax	Command Description	Section
%AANNTTCCFF	Set the address, input range, baud rate, data format, checksum status and/or integration time	4.6.1
$AA2	Return the configuration parameters	4.6.2
$AA6	Returns the values of the digital I/O channels	4.6.3
#AABB(data)	Writes specified values to either a single channel or all channels simultaneously	4.6.4
$AAF	Return the firmware version code	4.4.9
$AAM	Return the module name	4.4.10
$AAX0TTTTDD	Write Safety value	4.6.5
$AAX1	Read safety value	4.6.6
$AAX2	Read safety flag	4.6.7
$AACIIIIIIIIIIIIOOOOOOOOOOOOOOOO	Set DI/O status	4.6.8
$AAC	Read DI/O Status	4.6.9
$AACOCjOO	Set DO Status	4.6.12
$AACOCj	Read DO Status	4.6.13
$AA9n(lw)(hw)(ld)(hd)	Set pulse output input width	4.6.16
$AA9n	Read pulse output Status	4.6.17
#aaERFFccvvvvvvvv(cr)	Set DO pulse output counts	4.6.19
$aaERFFcc(cr)	Read DO current pulse output counts	4.6.20
#AAFQm	Locate the module	4.4.18

4.6 Digital I/O Module Command Set

4.6.1

%AANNTTCCFF

Name Configuration

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

Syntax %AANNTTCCFF(cr)

% is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be configured.

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

TT represents the type of code, which is always set to 40 for digital I/O modules.

CC represents the baud rate code. (Please see next page, Table 4-3)

FF is a hexadecimal number that equals to an 8-bit parameter that represents the checksum status and protocol.

Bits 3 through 5 and bit 0, 1, 7 are not used and are being set to 0. (Please see Figure 4-2)

Bit 6 is the selection of checksum and bit 2 is the selection of protocol: (0: advantech, 1: modbus).

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

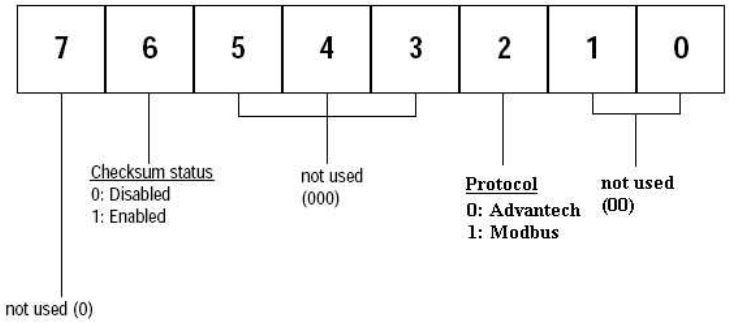


Figure 4-2 ***Checksum and protocol***

(Continue %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 changing baud rate or checksum settings are attempted.

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

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

? is a delimiter character indicating that 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 of 9600 and no checksum checking. It also supports Advantech protocol. The response indicates that the configuration was successful.

Table 4-3 *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
09	57.6 kbps
0A	115.2 kbps

NOTICE: *All configuration parameters can be changed dynamically, except checksum and baud rate parameters. They can only be altered when the module is under initial mode.*

4.6.2

$AA2

Name Configuration Status command

Description Returns the configuration parameters of the digital I/O module.

Syntax $AA2(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

2 is Configuration Status command.

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

This command requests the return of the configuration data from the digital I/O module at address AA.

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

?AA(cr) if an invalid command has been issued.

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

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

? is a delimiter character indicating that the command was invalid.

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

TT represents the type of code, which is always 40.

CC represents the baud rate code.

FF is a hexadecimal number that equals to an 8-bit parameter that represents the checksum status and protocol.

Bits 3 through 5 and bit 0, 1, 7 are not used and are being set to 0. (See Figure 4-3)

Bit 6 is the selection of checksum and bit 2 is the selection of protocol: (0: advantech, 1: modbus).

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

(continue $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 of 9600, no checksum function. At last, the module supports Advantech protocol.

4.6.3

$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 feedback value of its digital output channels.

Syntax $AA6(cr)

$ is a delimiter character.

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

6 is the Digital Data In command.

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

Response !(dataOutput)(dataInput)00(cr)
if the command was valid. (ADAM-4150)

!(dataOutput)0000(cr)
if the command was valid. (ADAM-4168)

?AA(cr) if an invalid command has been issued.

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

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

? is a delimiter character indicating that the command was invalid.

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

(dataOutput) two-character hexadecimal value which either is the feedback 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).

(continue $AA6)

Example

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

This example is for ADAM-4150. The first two characters of the response, value 11h (00010001), indicate that digital output channels 0 and 4 are both ON, and channels 1, 2, 3, 5, 6, 7 are OFF. The following two characters of the response, 22h (00100010), indicate that digital input channels 1 and 5 are both HIGH, and channels 0, 2, 3, 4, 6, 7 are LOW.

4.6.4

#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 for the output.
BB is used to indicate whether a single channel or all the channels will be set. In the last character of BB also indicates which channel it is.

1. Writing to all channels (write a byte): both characters should be equal to zero **(BB=00)**.
2. Writing to a single channel (write a bit): **First character is 1**, and the second character indicates channel number, which can range from 0 to B.

(data) is the hexadecimal representation of the digital output value(s). These two characters are for ADAM-4150 and 4168.

For writing to a single channel (bit), the first character is always 0. The value of the second character is either 0 or 1.
For writing to all channels (byte), both characters are significant (range 00h-FFh). The decimal equivalent of these two hexadecimal characters represents the channels' values. The value 7A can be converted to the following 8 channels representations for ADAM-4150 and ADAM-4168:

Digital Value:	0	1	1	1	1	0	1	0
ADAM-4150/4168 channel no.	7	6	5	4	3	2	1	0

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

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

? is a delimiter character indicating that the command was invalid.

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

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

Examples

command: #140005(cr)
response: >(cr)

An output byte with value 05h (00000101) is sent to address 14h. 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. Channel two of the digital I/O module is set to ON.

4.6.5

$AAX0TTTTDD

Name Write Safety Value

Description Force the DO channels to safety status when communication is in time-out and over pre-defined period.

Syntax $AAX0TTTTDD(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

X0 is to write the safety value command.

TTTT is the time, and it is 100ms per number. Turn off the Communication WDT when TTTT is set to 0.

DD is the two-hexadecimal character representing the desired input safety value. For Example: 7A

The meaning of 7A is as follows:

Adam-4117 channel no.	7	6	5	4	3	2	1	0
Digital value	0	1	1	1	1	0	1	0

7A means that the status of channels 1, 3, 4, 5, 6 are ON, and the rest are OFF.

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

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

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

? is a delimiter character indicating that the command was invalid.

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

4.6.6

$AAX1

Name Read Safety Value

Description Read the time-out setting and pre-defined safety status of DO channels.

Syntax $AAX1(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

X1 is the read safety value command.

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

DD is the two-hexadecimal character representing the desired input safety value.

For Example: 7A

The meaning of 7A is as follows:

Adam-4117 channel no.	7	6	5	4	3	2	1	0
Digital value	0	1	1	1	1	0	1	0

7A means channel 1, 3,4,5, 6 are ON; and the rest are OFF.

?AA(cr) if an invalid command has been issued.

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

? is a delimiter character indicating that the command was invalid

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

4.6.7

$AAX2

Name	Read Safety Flag
Description	Requests the Safty Flag of the digital I/O module to see whether the safety value has been executed since Write Safety Value command was set.
Syntax	$AAX2(cr) $ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed. X0 is the read safety flag command.
Response	!XX (cr) if the command is valid. XX is two-hexadecimal character – (00: OFF, 01: ON) ?AA(cr) if an invalid command has been issued. ! is a delimiter character indicating that a valid command was received ? is a delimiter character indicating that the command was invalid (cr) is the terminating character, carriage return (ODh).

4.6.8

$AACIIIIIIIIIIIIIOOOOOOOOOOOOOOOO

Name Set the status of all DI/O channels

Description Force the DI/O channels to different mode.

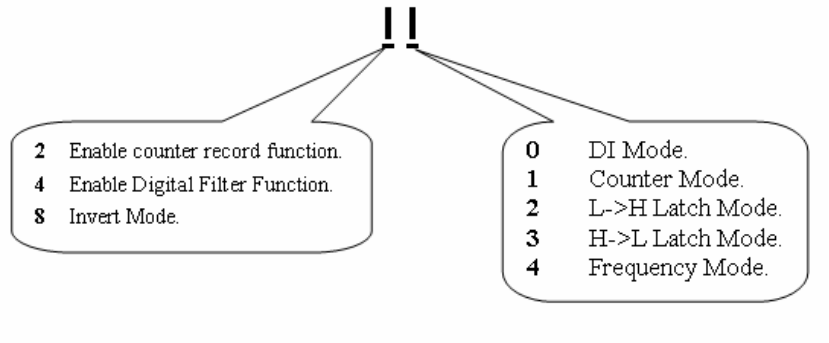
Syntax $ AACIIIIIIIIIIIIIOOOOOOOOOOOOOOOO

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

C is the set/read the DI/O status command.

IIIIIIIIIIIII is to set the 7 DI channels (every two I characters means one channel)



OOOOOOOOOOOOOOO is to set the 8 DO channels
(Every two O characters means one channel)

OO=00 DO Mode.

OO=01 Pulse Output Mode.

OO=02 L->H Delay Mode.

OO=03 H->L Delay Mode.

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

(continue $AACIIIIIIIIIIIIIOOOOOOOOOOOOOOO)

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

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

? is a delimiter character indicating that the command was invalid.

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

Example command: $02C00210440000000020000000000010203(cr)

response: >(cr)

The DI/O channels are set as the following at address 2

	II character	mode
0	00	DI Mode
1	21	Counter mode and enable counter record
2	04	Frequency Mode
3	40	Enable Digital Filter
4	00	DI Mode
5	00	DI Mode
6	00	DI Mode

DO channel	OO character	mode
0	02	L->H Delay Mode
1	00	DO Mode
2	00	DO Mode
3	00	DO Mode
4	00	DO Mode
5	01	Pulse Output Mod
6	02	L->H Delay Mode
7	03	H->H Delay Mode

4.6.9

$AAC

Description Read the status of all DI/O channels

Syntax $AAC(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

C is the set/read the DI/O status command.

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

Response **!**AAIIIIIIIIIIIIIIOOOOOOOOOOOOOOOO(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 or communication error, or even if the specified address does not exist.

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

IIIIIIIIIIIIIIOOOOOOOOOOOOOOOO is defined as te same $AACIIIIIIIIIIIIIIOOOOOOOOOOOOOOOO

? is a delimiter character indicating that the command was invalid.

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

IIIIIIIIIIIIII is to set the 7 DI channels (every two I characters mean one channel). Refer command **$AACIIIIIIIIIIIIIIOOOOOOOOOOOOOOOO**

OOOOOOOOOOOOOOOO is to set the 8 DO channels

OO=00 DO Mode.

OO=01 Pulse Output Mode.

OO=02 L->H Delay Mode.

OO=03 H->L Delay Mode.

4.6.10

$AACICjII

Name Set DI status of specified channel

Description Force the DI channel to different mode.

Syntax $AACICjII(cr)

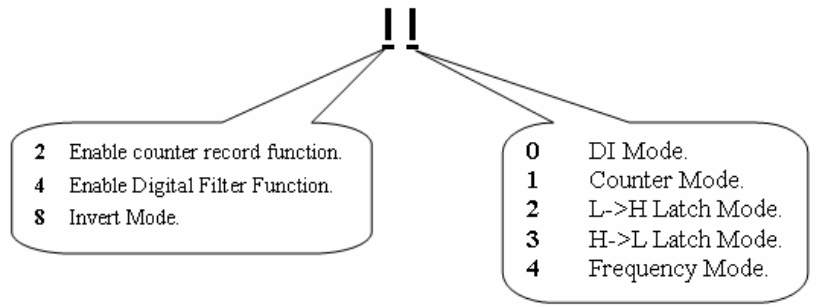
$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

CI is the set/read the DI status command.

Cj is the channel j

II is the two characters to set the DI mode



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

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

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

? is a delimiter character indicating that the command was invalid.

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

(continue $AACICjII)

Example command: $02CIC202(cr)
response: >(cr)
The channel 2 is set from Low to High latch mode

4.6.11

$AACICj

Name Read DI status of specified channel

Syntax $AACICj(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

CI is the set/read the DI status command.

Cj is the channel j

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

Response **!**AAII if the command was valid.

?AA(cr) if an invalid command has been issued.

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

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

II is defined as the same $AACICjII.

? is a delimiter character indicating that the command was invalid.

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

4.6.12

$AACOCjII

Name Set DO status of specified channel

Description Force the DO channel to different mode.

Syntax $ AACOCjII(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

CO is the set/read the DO status command.

Cj is the channel j

II is the two characters to set the DO mode

OO=00	DO Mode.
OO=01	Pulse Output Mode.
OO=02	L->H Delay Mode.
OO=03	H->L Delay Mode.

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

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

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

? is a delimiter character indicating that the command was invalid.

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

Example command: $02COC201(cr)

response: >(cr)

The channel 2 is set as the pulse output mode

4.6.13

$AACOCj

Name Read DO status of specified channel

Syntax $AACOCj(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

CO is the set/read the DO status command.

Cj is the channel j

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

Response **!**AAOO if the command was valid.

?AA(cr) if an invalid command has been issued.

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

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

OO is defined as the same $AACOCjOO.

? is a delimiter character indicating that the command was invalid.

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

4.6.14

$AA0CjLLLLLLLLHHHHHHHH

Name Set DI filter input width

Syntax $AA0CjLLLLLLLLHHHHHHHH(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

0 is the DI filter input command

Cj is the channel j

LLLLLLLL is the low level time and its range is 0x0~0xffffffff
HHHHHHHH is the high level time and its range is 0x0~0xffffffff
Unit: 0.1ms

Digital Filter Function is working on Counter mode and it can set the minimum width of low and high signal to filter unwanted noise.

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

Response !AA(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 or communication error, or even if the specified address does not exist.

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

? is a delimiter character indicating that the command was invalid.

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

4.6.15

$AA0Cj

Name Read DI filter input width

Syntax $AA0Cj (cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

0 is the DI filter input command

Cj is the channel j

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

Response !AALLLLLLLLHHHHHHHH(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 or communication error, or even if the specified address does not exist.

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

LLLLLLLL is the low level time and its range is 0x0~0xffffffff
HHHHHHHH is the high level time and its range is 0x0~0xffffffff
Unit: 0.1ms

? is a delimiter character indicating that the command was invalid.

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

4.6.16

$ AA9n(lw)(hw)(ld)(hd)

Name Set pulse output input width

Syntax $AA9n(lw)(hw)(ld)(hd)(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

9 is the DI filter input command

n is the channel n

(lw): 8-hexadicimal chrs Low Width=>0x0~0xffffffff

(hw): 8-hexadicimal chrs High Width=>0x0~0xffffffff

(ld): 8-hexadicimal chrs Low Delay=>0x0~0xffffffff

(hd): 8-hexadicimal chrs High Delay=>0x0~0xffffffff

Unit: 0.1ms

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

Response !AA(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 or communication error, or even if the specified address does not exist.

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

? is a delimiter character indicating that the command was invalid.

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

4.6.17

$AA9n

Name Read pulse output input width

Syntax $AA9n (cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

9 is the DI filter input command

n is the channel n

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

Response !AA(lw)(hw)(ld)(hd)(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 or communication error, or even if the specified address does not exist.

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

(lw): 8-hexadicimal chrs Low Width=>0x0~0xffffffff

(hw): 8-hexadicimal chrs High Width=>0x0~0xffffffff

(ld): 8-hexadicimal chrs Low Delay=>0x0~0xffffffff

(hd): 8-hexadicimal chrs High Delay=>0x0~0xffffffff

Unit: 0.1ms

? is a delimiter character indicating that the command was invalid.

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

4.6.18

#AAN

Name: Read Counter or Frequency Value

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

Syntax: #AAN(cr)

is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

N represents the Channel number.
(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 or communication error, or even if the specified address does not exist.

? is a delimiter character indicating that 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 module reading counter. 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 to return the data. The counter/frequency module at address 12 responds with value 000002FE, which is also equivalent to 766 in decimals, from counter 0.

4.6.19

#AAERFFccvvvvvvvv

Name Set DO current pulse output counts

Syntax #AAERFFccvvvvvvvv (cr)

is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

ERFF is the DO current pulse output counts command

cc is the channel number (00~07 means Ch0~Ch7)

vvvvvvvv: 8-hex character pulse output value. When it is set to **0, it will operate as continuous pulse output**

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

Response !AA if the command was valid.

?AA(cr) if an invalid command has been issued.

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

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

? is a delimiter character indicating that the command was invalid.

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

4.6.20

$AAERFFcc

Name Read DO current pulse output counts

Syntax $AAERFFcc (cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

ERFF is the DO current pulse output counts command

cc is the channel number (00~07 means Ch0~Ch7)

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

Response >AAmvvvvvvvv if the command was valid.

?AA(cr) if an invalid command has been issued.

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

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

m: 0 means non-continue mode
1 means continue mode

vvvvvvvv: 8-hex character pulse output value. When it is set to **0, it will operate as continuous pulse output**

? is a delimiter character indicating that the command was invalid.

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

4.6.21

@AACACj

Name Clear Latch Alarm

Description Both alarm states (High and Low) of the counter module are set to OFF.

Syntax @AACACj(cr)

@ is a delimiter character.

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

CA is the Clear Latch Alarm command

Cj is the channel j

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

Response !AA(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 or communication error, or even if the specified address does not exist.

! is a delimiter character indicating that 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: @05CAC1(cr)
response: !05(cr)

The counter module at address 05h and channel 1 are instructed to set both alarm states (High and Low) to OFF. The module confirms the execution.

4.6.22

$AA5NS

Name	Start/Stop Counter
Description	Request the counter/frequency module to start or stop the counting for counter 0 or 1.
Syntax	$AA5NS(cr) $ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed. 5 identifies the Start/Stop Counter command N determines whether the counter 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 even if the specified address does not exist. ! is a delimiter character indicating that 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 module replies with its address indicating that the command has been executed and counter 0 has started.

4.6.23

$AA5N

Name Read Counter Start/Stop Status

Description Requests the counter/frequency module to indicate whether counter 0 or counter 1 is active.

Syntax $AA5N(cr)

$ is a delimiter character.

AA (range 00-FF) represents the 2-character hexadecimal address which is to be accessed.

N determines which counter is active.
N = 0 represents counter 0
N = 1 represents counter 1

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

! is a delimiter character indicating that the command was valid.

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

S represents the counter status.
S = 0 indicates counting
S = 1 indicates not counting

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

Example
command: $0650(cr)
response: !061(cr)

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

4.6.24

$AA6N

Name	Clear Counter
Description	Clears 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 which is to be accessed. 6 the Clear Counter command. N determines which the counter 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 or communication error, or even if the specified address does not exist. ! is a delimiter character indicating that 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 indicating that the counter has been cleared.

5

Calibration

Analog input modules are already calibrated when you receive them. However, calibration sometimes does required and it is done through software. Calibration parameters are stored in the ADAM module's onboard EEPROM.

The ADAM modules come with utility software that supports the calibration of analog input and output. Aside from the calibration that is carried out through software, the modules incorporate automatic Zero Calibration and automatic Span Calibration at boot-up or reset.

5.1 Analog Input Module Calibration

Models: ADAM-4117, 4118

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 that you want to calibrate. You can do this by using the ADAM utility software. (Please refer to Appendix A, Utility Software.)
3. Use a precise voltage source to calibrate the module through Vin+ and Vin-.
4. Execute the Zero Calibration command. This is also done through the ADAM utility software. Apply the indicating signal to input channel then save the exact value.

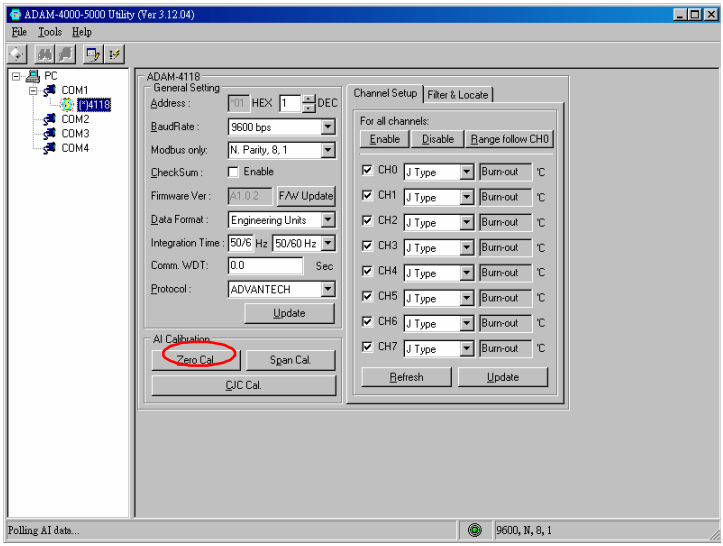


Figure 5-1 ***Zero Calibration***

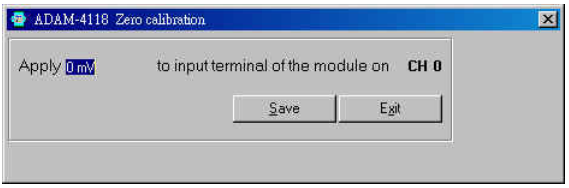


Figure 5-1(a) *Zero Calibratio*

5. Execute the Span Calibration command. This can be done with the ADAM utility software. Apply the indicating signal to input channel then save the exact value.

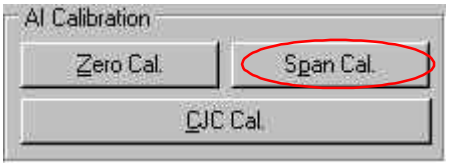


Figure 5-2 *Span Calibration*

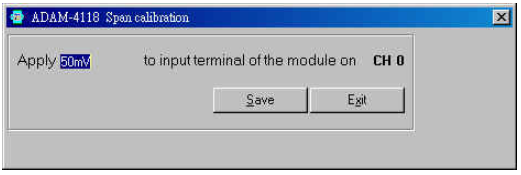


Figure 5-2(a) *Span Calibration*

6. For ADAM-4118 only, execute the CJC (cold junction compensation sensor) which calibration command. This is also done through the ADAM utility software. User can use CJC offset to adjust the exact temperature. For example, the input signal is 24 degree. But the reading is 23.8 degree. We can set the CJC offset is +0.2 degree to compensate.

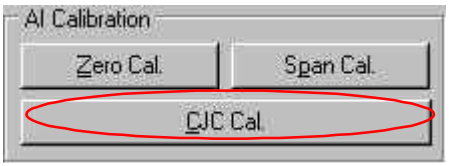


Figure 5-3(a) *Cold Junction Calibration*

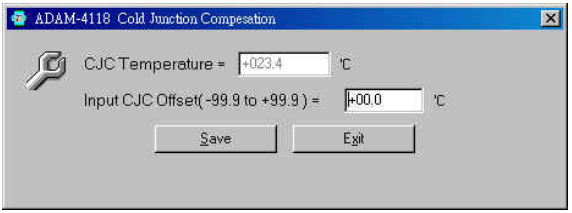


Figure 5-3*(b) Cold Junction Calibration*

NOTE:

11. Because the CJC sensor of ADAM-4118 is located in the side of channel 0 to 4, the measurement will have the difference **± 1 ^{o}C** between channel 0 ~ 4 and channel 5 ~ 7. The following table is *ADAM-4118 Input Range Accuracy for Thermocouple.*

2. It had better send back to Advantech Repair Service (RMA) to conduct calibration if user find or doubt the standard shift.

3. If user wants to conduct calibration by themselves, please use the high precision instruments to be a calibrating source and follow up process the "Zero", "Span" and "CJC" calibration in sequence.

Input Range	Typical Accuracy	Maximum Error	Units
J thermocouple 0 to 760 °C	±1.0	±1.5	°C
K thermocouple 0 to 1370 °C	±1.0	±1.5	°C
T thermocouple -100 to 400 °C	±1.0	±1.5	°C
E thermocouple 0 to 1000 °C	±1.0	±1.5	°C
R thermocouple 500 to 1750 °C	±1.2	±2.5	°C
S thermocouple 500 to 1750 °C	±1.2	±2.5	°C
B thermocouple 500 to 1800 °C	±2.0	±3.0	°C

A

Utility Software

ADAM-4100 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
- Auto-scan of connected modules
- Terminal emulation

The following text will give you a brief instruction on how to use the program.

A.1 Utility overview

Search the installed modules

The main screen consists of a menu bar at the top and a status field which displays information about the connected modules. When the modules are well connected, you may start the program and search for the modules by clicking the search icon as below. Please do check if the COM port and related settings are correct.

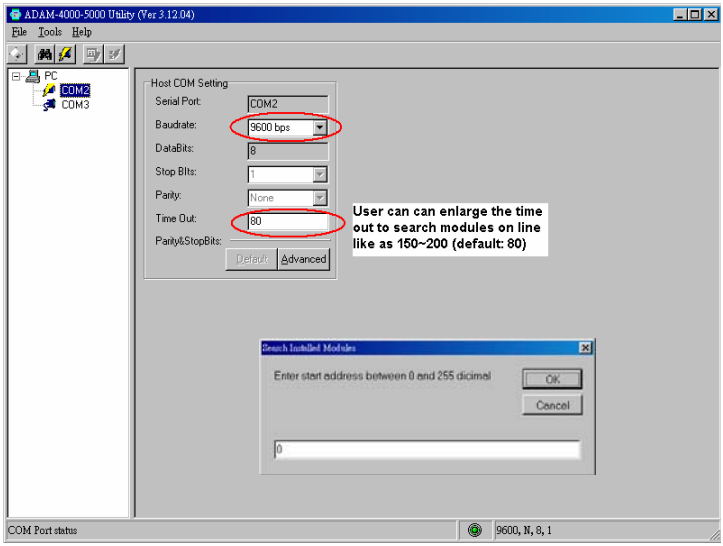


Figure A-1 Search screen

NOTICE: For configuring, 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*

Configuration

Click on the searched module which you would like to configure. Then, you will find a Setup page and related settings. An example is shown in Figure A-2 for an ADAM-4117 module.

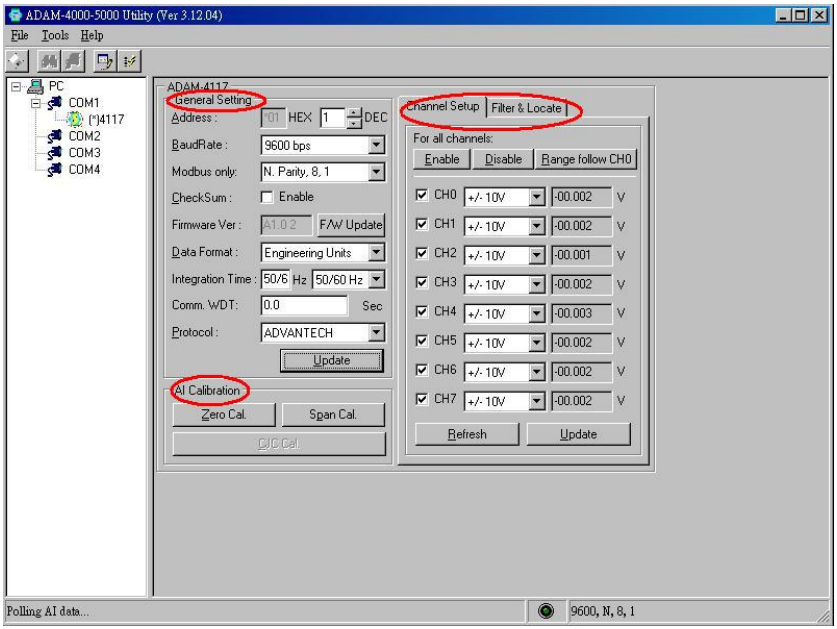


Figure A-2 Configuration Screen

There are three major areas on the status field, General Setting, calibration Area and Channels setting. You may change the settings by selecting the preferred items. Then click on the Update button.

The Checksum and Baud rate options need special attention since they can only be changed when an ADAM module is in the initial state. After you have made all the necessary changes to the module configuration, the utility will display the processed data automatically.

The data format is ***1 start bit, 8 data bits, 1 stop bit, no parity*** for *Advantech protocol* and ***1 start bit, 8 data bits, 1 or 2 stop bit, parity check (none, odd, even)*** for *Modbus protocol only*.

Terminal Function

When you would like to send and receive commands on the RS-485 line directly, you can use the Terminal function under Tools as shown below.

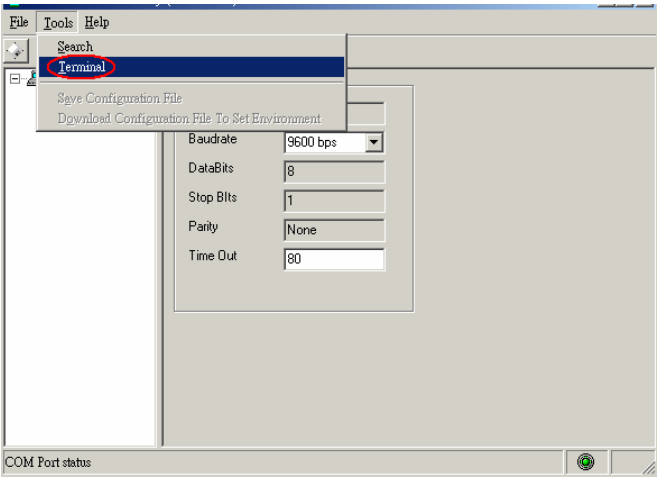


Figure A-3 Terminal Function

You can type in the ADAM ASCII command in the text box and click on the Send button for testing commands which are listed in Chapter 4 Command Set.

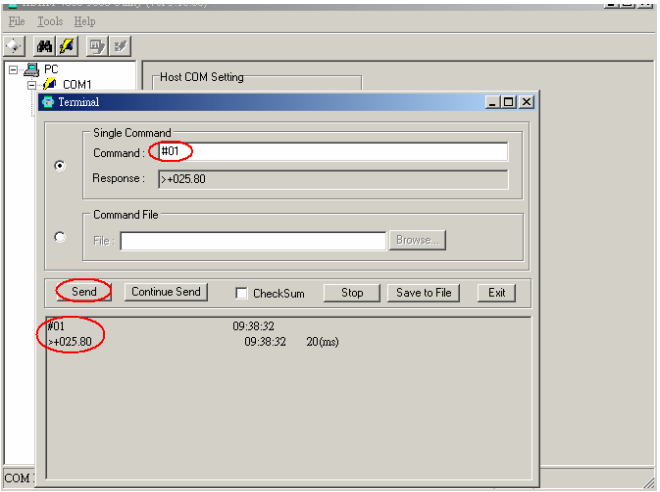


Figure A-4 Terminal Function

Utility Software

Notice: User can refer our help file to see more details for explanation of Utility operation.

A.2 Firmware update

The ADAM-4100 series are enhanced with the new and friendly function to update firmware online. In the past, the ADAM-4000 series need to be sent back to Repair Center for updating firmware if it is necessary. Now, user can use the new capability for following design and update.

1. They can only be implemented when an ADAM module is in the Initial state. Then, click on the button "F/W Update".

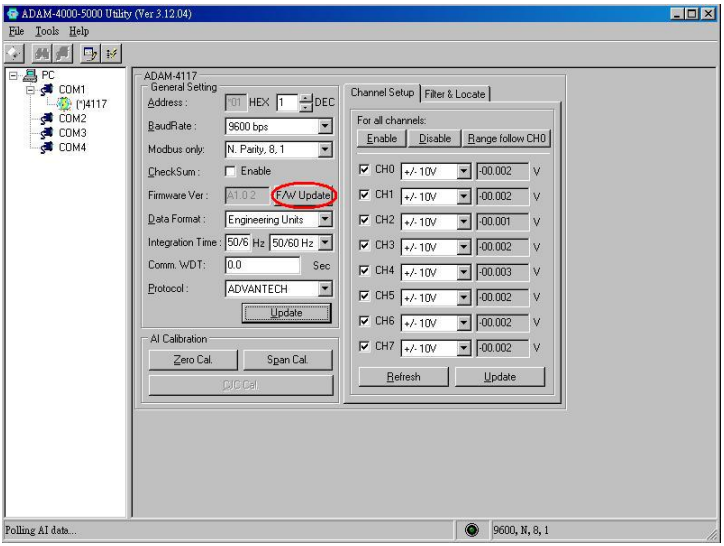


Figure A-5 Firmware update Function (a)

2. The two dialog windows hint to search again directly. It **doesn't** need to change ADAM status like initial or normal mode

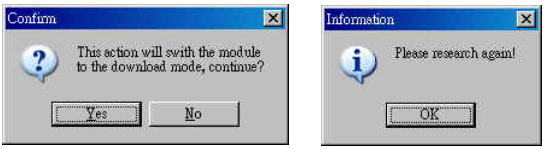


Figure A-6 Firmware update Function(b)

3. You will get a new sub window with firmware download option.
   The model is renamed 41XX, and the user can select the fast baud rate for download. Below the baud rate selection, you can choose the pathway for the firmware and download the file into the hardware.

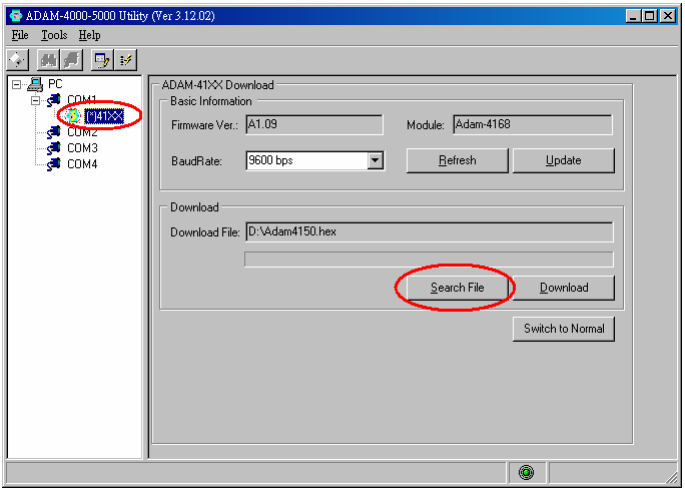


Figure A-7 Firmware update Function(c)

4. Once the download is successful, user can click on the button
   "Switch to Normal" for switching to normal mode.

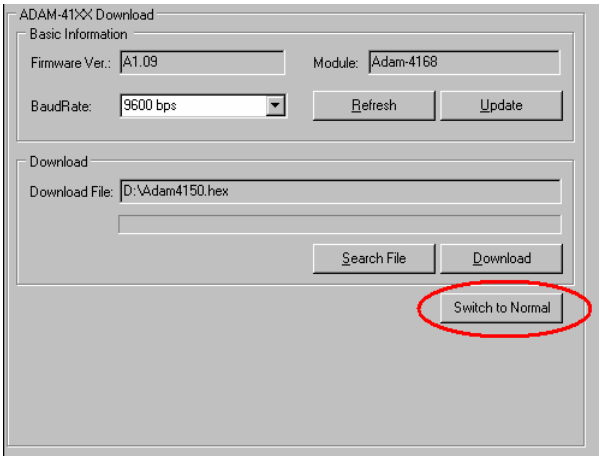


Figure A-8 Firmware update Function(d)

A.3 Address mode

The ADAM-4100 series not only has the original two status modes, but a new friendly mode is also added. It is called the **"address mode"**. The followings are the description for these three modes

Normal mode:
Once it is set to normal mode, the module will use the user defined settings during operation. A power reset will not alter these settings.

Initial mode:
Once it is set to initial mode, the module will use its factory settings. (Address 0 with data format 9600, N, 8, 1).

Note: If you need to change between Initial and Normal modes, the module needs to be reset before the changes will take effect.

Address mode:
For address mode, turn the switch directly from normal to initial mode without any power reset. The module will use the user defined settings during operation, and the LEDs will show the node ID as the following diagram below. These LEDs are common use for channel status and address mode.

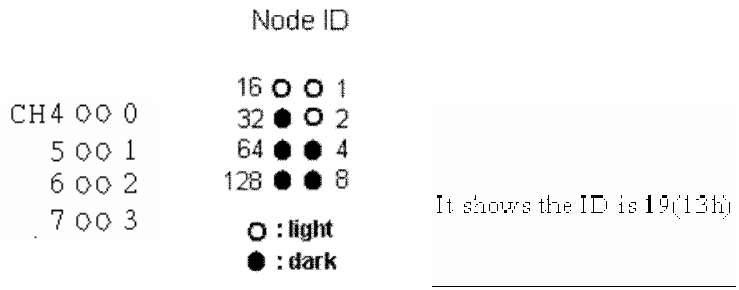


Figure A-9 Address mode Function

It shows that the ID is 19(13h). In the past, we can only use the utility for checking node ID. Now we can make use of the Address mode to help user read the module address directly.

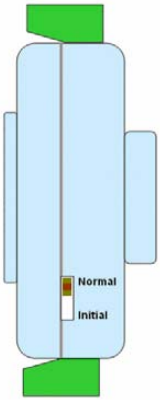
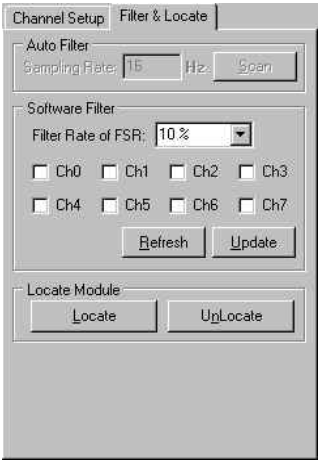


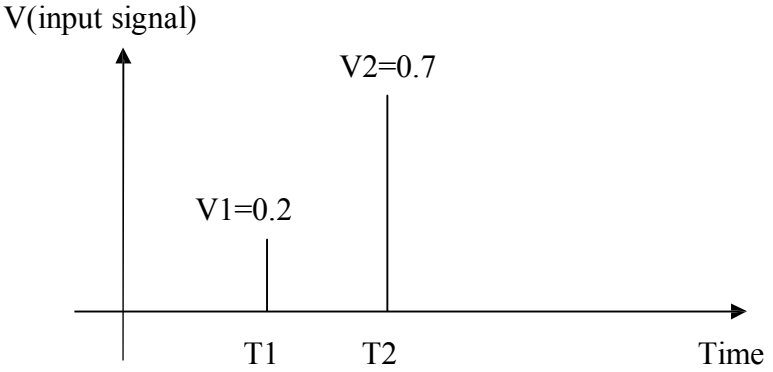
Figure A-10 Switch for Initial & Normal mode

A.4 Software Filter & Locate mode



1. Auto Filter: when Integration time is selected the auto-filter, it will auto scan major noise and filter it actively. Systems will response a proper setting. If it cannot find a proper setting, it will return 50/60Hz
   - 1.1 50/60 Hz: When system finds this base, the 50/60Hz will be filtered.
   - 1.2 100 Hz: When system finds this base, the 100Hz will be filtered.
   - 1.3 Auto Filter: When we set a fixed voltage, for example it is 5V. Maybe we can find a base which is 10Hz and filter it. But if we cannot find it, it will display 16Hz.
2. Software Filter: to ignore the sudden noise. Following illustration shows its concept.

The input signal of Channel 1



2.1 The condition of channel 1 is input range +/- 1V and filter 20%

2.2 20% of FSR(full scale range) means 0.4V

2.3 The time interval T1~T2 equals to sampling time

2.4 Conclusion is the difference between V1@ T1 and V2@T2 is 0.7-0.2 =0.5V > 0.4V, so signal V2 will be ignored

If you want to locate specific ADAM-4100 series, the configuration utility provides a "Locate" function to assist you. When you select a specific device, the LED that represents "Status" will be flashing for 8 minutes. If the "Locate" button is clicked, the "Status" LED will stay on. If the "UnLocate" is clicked, the "Status" will remain default status.

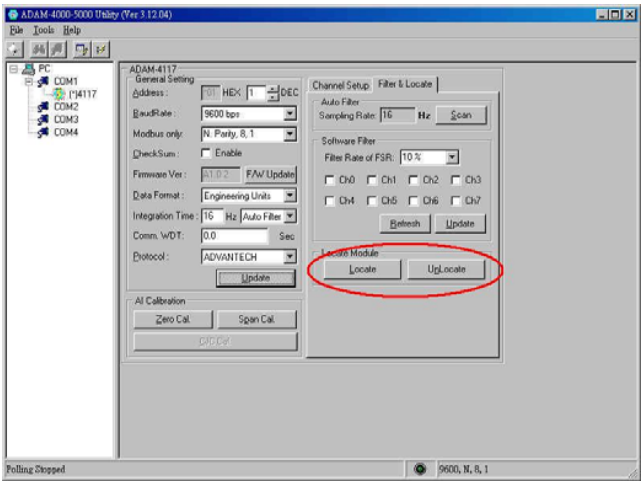


Figure A-11 Located mode Function

ADAM-4100 I/O Modbus Mapping Table

B

The model list of ADAM-4100 I/O series support Modbus protocol

	Model	Description
1	ADAM-4117	8-channel Analog Input Module
2	ADAM-4118	8-channel Thermocouple Input Module
3	ADAM-4150	Digital I/O Module
4	ADAM-4168	Relay Output Module

B.1 ADAM-4117 8-channel Analog Input Module

ADDR 4X	Channel	Item	Attribute	Memo
00201	0	Burn-out	R	
00202	1	Burn-out	R	
00203	2	Burn-out	R	
00204	3	Burn-out	R	
00205	4	Burn-out	R	
00206	5	Burn-out	R	
00207	6	Burn-out	R	
00208	7	Burn-out	R	
40001	0	Current Value	R	
40002	1	Current Value	R	
40003	2	Current Value	R	
40004	3	Current Value	R	
40005	4	Current Value	R	
40006	5	Current Value	R	
40007	6	Current Value	R	
40008	7	Current Value	R	
40201	0	Type Code	R/W	
40202	1	Type Code	R/W	
40203	2	Type Code	R/W	
40204	3	Type Code	R/W	
40205	4	Type Code	R/W	
40206	5	Type Code	R/W	
40207	6	Type Code	R/W	
40208	7	Type Code	R/W	
40211		Module Name 1	R	0x41 0x17
40212		Module Name 2	R	0x50 0x00
40213		Version 1	R	0xa2 0x00
40214		Version 2	R	0x00 0x00
40221		Channel Enable	R/W	0x00 0xff

B.2 ADAM-4118 8-channel Thermocouple Input Module

ADDR 4X	Channel	Item	Attribute	Memo
00201	0	Burn-out	R	
00202	1	Burn-out	R	
00203	2	Burn-out	R	
00204	3	Burn-out	R	
00205	4	Burn-out	R	
00206	5	Burn-out	R	
00207	6	Burn-out	R	
00208	7	Burn-out	R	
40001	0	Current Value	R	
40002	1	Current Value	R	
40003	2	Current Value	R	
40004	3	Current Value	R	
40005	4	Current Value	R	
40006	5	Current Value	R	
40007	6	Current Value	R	
40008	7	Current Value	R	
40201	0	Type Code	R/W	
40202	1	Type Code	R/W	
40203	2	Type Code	R/W	
40204	3	Type Code	R/W	
40205	4	Type Code	R/W	
40206	5	Type Code	R/W	
40207	6	Type Code	R/W	
40208	7	Type Code	R/W	
40211		Module Name 1	R	0x41 0x18
40212		Module Name 2	R	0x50 0x00
40213		Version 1	R	0xa2 0x00
40214		Version 2	R	0x00 0x00
40221		Channel Enable	R/W	0x00 0xff

B.3 ADAM-4150 Digital Input/Output Module

ADDR 0X	Channel	Item	Attribute	Memo
00001	0	DI Signal	R	
00002	1	DI Signal	R	
00003	2	DI Signal	R	
00004	3	DI Signal	R	
00005	4	DI Signal	R	
00006	5	DI Signal	R	
00007	6	DI Signal	R	
00017	0	DO Signal	W	
00018	1	DO Signal	W	
00019	2	DO Signal	W	
00020	3	DO Signal	W	
00021	4	DO Signal	W	
00022	5	DO Signal	W	
00023	6	DO Signal	W	
00024	7	DO Signal	W	
00033	0	Counter Mode: START(1)/STOP(0)	R/W	
00034	0	Counter Mode: Clear Counter(1)	R/W	
00035	0	Counter Mode: Clear Overflow	R/W	
00036	0	Counter Mode: Latch Status(read)/Clear Status(Write)	R/W	
00037	1	Counter Mode: START(1)/STOP(0)	R/W	
00038	1	Counter Mode: Clear Counter(1)	R/W	
00039	1	Counter Mode: Clear Overflow	R/W	
00040	1	Counter Mode: Latch Status(read)/Clear Status(Write)	R/W	
00041	2	Counter Mode: START(1)/STOP(0)	R/W	
00042	2	Counter Mode: Clear Counter(1)	R/W	

00043	2	Counter Mode: Clear Overflow	R/W	
00044	2	Counter Mode: Latch Status(read)/Clear Status(Write)	R/W	
00045	3	Counter Mode: START(1)/STOP(0)	R/W	
00046	3	Counter Mode: Clear Counter(1)	R/W	
00047	3	Counter Mode: Clear Overflow	R/W	
00048	3	Counter Mode: Latch Status(read)/Clear Status(Write)	R/W	
00049	4	Counter Mode: START(1)/STOP(0)	R/W	
00050	4	Counter Mode: Clear Counter(1)	R/W	
00051	4	Counter Mode: Clear Overflow	R/W	
00052	4	Counter Mode: Latch Status(read)/Clear Status(Write)	R/W	
00053	5	Counter Mode: START(1)/STOP(0)	R/W	
00054	5	Counter Mode: Clear Counter(1)	R/W	
00055	5	Counter Mode: Clear Overflow	R/W	
00056	5	Counter Mode: Latch Status(read)/Clear Status(Write)	R/W	
00057	6	Counter Mode: START(1)/STOP(0)	R/W	
00058	6	Counter Mode: Clear Counter(1)	R/W	
00059	6	Counter Mode: Clear Overflow	R/W	
00060	6	Counter Mode: Latch Status(read)/Clear Status(Write)	R/W	

00061	0	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00062	1	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00063	2	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00064	3	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00065	4	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00066	5	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00067	6	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00068	7	Pulse output Mode: Continue(1)/Non-Continue(0)	R	

ADDR 4X	Channel	Item	Attribute	Memo
40001~40014	0~6	For Counter/Frequency (7Channels)[32Bits] Frequency= ADDR/10Hz	R	
40015~40030	0~7	For Pulse Output L level time Unit:0.1ms 8 Channel[32Bits]	R/W	
40031~40046	0~7	For Pulse Output H level time Unit:0.1ms 8 Channel[32Bits]	R/W	
40047~40062	0~7	Set Absolute pulse(Set to 0=Continue mode) 8 Channel[32Bits]	R/W	
40063~40078	0~7	Set Incremental pulse 8 Channel[32Bits]	R/W	
40079~40085	0~6	Reference	R/W	
40086~40093	0~7	DO mode	R/W	
40094~40107	0~6	DI filter Low width	R/W	
40108~40121	0~6	DI filter High width	R/W	
40122~40137	0~7	DO Low Delay width	R/W	
40138~40155	0~7	DO High Delay width	R/W	
40211		Module Name 1	R	0x41 0x50
40212		Module Name 2	R	0x00 0x00
40213		Versoin 1	R	0xa2 0x00
40214		Versoin 2	R	0xB0 0x01
40215		Comm Safety Enable	R	Enable: 0x00 0x01
40216		Comm Safety Flag	R	Occur: 0x00 0x01
40301		DI data in word	R	
40302		Reserved		
40303		DO data in word	R/W	

Reference: II&0x07=00 DI Mode.
II&0x07=01 Counter Mode.
II&0x07=02 Low-->High Latch Mode.
II&0x07=03 High-->Low Latch Mode.
II&0x07=04 Frequency Mode.
II&0x20=20 DI Enable Counter record Function.
II&0x40=40 DI Enable Digital Filter Function.
II&0x80=80 DI Invert Mode

B.4 ADAM-4168 8 Relay Output Module

ADDR 0X	Channel	Item	Attribute	Memo
00017	0	Relay Output Value	R/W	
00018	1	Relay Output Value	R/W	
00019	2	Relay Output Value	R/W	
00020	3	Relay Output Value	R/W	
00021	4	Relay Output Value	R/W	
00022	5	Relay Output Value	R/W	
00023	6	Relay Output Value	R/W	
00024	7	Relay Output Value	R/W	
00033	0	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00034	1	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00035	2	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00036	3	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00037	4	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00038	5	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00039	6	Pulse output Mode: Continue(1)/Non-Continue(0)	R	
00040	7	Pulse output Mode: Continue(1)/Non-Continue(0)	R	

ADDR 0X	Channel	Item	Attribute	Memo
40001~40016	0~7	For Pulse Output L level time Unit:0.1ms 8 Channel[32Bits]	R/W	0xa2 0x00
40017~40032	0~7	For Pulse Output H level time Unit:0.1ms 8 Channel[32Bits]	R/W	0xB0 0x01
40033~40048	0~7	Set Absolute pulse(Set to 0=Continue mode) 8 Channel[32Bits]	R/W	Enable: 0x00 0x01
40049~40064	0~7	Set Incremental pulse 8 Channel[32Bits]	R/W	Occur: 0x00 0x01
40065~40072	0~7	DO mode	R/W	
40073~40088	0~7	DO Low Delay width	R/W	
40089~40104	0~7	DO High Delay width	R/W	
40211		Module Name 1	R	
40212		Module Name 2	R	
40213		Version 1	R	
40214		Version 2	R	
40215		Comm Safety Enable	R	
40216		Comm Safety Flag	R	
40301		Reserved		
40302		Reserved		
40303		DO data in word	R/W	

C

Technical Diagrams

C.1 ADAM Dimensions

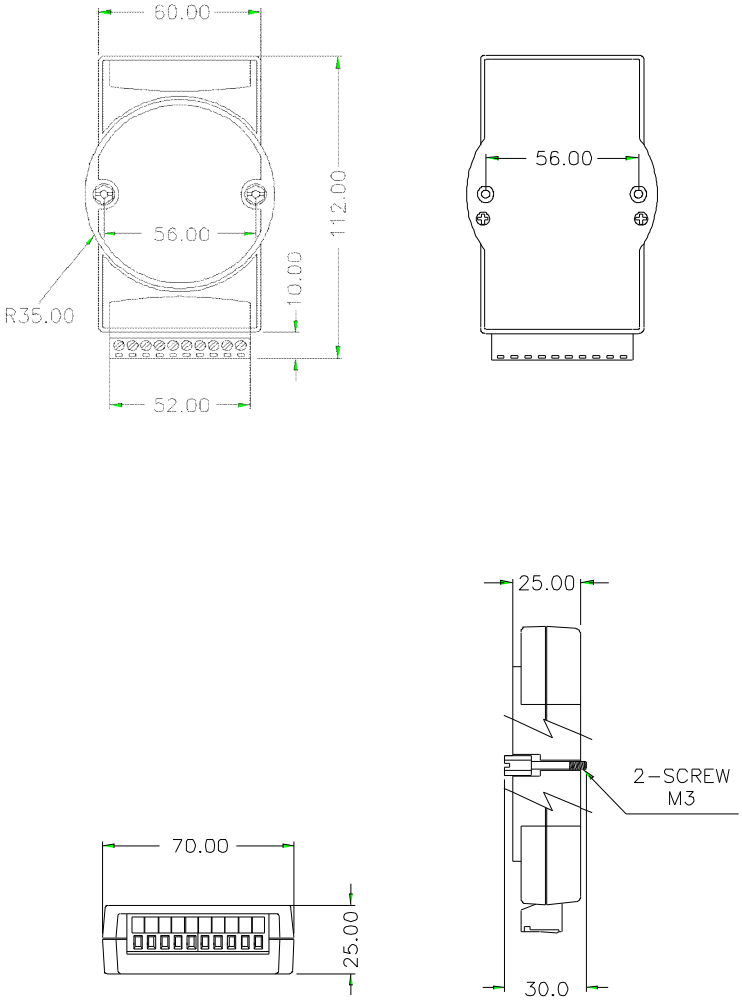


Figure C-1 *ADAM Modules Dimensions*

C.2 Installation

C.2.1 DIN-Rail Mounting

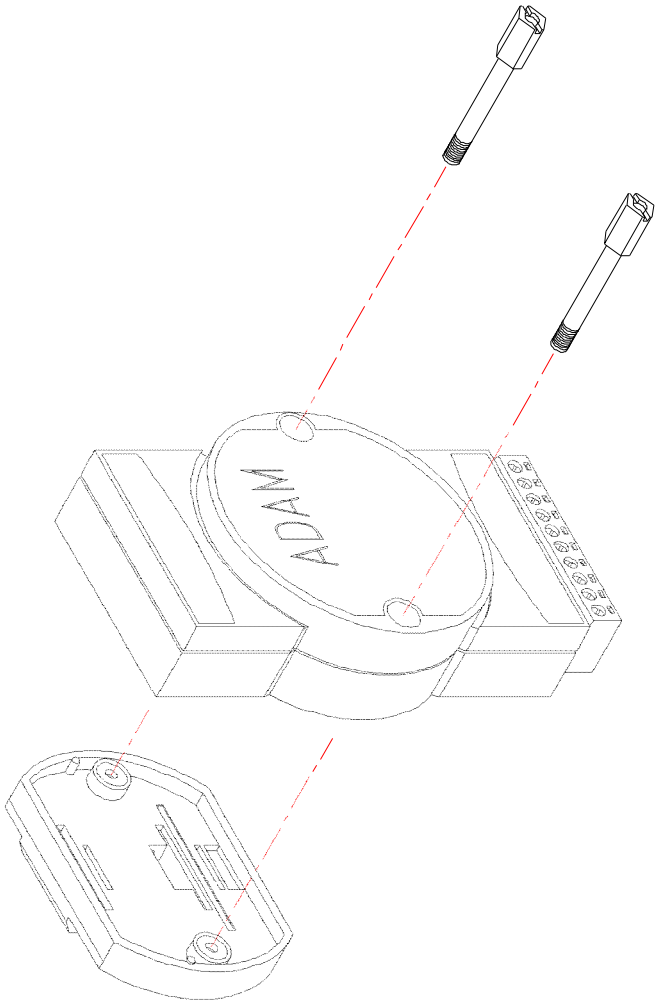


Figure C-2 *DIN-Rail Adapter*

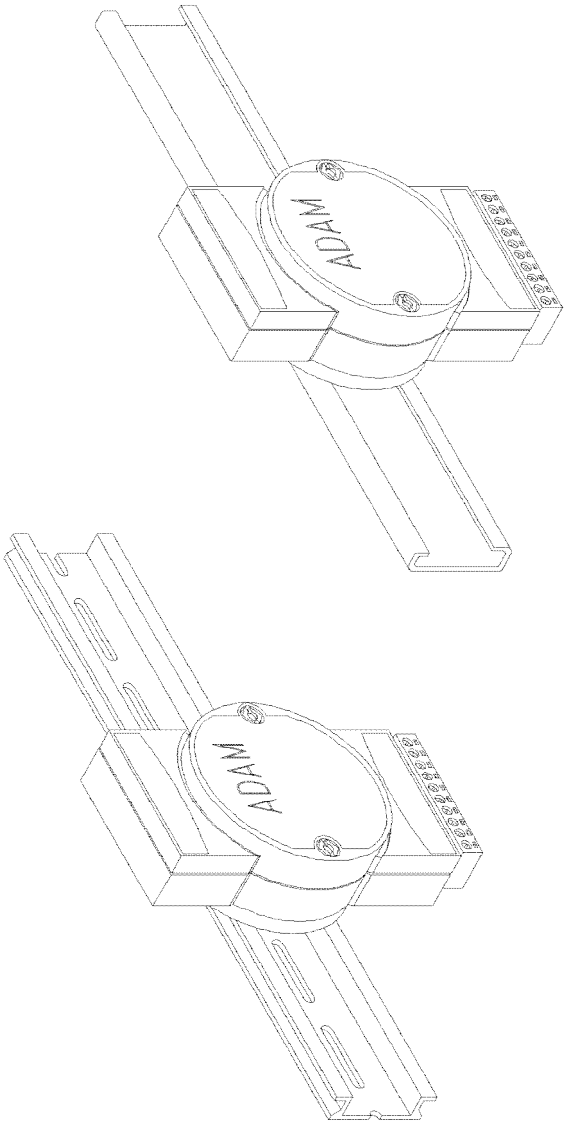


Figure C-3 *DIN-Rail Mounting*

C.2.2 Panel Mounting

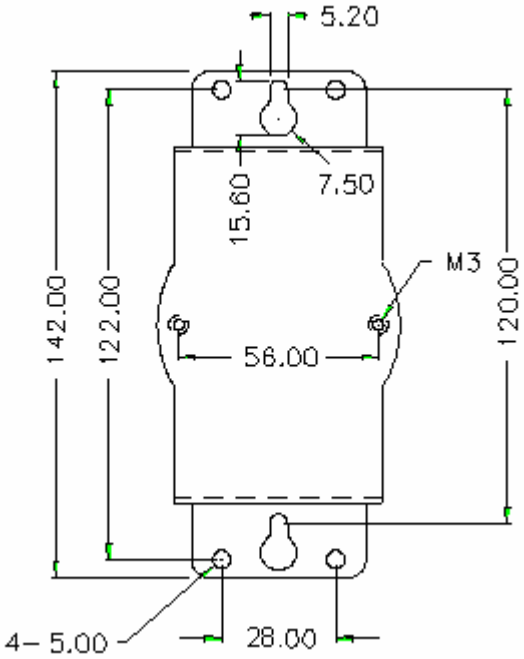


Figure C-4 *Panel Mounting Bracket Dimensions*

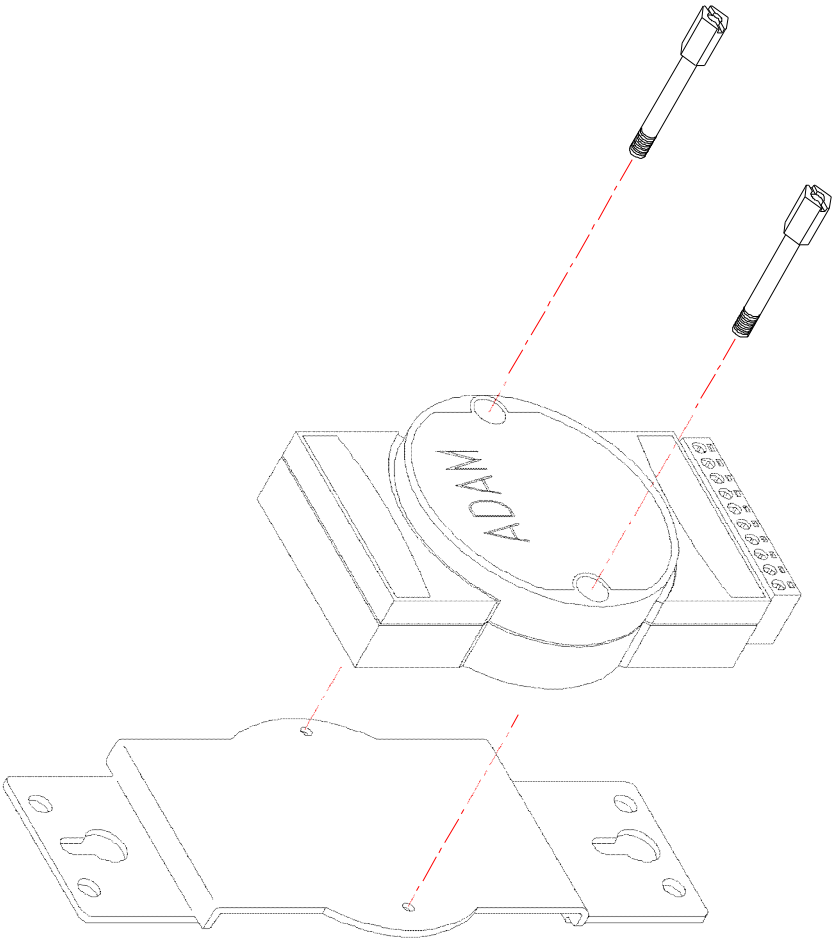


Figure C-5 *Panel Mounting*

C.2.3 Piggyback Stack

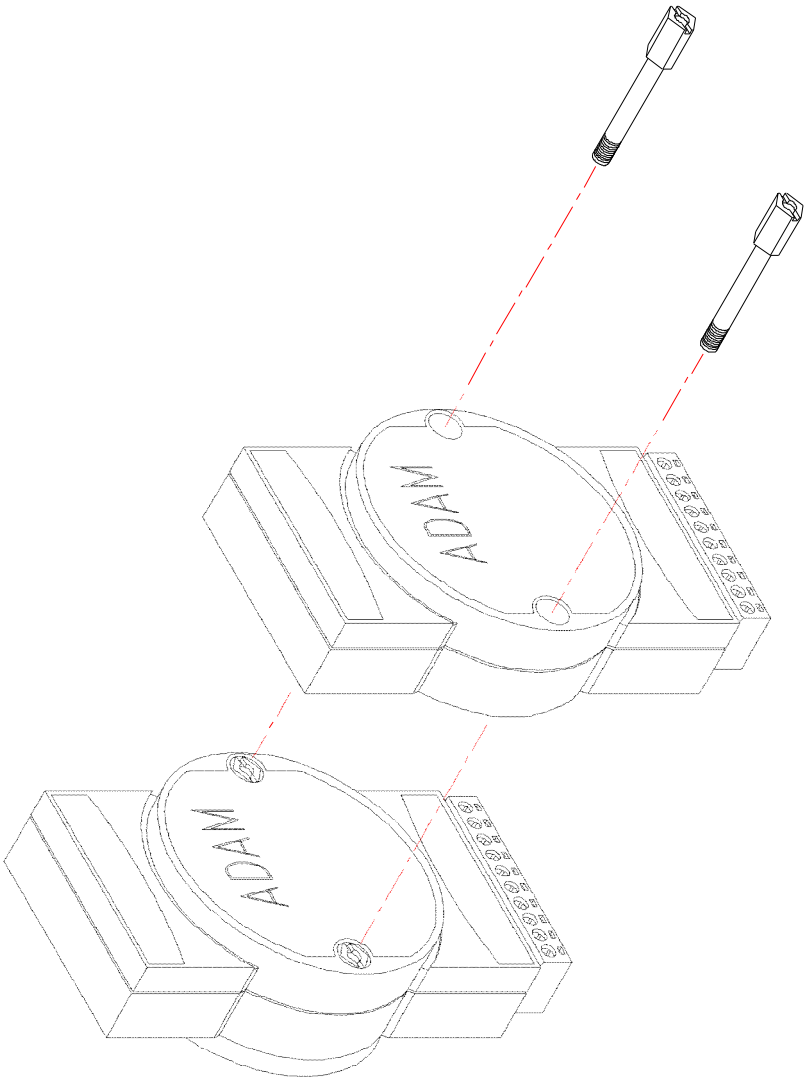


Figure C-6 *Piggyback Stack*

D

Data Formats and I/O Ranges

D.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

D.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 with value 00. This format presents data in standard units such as degrees, volts, mill volts and milliamps. When the value in engineering format is converted to computer language, it is presented in seven characters. These characters may include sign and decimals. However, the number of characters can not exceed seven.

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 examples:

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, and 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 unit format is used, the ADAM Series analog input modules are configured so that they automatically provide an over-range capability. The response to the Analog Data In command in this case is: +5.6530 (cr)

D.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 possible resolution is 0.01% with the decimal point fixed.

Data are given as the ratio of the input signal to 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 to (+(40/100) x 5 V = +2.0 V) the actual input value.

Example 2

The input value is 652.5°C, and a type E 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 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 we use 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 of the specified operation range for the thermocouple, zero will remain as 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. For instance, a nickel RTD is specified for -80°C to +100°C. Then, the lower value of -80°C equals to 0% of span and the upper value of +100°C equals to 100% of span.

In the FSR mode, an over-range feature is automatically invoked by the ADAM analog input modules if the value exceeds the uppermost value of the input range. For instance, an analog module which is configured for a ±5 V range has one of the values reading + 5.5V. The resulting value would then be 110%.

The readings must fall within the input range for accuracy assurance. Although they are typically linear readings, anything which falls between ±100% and ±115% limits may not be accurate. Furthermore, readings beyond these limits are neither accurate nor linear.

D.1.3 Twos complement hexadecimal

Twos Complement Hexadecimal format presents the data in ASCII hexadecimal form providing a rapid communication, high resolution and easy conversion to computer-compatible integer format.

In order 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).

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-4118 module is given a ±2.5 V input range. In this case, -2.5 V is represented as 8000h and +2.5 V is denoted as 7FFFh. When thermocouple input ranges are used, an input range which is bipolar and symmetric at 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

RS-485 Network

E

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 of 32 nodes per segment

-Bi-directional master-slave communication over a single set of twisted pair cables

-Parallel connected nodes, multi-drop capability

ADAM modules are fully isolated, and they use just a single set of twisted pair wires to send and receive! Since the nodes are connected in parallel, they can be disconnected from the host without affecting the performance of the remaining nodes. For industrial use, shielded twisted pair is preferred due to the high noise ratio of the environment.

When nodes communicate through the network, no conflicts during the transmission will occur since only 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 through 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 listening mode. The host computer initiates a command/response sequence with one of the modules. Commands normally contain the address of the module that the host wants to communicate with. The module will respond back to the host once a match is occurred between the module and the command.

E.1 Basic Network Layout

Multi-drop RS-485 implies that there are two main wires in a segment. The connected modules are connected by the so called drop cables, and all the connections are in parallel. As a result, connecting or disconnecting of a node doesn't affect the network as a whole. Since ADAM modules use the RS-485 standard with an ASCII-based commands set, they can connect and communicate with all the 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, and it is directly connected to the main-wires. Therefore, it acts as a medium which repeats the signals between two segments. However, there is a limitation towards this topology. It can only sustain up to 32 addressable modules. If more modules per segment are used, the IC driver current will rapidly decrease which may cause communication errors. Furthermore, the entire network can only hold up to 256 addressable modules because of the limitation of two numbered hexadecimal representation. The maximum representation of two numbered hexadecimal representation is 256. The ADAM converter, repeaters and the host computer are non addressable units; therefore, they are not included in these numbers.

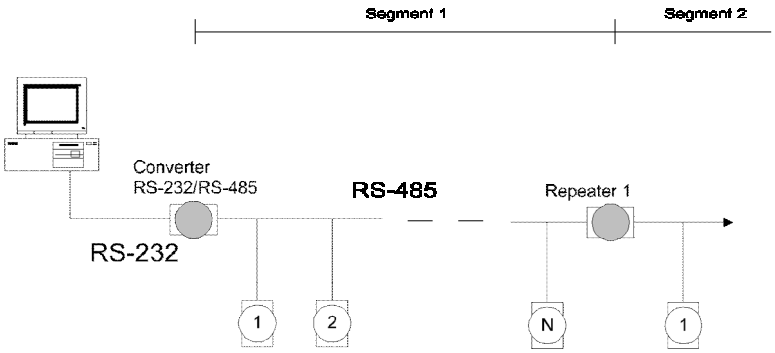


Figure E-1 *Daisychaining*

Star Layout

In this scheme, the repeaters are connected to drop-down cables from the main wires with modules connected after it. A tree structure is formed as the result. However, this scheme is not recommended when long lines are implemented since it will cause a serious amount of signal distortion due to a signal reflection at each end of the lines.

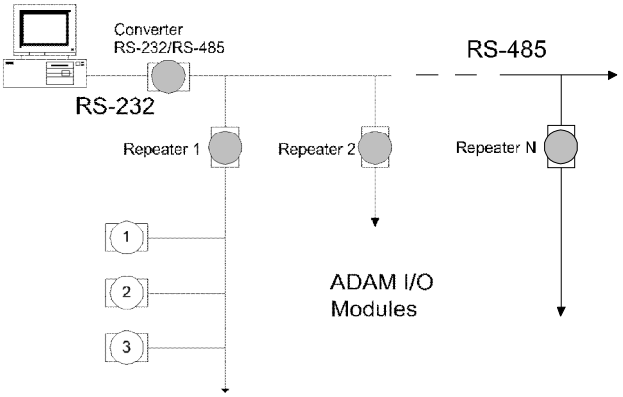


Figure E-2 *Star structure*

Random

This is a combination of daisychain and hierarchical structure

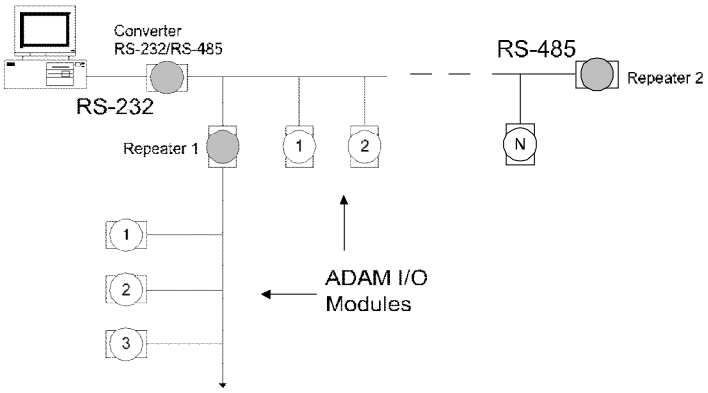


Figure E-3 *Random structure*

E. 2 Line Termination

Whenever the cable is long or the modules are different in the network, signal reflections are very likely to occur. As a result, the quality of the signals will be affected, and the signals will be distorted. In order to eliminate this problem, a resistor should be implemented at the beginning and the end of the cable.

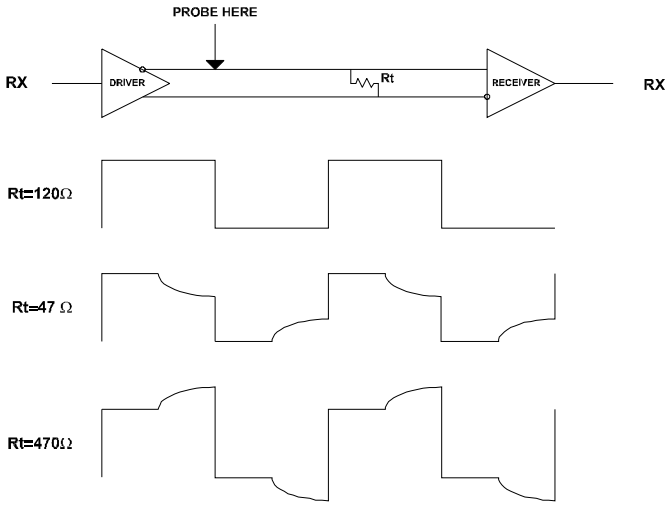


Figure E-4 *Signal Distortion*

The value of the resistor should be as close as possible to the characteristic impedance of the line. Although the receiving devices will add some resistance to the transmission line, having the resistor impedance equals to the characteristic impedance of the line should be sufficient enough.

Example:

Each input of the receivers has a nominal input impedance of 18 kW feeding into a diode transistor- resistor biasing network that is equivalent to an 18 kΩ input resistor tied to a common mode voltage of 2.4 V. It is this configuration which provides the large common range of the receiver required for RS-485 systems! (See Figure E-5 below).

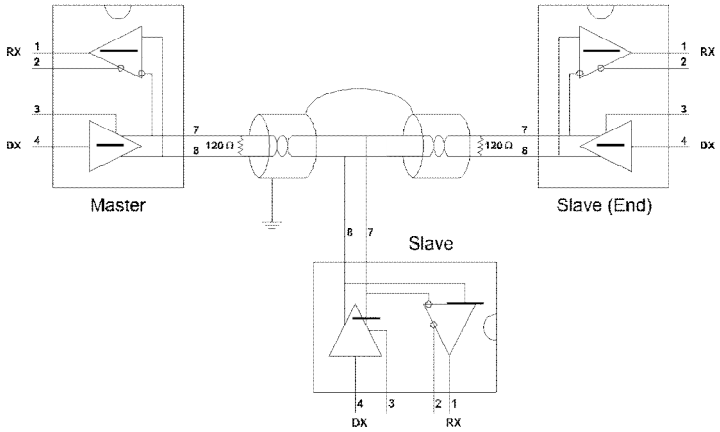


Figure E-5 *Termination resistor locations*

Because each input is biased to 2.4 V, the nominal common mode voltage of balanced RS-485 systems, the 18 kΩ on the input can be taken as being in series across the input of each individual receiver.

If thirty of these receivers are put closely together at the end of the transmission line, they will tend to react as thirty 36kΩ resistors in parallel with the termination resistor. The overall effective resistance will need to be close to the characteristics of the line.

The effective parallel receiver resistance RP will therefore be equal to:

R_P = 36 x 10 /30 = 1200 W

While the termination receiptor R_T will equal:

$R_T = R_O / [1 - R_O/R_P]$

Thus for a line with a characteristic impedance of 100 Ω resistor, the termination resistor R_T should be:

$R_T = [1 - 100/1200] = 110\ \Omega$

Since this value lies within 10% of the line characteristic impedance. Thus as already stated above the line termination resistor R_T will normally equal the characteristic impedance Z_O.

The star connection causes a multitude of these discontinuities since there are several transmission lines and is therefore not recommend.

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 wire to send and receive data. However, some controls to the direction of the data flow are required. RTS (Request to Sent) and CTS (Clear to Sent) are the most commonly used methods.

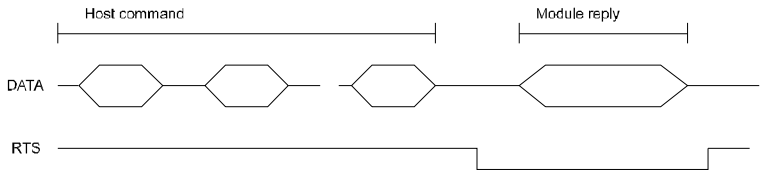


Figure E-6 *RS-485 data flow control with RTS*

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. Any software, which is written for half-duplex RS-232, is compatible with an ADAM network without modification required. The RS-485 control is completely transparent to the user.

F

How to use the Checksum feature

A checksum helps you detect communication errors between the host and module. This feature adds two extra checksum characters to the command or response string; therefore, it reduces the throughput.

F.1 Checksum Enable/Disable

In order to enable configuration of a module's checksum feature, its INIT* terminal should be shorted to its GND terminal. Then, the module should be rebooted. The checksum feature is enabled by setting bit 6 of the data format/checksum parameter to 1. On the other hand, the checksum is disabled by setting the parameter to 0. Whenever the checksum feature is used, all the connected devices including the host computer should be in enable mode.

The checksum is represented by a 2-character ASCII hexadecimal format and is transmitted just prior to the carriage return. The checksum equals to the result after performing modulus-256 (100h) of all the ASCII values' sum preceding the checksum. If the checksum is missing or incorrect, the module will not respond.

Example 1

The following example is an Analog Data In command and response when the checksum is enabled:

Command: #0588(CR)

Response: +3.56719D(CR)

The input value of the module at address 05h is +3.5671 V. (The data format is in engineering units.) The command checksum (88h) is the sum of the ASCII values for the following characters: #, 0, and 5. The response checksum (9Dh) is the sum of the ASCII values for 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) 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, and the result equals to 25h after modulus-256(100h) execution.

Table F-1 *Printable ASCII Characters*

HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII
21	!	40	@	5F	_	7E	~
22	""	41	A	60	'		
23	#	42	B	61	a		
24	$	43	C	62	b		
25	%	44	D	63	c		
26	&	45	E	64	d		
27	'	46	F	65	e		
28	(	47	G	66	f		
29	)	48	H	67	g		
2A	*	49	I	68	h		
2B	+	4A	J	69	i		
2C	,	4B	K	6A	j		
2D	-	4C	L	6B	k		
2E	.	4D	M	6C	l		
2F	/	4E	N	6D	m		
30	0	4F	O	6E	n		
31	1	50	P	6F	o		
32	2	51	Q	70	p		
33	3	52	R	71	q		
34	4	53	S	72	r		
35	5	54	T	73	s		
36	6	55	U	74	t		
37	7	56	V	75	u		
38	8	57	W	76	v		
39	9	58	X	77	w		
3A	:	59	Y	78	x		
3B	;	5A	Z	79	y		
3C	<	5B	[	7A	z		
3D	=	5C	\	7B	{		
3E	>	5D	]	7C	|		
3F	?	5E	^	7D	}		

Changing Configuration to Modbus Protocol G

Changing Configuration to Modbus Protocol

The ADAM-4100 Modbus version modules may come from the factory set for which ADAM ASCII protocol are set as the default protocol.

If the module is connected to a Modbus network, the Modbus network may not recognize the module. This may be caused by the incorrect settings. ADAM-4100 module should be set-up for Modbus protocol instead of ADAM ASCII protocol.

Please follow the steps as below for configuring an ADAM-4100 module to Modbus protocol.

1. Configure the ADAM-4100 Module with the ADAM-4000-5000 utility(latest utility can be found at www.advantech.com service & support.)
2. Initialize the ADAM-4100 on a RS-485 network (the preferred method is one module at a time on the RS-485 network).
3. With the module powered off, turn the switch in the "Init" position.
4. Power up the module
5. Wait 10 seconds for the module to initialize.
6. Using the ADAM-4000 utility, search (scan) for the module to change the protocol. (Initial COM settings: 9600 baud, N-8-1)
7. The utility will identify the module from the search function.
8. The ADAM-4000 utility will now permit the serial data protocol to be changed to the Modbus protocol.
9. The address and COM port settings can also be changed at this time.
10. To access the module, click on the module icon in the utility.
11. Update the settings by pressing the "Update" button.
12. Power off the module.
13. Turn the switch back to NORMAL* position.
14. The module is now ready to be placed in the Modbus network.