User Manual

ADAM-4100 Series

ADVANTECH

Enabling an Intelligent Planet

Copyright

Acknowledgements

Product Warranty (2 years)

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

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

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

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

1. Collect all the information about the problem encountered. (For example, CPU speed, Advantech products used, other hardware and software used, etc.) Note anything abnormal and list any on-screen messages you get when the problem occurs.
2. Call your dealer and describe the problem. Please have your manual, product, and any helpful information readily available.
3. If your product is diagnosed as defective, obtain an RMA (return merchandize authorization) number from your dealer. This allows us to process your return more quickly.
4. Carefully pack the defective product, a fully-completed Repair and Replacement Order Card and a photocopy proof of purchase date (such as your sales receipt) in a shippable container. A product returned without proof of the purchase date is not eligible for warranty service.
5. Write the RMA number visibly on the outside of the package and ship it prepaid to your dealer.

Part No. 2003M41003 Edition 4

Printed in Taiwan June 2021

Declaration of Conformity

CE

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

FCC Class A

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

Technical Support and Assistance

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

Safety Precaution - Static Electricity

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

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

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

Safety Instructions

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

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

Contents

Chapter 1

Introduction

1.1 Overview

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 and digital I/O functions. The LED display can also be utilized to locate modules in a network and even to directly read a module's address.

1.2 Modular Industrial Design

ADAM-4100 series modules have been designed to endure adverse conditions with their robust design. You can use them under special circumstances in a wide range of applications.

1.3 Suitability for Industrial Environments

Low Operating Temperature for Environment Monitoring

The ADAM-4100 series supports a broad operating temperature range of -40 to 85°C.

High Noise Immunity

To counteract noise that can occur in particular environments, ADAM-4100 modules provide additional protection for surge input, EFT, and ESD protection.

Wide Power Input Range

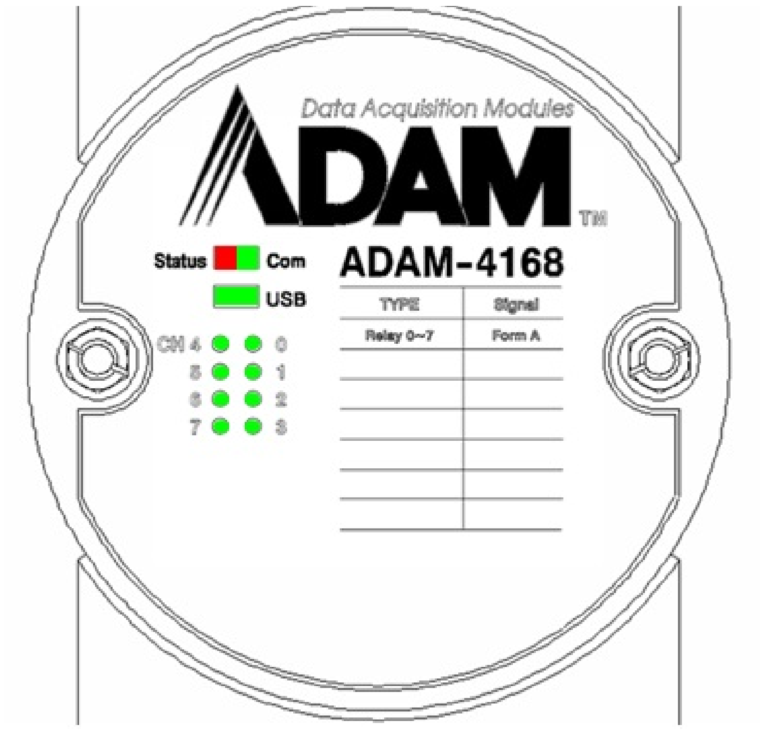
ADAM-4100 series modules accept any unregulated power source between +10 and +48 V_{DC}. They are also protected against accidental power supply reversals and can be safely connected or disconnected without disturbing a running network.

New Features for Individual I/O Modules

- Supports 200 V_{DC} high common mode voltage (ADAM-4117)
- Supports unipolar and bipolar inputs (ADAM-4117)
- Supports ±15 V input range (ADAM-4117)
- Supports filter auto-tuning or 50/60 Hz filter (ADAM-4117/4118)
- Digital filter function (ADAM-4150)
- Digital input channels can be used as a 3-kHz counter (ADAM-4150)
- Digital output channels support pulse output function (ADAM-4150/4168)

1.4 LED Display

ADAM-4100 series modules feature an LED display on the face. This allows you to monitor their status and read their address. In addition to the original two operating modes (Initial and Normal modes), these modules also feature a new mode called Address mode, which allows you to directly read a module's address via the LED display.



Note! *When the Status and Com LED blink at the same time, the LED will appear orange.*

LED	Color	Indication	Behavior
Status/Com	Red (status)	Blinking	Normal
		ON for 5 minutes	LOCATE module function
		Always ON	Download mode
	Green (Com)	Blinking	When data is transmitted via RS-485
USB	Green	Blinking	When data is transmitted via USB
I/O	Green	ON/ OFF	Logic 1/Logic 0 (Digital I/O, relay module)
			Channel enable/ disable (analog input module)

1.5 Firmware Online Update

ADAM-4100 series modules have a friendly and convenient design for you to update the firmware online. This can save a considerable amount of time and money when updating firmware.

1.6 Built-In Dual Watchdog Timer

A watchdog timer supervisory function will automatically reset an ADAM-4100 series module when required, which reduces the need for maintenance. This series of modules includes system and communication watchdog timers.

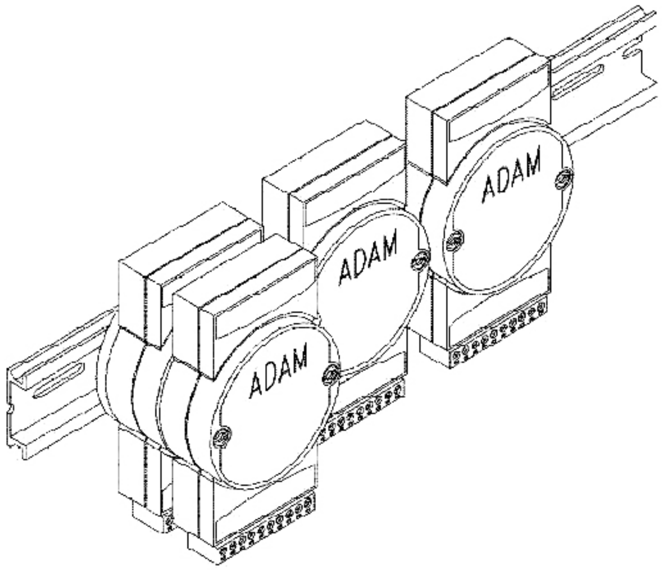
1.7 Dual Communication Protocol Support

ADAM-4100 modules support both the ADAM protocol and the Modbus/RTU protocol. You can select which communication mode you wish to use via Adam/Apax .NET Utility. If you apply the ADAM protocol, the ASCII command/response will remain the same as usual. In RTU mode, data are sent as two four-bit, hexadecimal characters, providing for higher throughput than ASCII mode would at the same baud rate. The ADAM-4100 series is a complete I/O solution, featuring Modbus network support in addition to the robust, intelligent design. They are easy to use and offer a cost-effective option for your I/O system needs.

1.8 RS-485 Support

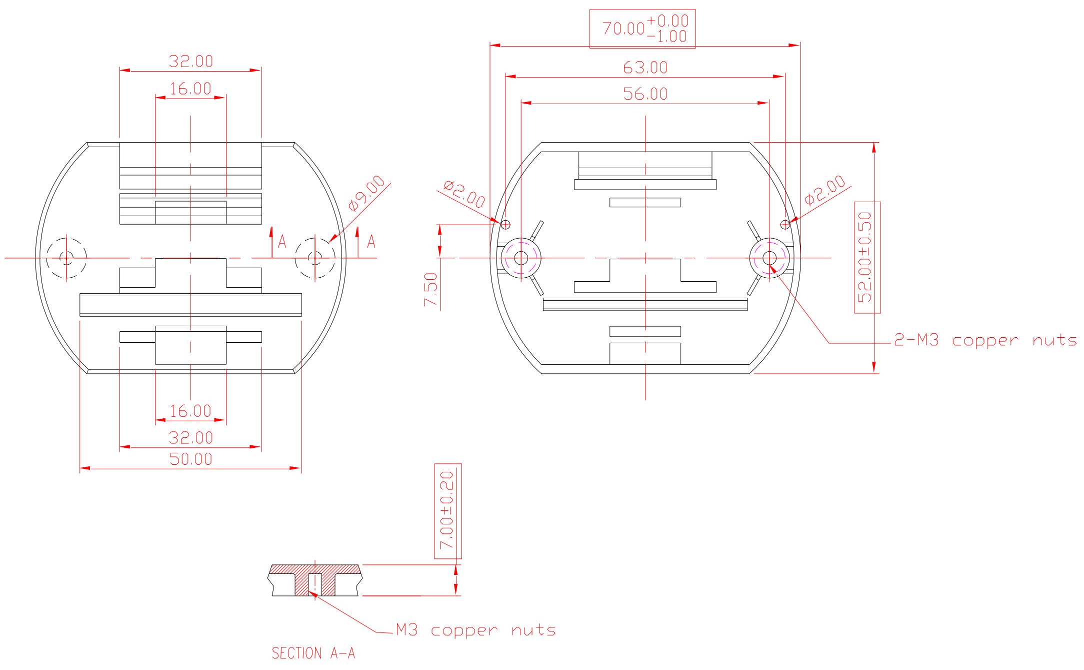
ADAM-4100 modules use the EIA RS-485 communication protocol, the industry's most widely used bidirectional, balanced transmission line standard. The EIA RS-485 standard was specifically developed for industrial applications.

1.9 Panel/DIN Rail Mounting Options



ADAM modules may be mounted on any panel, the provided brackets, DIN rails, or they may be piggybacked on top of each other. The RS-485 network, together with screw-terminal plug connectors, allows for system expansion, reconfiguration, and repair without disturbing the wiring.

Note! *The maximum number of piggybacked modules = 2*


32.00
16.00
ø9.00
A
A
16.00
32.00
50.00
70.00 +0.00 −1.00
63.00
56.00
ø2.00
ø2.00
7.50
52.00±0.50
2-M3 copper nuts
7.00±0.20
M3 copper nuts
SECTION A-A

Chapter 2

Installation Guide

This chapter provides guidelines on setting up and installing an ADAM network. A simple configuration scheme is provided to help you configure your modules before installing them in your network. To help you connect ADAM modules with sensor inputs, several wiring examples are provided. Finally, programming examples using the ADAM command set are given at the end of the chapter.

Be sure to plan the layout and configuration of your network carefully before you start. Guidelines for this can be found in Appendix E.

2.1 System Requirements for an ADAM Network

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

- ADAM modules.
- A host computer that can output ASCII characters with an RS-232C or RS-485 port.
- Power supply for the ADAM modules (+10 to +48 V_{DC}).
- Adam/Apax .NET Utility.
- 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 used 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

ADAM modules are designed to accept industry standard +24 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 the module connector. When modules are powered remotely, the effects of DC voltage drops must be considered.

All modules use onboard switching regulators to sustain efficiency over the 10~48 V input range; therefore, the actual drawn current can be assumed to be inversely proportional to the DC voltage.

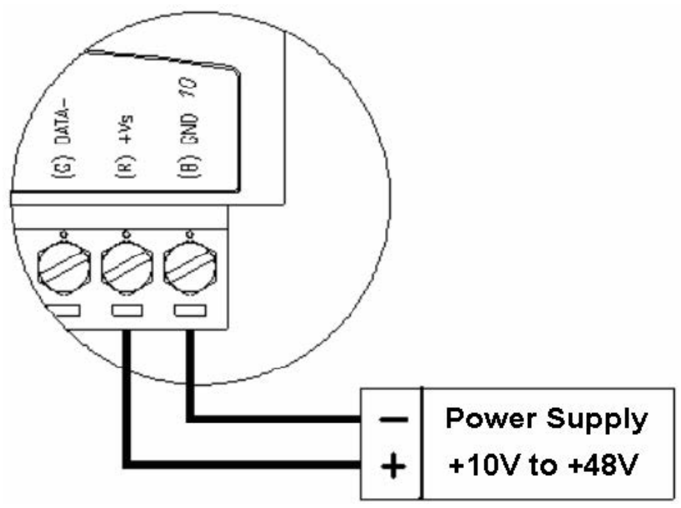


Figure 2.1 Power Supply Connections

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

+Vs	(R)	Red
GND	(B)	Black

Communication Wiring for RS-485

We recommend using shielded twisted-pair cables in ADAM networks in order to reduce interference; however, such cables must comply with the EIA RS-485 standard. Only one set of twisted-pair cables is required for data transmission. We advise the following standard colors (as indicated on the modules) for each the communication line:

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

Communication Wiring for Micro USB

Since the B versions of ADAM-4100 series modules support the micro USB interface, this can be used as a power source and communication interface.

Adam/Apax .NET Utility

Adam/Apax .NET Utility is a menu-driven utility program for ADAM module configuration, monitoring, and calibration. It also includes a terminal emulation program that lets you communicate using the ADAM command set (see Appendix A).

ADAM Communication Speed

The baud rate of ADAM-4100 series modules can be configured from 1200 bps to 115.2 Kbps. However, the baud rate of all modules in an RS-485 network must be identical.

ADAM Isolated RS-232/RS485 Converter (Optional)

When the host computer or terminal only has an 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. (1,200 m) or when more than 32 ADAM modules are connected, a repeater should be implemented. In a network, up to eight repeater modules can be connected, allowing for a maximum of 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 Network Setup

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

2.2.1 Default Factory Settings

Baud rate	9600 bit/s
Address	01h

A basic module network configuration is shown in Figure 2.2.

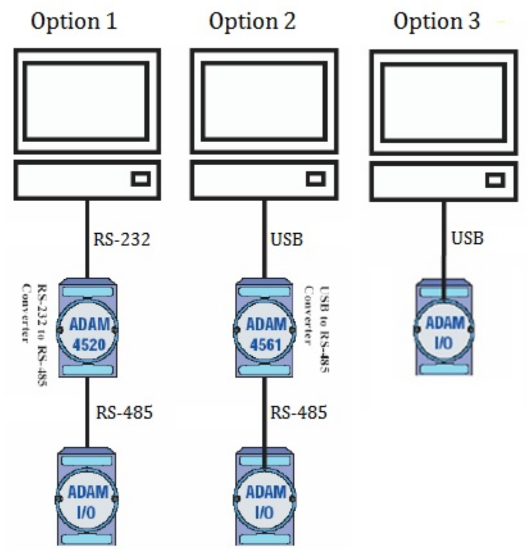


Figure 2.2 Connecting ADAM Modules to Host Switches

The following items are required to configure a module:

- ADAM converter module
- PC with an RS-232 port (baud rate set to 9600)
- Adam/Apax .NET Utility

2.2.2 Grounding Protection

Grounding is one of the most important issues for ADAM systems. Just like the frame ground of a computer, this signal offers a reference point for the electric circuit inside the computer. If we want to communicate with this computer, both the signal ground and frame ground should be connected to make a reference point of each other's electric circuit. Generally, it is necessary to install an individual grounding bar for each system, such as computer networks, power systems, and telecommunication networks. Those individual grounding bars not only provide an individual reference point, but they also make the earth a ground.

The ADAM-4100 series of modules provides flexible grounding options, in that you can use the ground connector on the right side of the module to ground the unit, as shown in the following image:

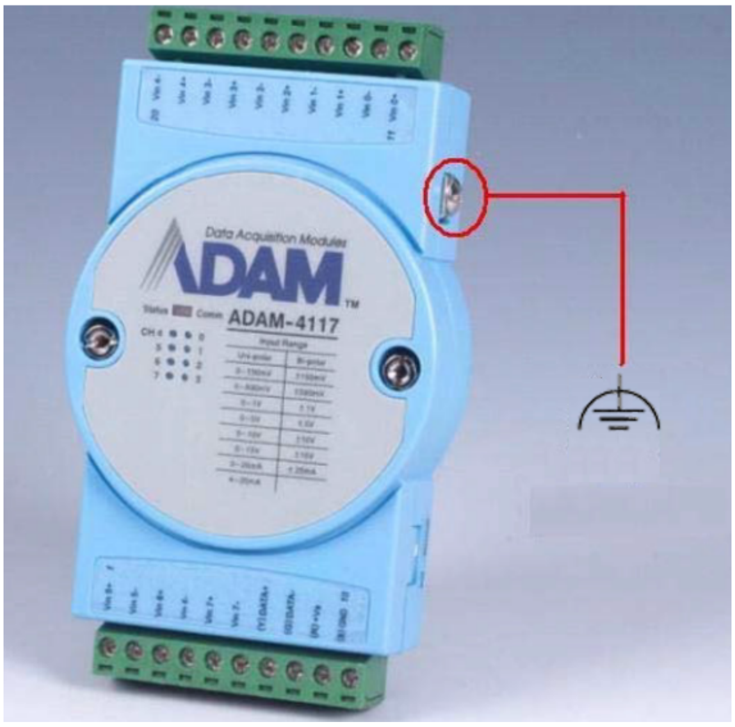


Figure 2.3 Ground Connector on an ADAM-4117

2.2.3 Module Configuration with Adam/Apax .NET Utility

The easiest way to configure ADAM modules is by /Apax .NET Utility. It is a user-friendly structured program that will guide you through every step of the configuration (see Appendix A)

2.2.4 Changing the Protocol from ADAM ASCII to Modbus

Some ADAM-4100 modules support both ADAM ASCII and Modbus protocols. By default, these modules are set to the ADAM ASCII protocol. Refer to Appendix G for information on switching your modules to the Modbus protocol.

2.2.5 Module 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 Adam/Apax .NET Utility (refer to Chapter 4).

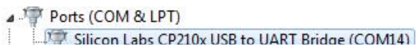
2.2.6 Communication via Micro USB

The USB interface has become common in IoT devices. Furthermore, it makes access by laptop or PC easy. To expand the accessibility of ADAM-4100 series modules, other than the RS-485 serial port, ADAM-4100 series modules (B version) feature a micro USB interface. Modules can be powered on and communicate through this interface. When a module is communicating via RS-485, the USB can also be used at the same time for communication. You can choose either the micro USB or RS-485 interface to access the modules.

Note! *The baud rate of micro USB and RS-485 will be identical for the modules; this means that if the RS-485 or USB baud rate is changed, the other one will automatically align to the same baud rate.*

Micro USB Driver Installation

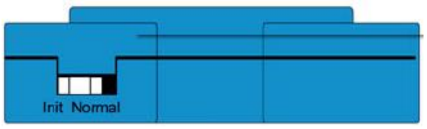
You can access ADAM modules by /Apax .NET Utility. Before using the micro USB interface, you must install the "CP210x" chip driver on your computer. The driver can be downloaded from the Advantech website. After installation, you will see the virtual COM port in the management system.


Ports (COM & LPT)
Silicon Labs CP210x USB to UART Bridge (COM14)

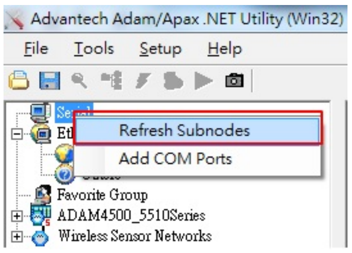
The ADAM-4100 micro USB interface was developed based on the standard COM port, which makes it easy to be integrated in SCADA with the standard COM port driver. Unlike other USB devices, extra effort does not need to be invested in USB driver development for your SCADA system.

Module Configuration /Apax .NET Utility

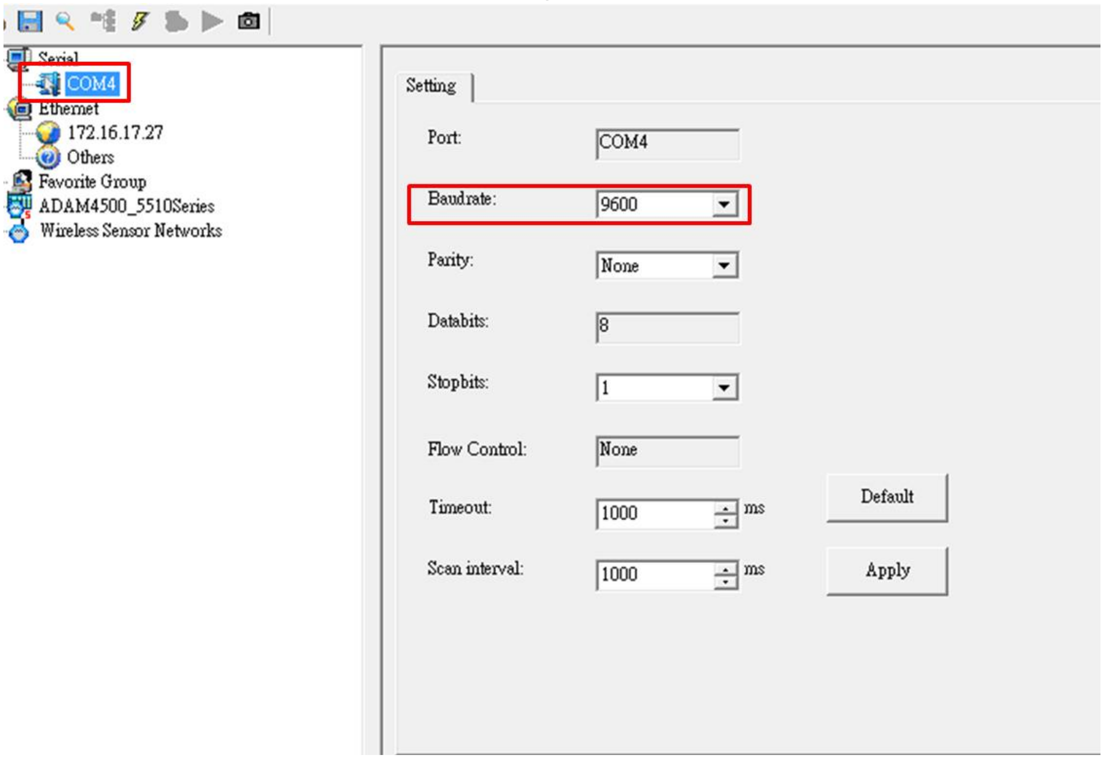
1. Turn the switch to **Init** mode



2. Connect the module to your computer via a USB cable. Open Adam/Apax .NET Utility and click **Refresh Subnodes** on the serial icon in the Module Tree Display Area.

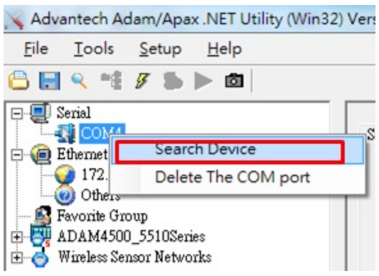


3. The COM port for the module will be shown in the utility; select it and change the baud rate to **9600** and click **Apply**

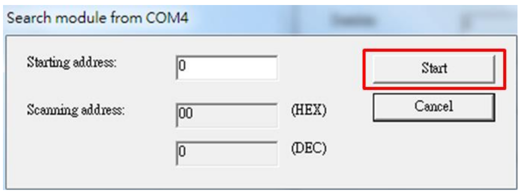


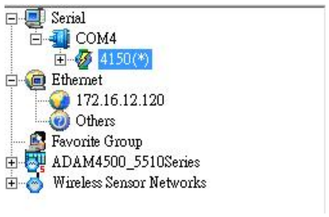
Port: COM4
Baudrate: 9600
Parity: None
Databits: 8
Stopbits: 1
Flow Control: None
Timeout: 1000 ms
Scan interval: 1000 ms

4. Right-click on the COM port and click **Search Device**



5. Click **Start** to search for the module. After the searching process is finished, the model name of the module will be displayed on the list. You can then start to configure the module.

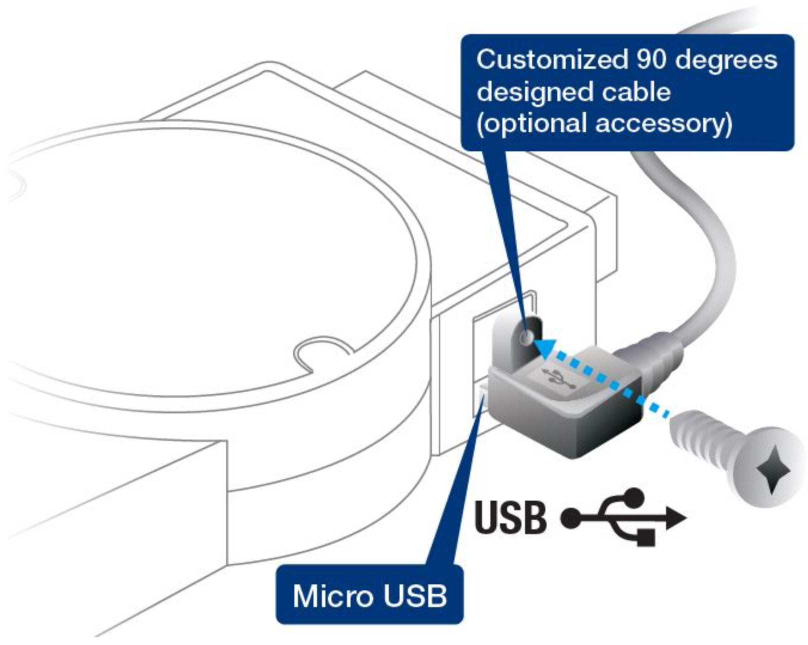




6. After configuration, power off the module and turn the switch to **Normal** mode. The setting will be applied when you power on the module.

Micro USB Connection

The ADAM micro USB interface can be adapted to a standard micro USB cable. Advantech also provides a 90° micro USB to type-A USB cable with a fixed hole (the cable is optional accessory 96PD-YH3874) to enhance the connection stability.



2.2.7 Baud Rate and Checksum

ADAM modules contain EEPROMs that store configuration information and calibration constants. The EEPROM replaces the conventional array of switches and pots that were originally used for specifying the baud rate, I/O range, etc.

Since there is no visual indication of a module's configuration status, it is impossible to know the baud rate, address, or 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	**Default**
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 the Baud Rate and Checksum

Baud rate and checksum settings have several things in common:

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

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

1. Power on all components except the ADAM module
2. Power the ADAM module on while turning the switch to "initial" (see Figure 2-3)
3. Wait at least 7s to let the self-calibration and ranging take effect
4. Configure the checksum status and/or baud rate
5. Switch the ADAM module power OFF
6. Turn the switch to Normal mode and power the module on
7. Wait at least 7s for the self-calibration and ranging to take effect
8. Check the settings (if the baud rate has changed, the settings on the host computer should be changed accordingly)

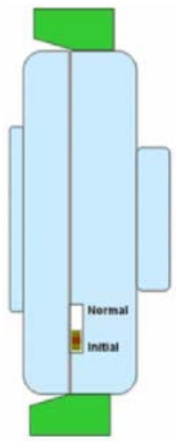


Figure 2.4 Module Switch Set to Initial Mode

2.2.8 Connecting Multiple Modules

Figure 2.4 provides an example of how ADAM modules are connected in a multi-module network:

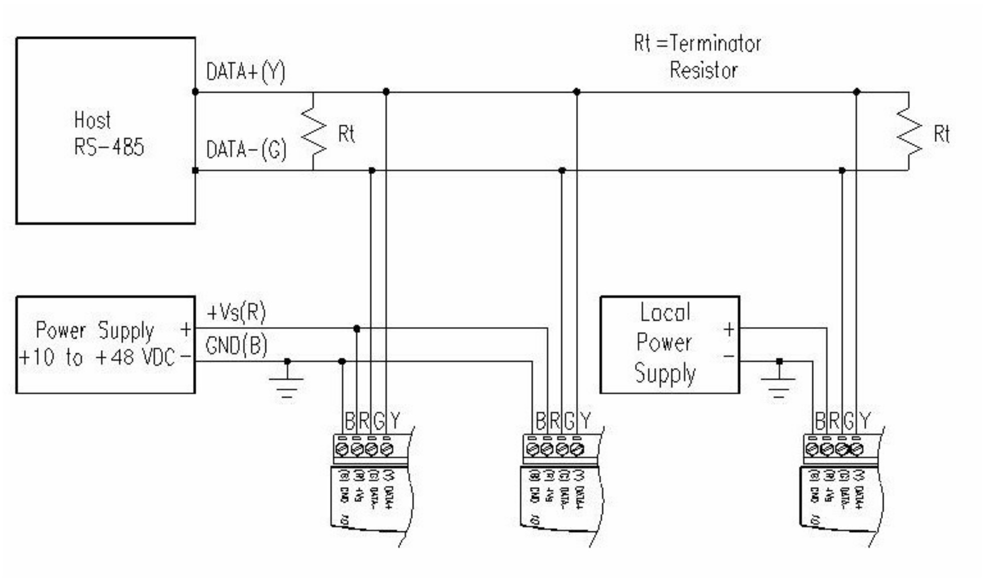


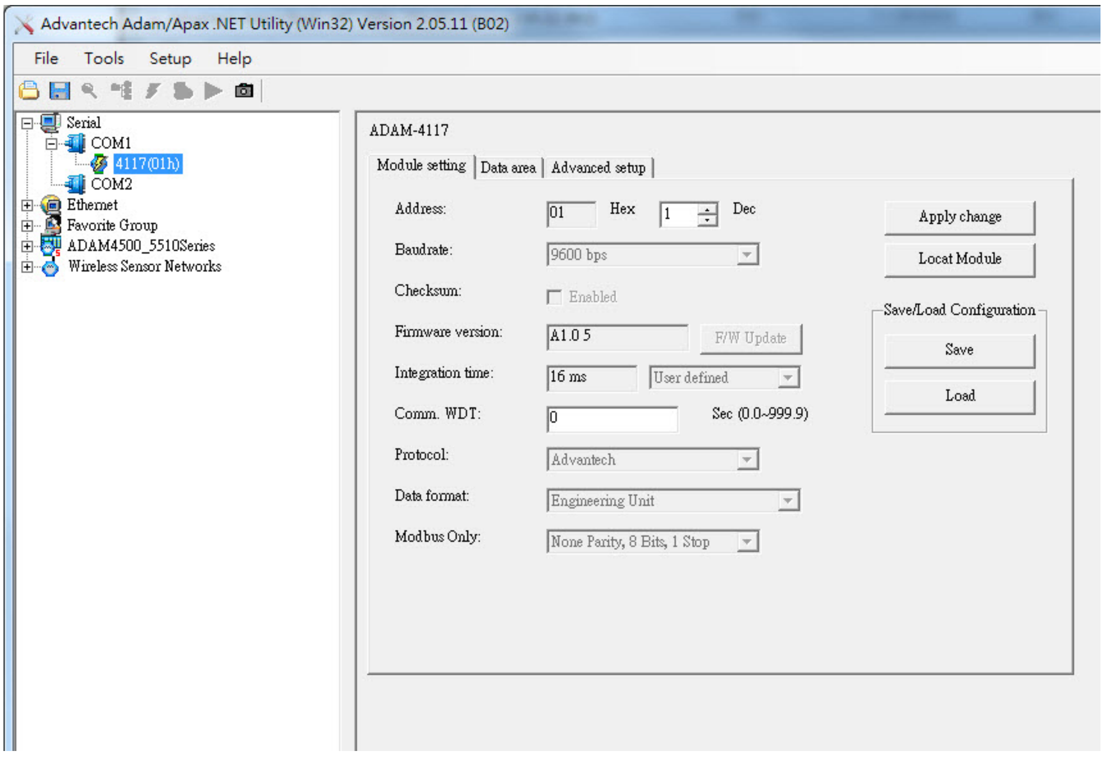
Figure 2.5 Multi-Module Network Configuration

2.3 Programming Example

The following example is a simple program written in Visual Basic 6.0. It demonstrates how to get a temperature reading stored in Address 01H of an ADAM-4117 module.

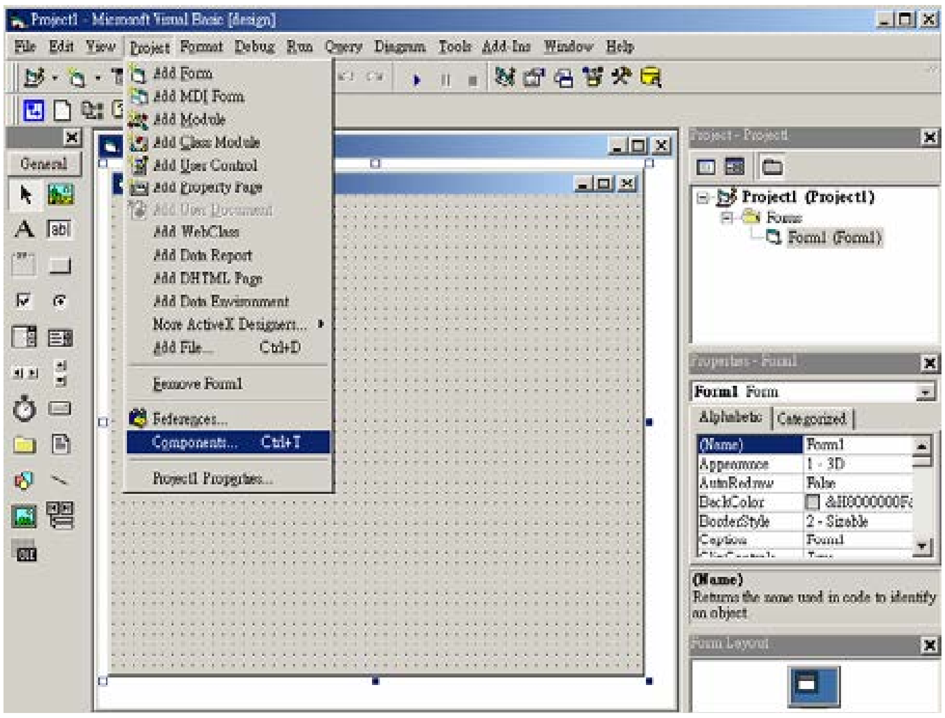
1. Use Adam/Apax .NET Utility to check the settings, as follows:

- **Address** = "01H"
- **Baud rate** = "9600"
- **Checksum** = "Disabled" (unchecked)

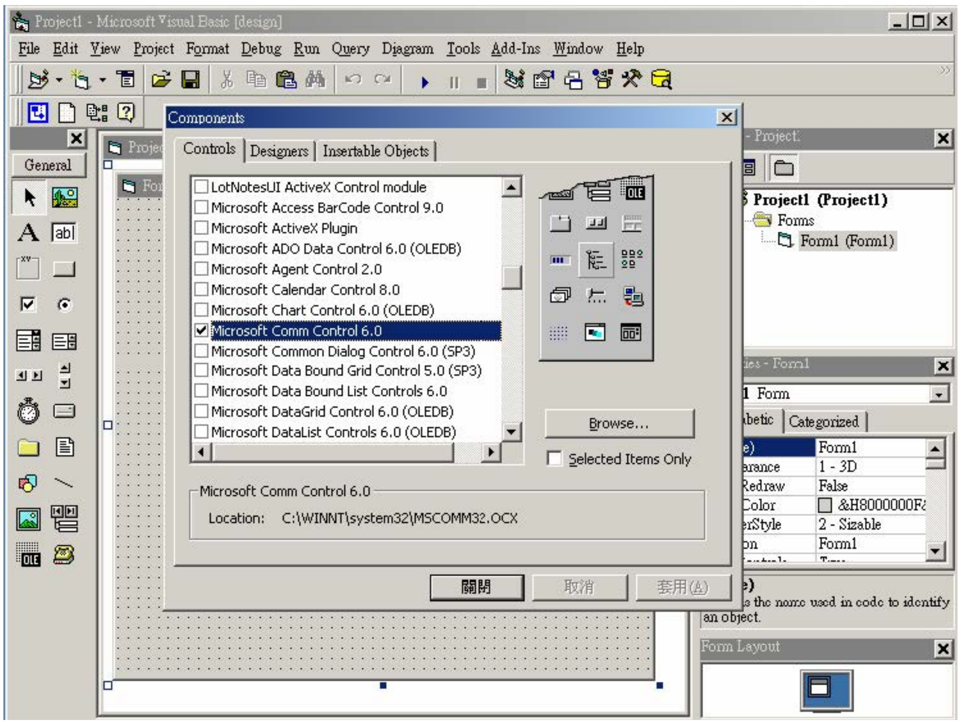

Advantech Adam/Apax .NET Utility (Win32) Version 2.05.11 (B02)

Address: 01 Hex 1 Dec
Baudrate: 9600 bps
Checksum: ☐ Enabled
Firmware version: A1.0 5
Integration time: 16 ms User defined
Comm. WDT: 0 Sec (0.0~999.9)
Protocol: Advantech
Data format: Engineering Unit
Modbus Only: None Parity, 8 Bits, 1 Stop

2. Run VB 6.0 and add a control by selecting **Component** under the **Project** menu



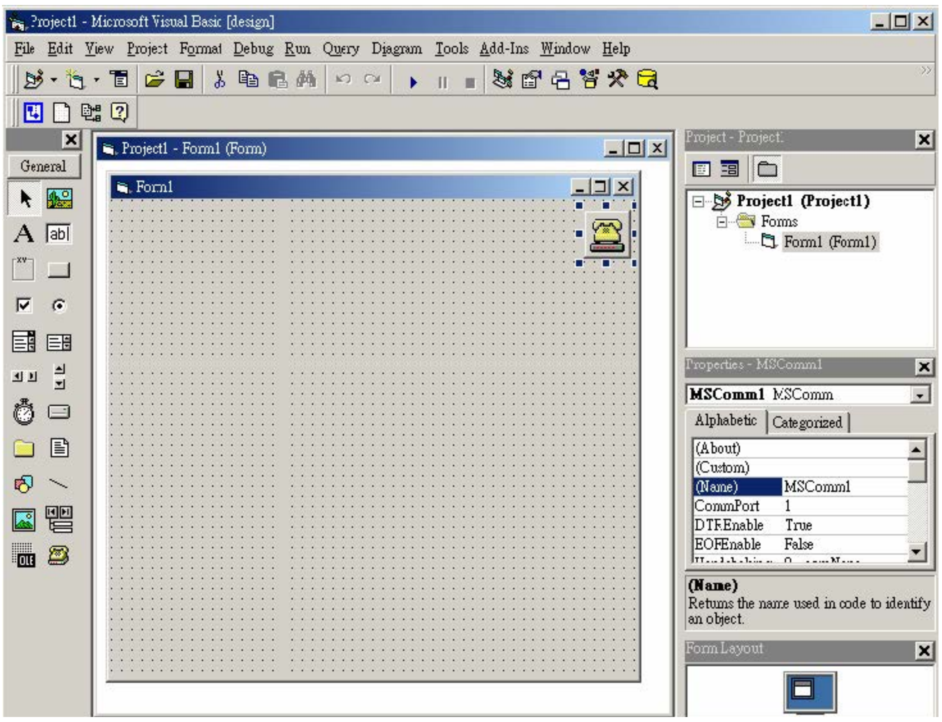
3. Select **Microsoft Comm Control**



LotNotesUI ActiveX Control module
Microsoft Access BarCode Control 9.0
Microsoft ActiveX Plugin
Microsoft ADO Data Control 6.0 (OLEDB)
Microsoft Agent Control 2.0
Microsoft Calendar Control 8.0
Microsoft Chart Control 6.0 (OLEDB)
[✓ Microsoft Comm Control 6.0]
Microsoft Common Dialog Control 6.0 (SP3)
Microsoft Data Bound Grid Control 5.0 (SP3)
Microsoft Data Bound List Controls 6.0
Microsoft DataGrid Control 6.0 (OLEDB)
Microsoft DataList Controls 6.0 (OLEDB)

Microsoft Comm Control 6.0
Location: C:\WINNT\system32\MSCOMM32.OCX

4. Add the comm control to the form

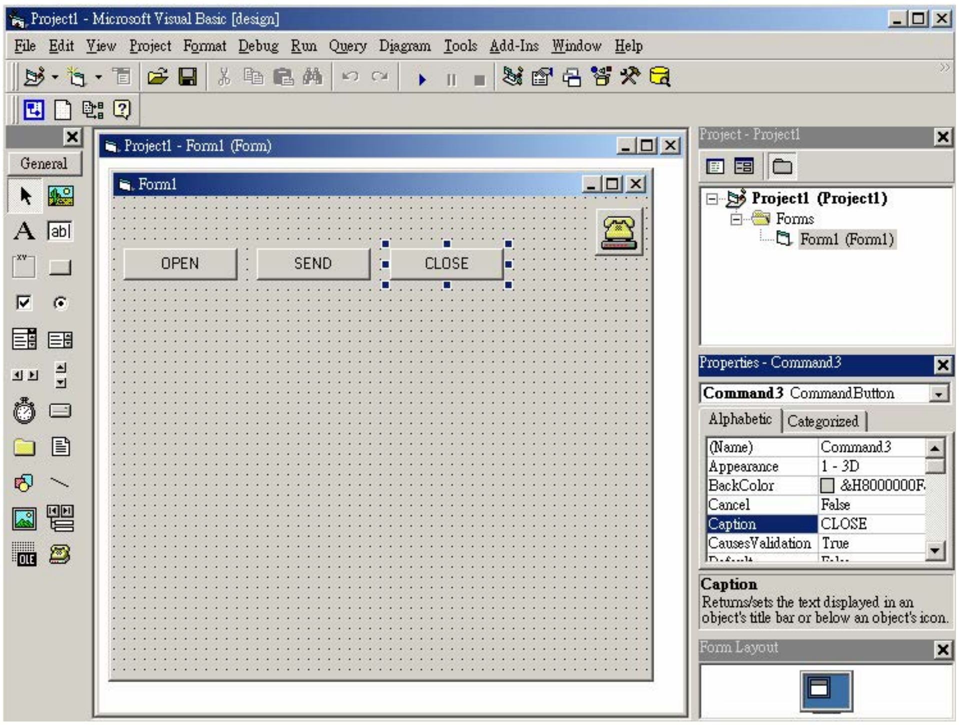


MSComm1 MSComm

(About)	
(Custom)	
(Name)	MSComm1
CommPort	1
DTREnable	True
EOFEnable	False

(Name)
Returns the name used in code to identify an object.

5. Add three command buttons to the form, as follows:

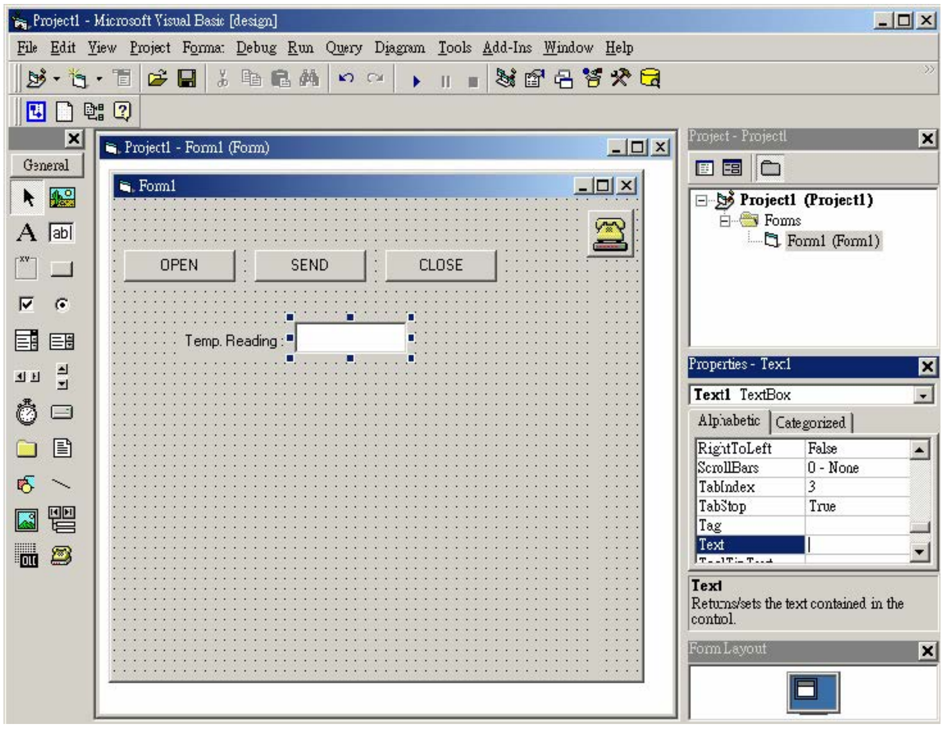


Command3 CommandButton

(Name)	Command3
Appearance	1 - 3D
BackColor	&H8000000F
Cancel	False
Caption	CLOSE
CausesValidation	True

Caption
Returns/sets the text displayed in an object's title bar or below an object's icon.

6. Add one label and one text box to the form, as follows:

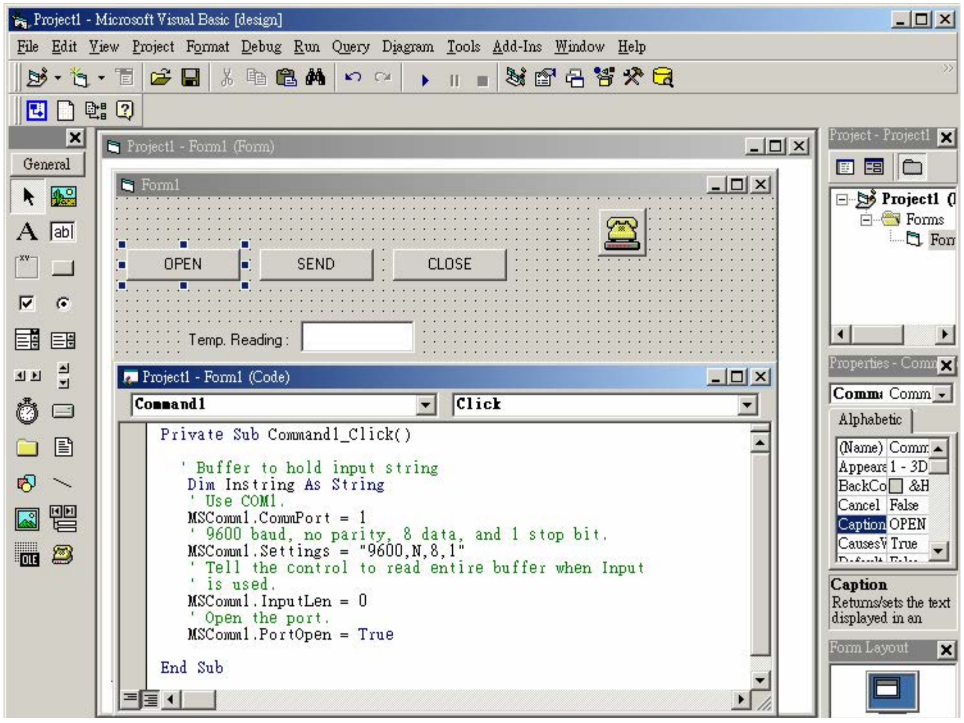


Text1 TextBox

RightToLeft	False
ScrollBars	0 - None
TabIndex	3
TabStop	True
Tag	
Text	

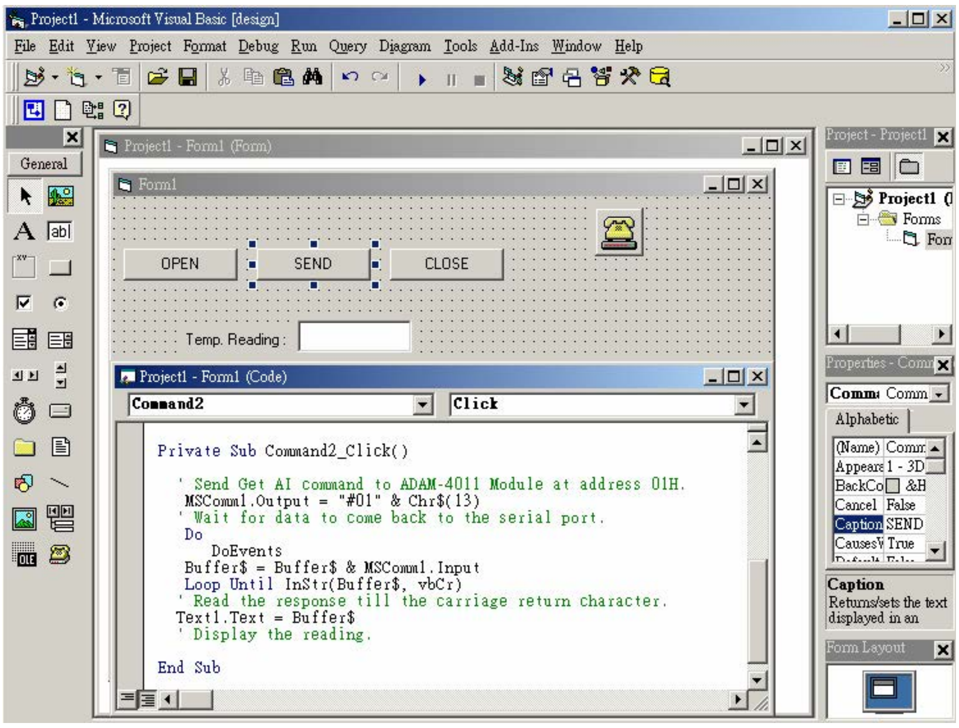
Text
Returns/sets the text contained in the control.

7. Click **OPEN** and enter the following code (source code examples are listed at the end of this section).



```
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
```

8. Click **SEND** and ensure the following code (source code examples are listed at the end of this section).

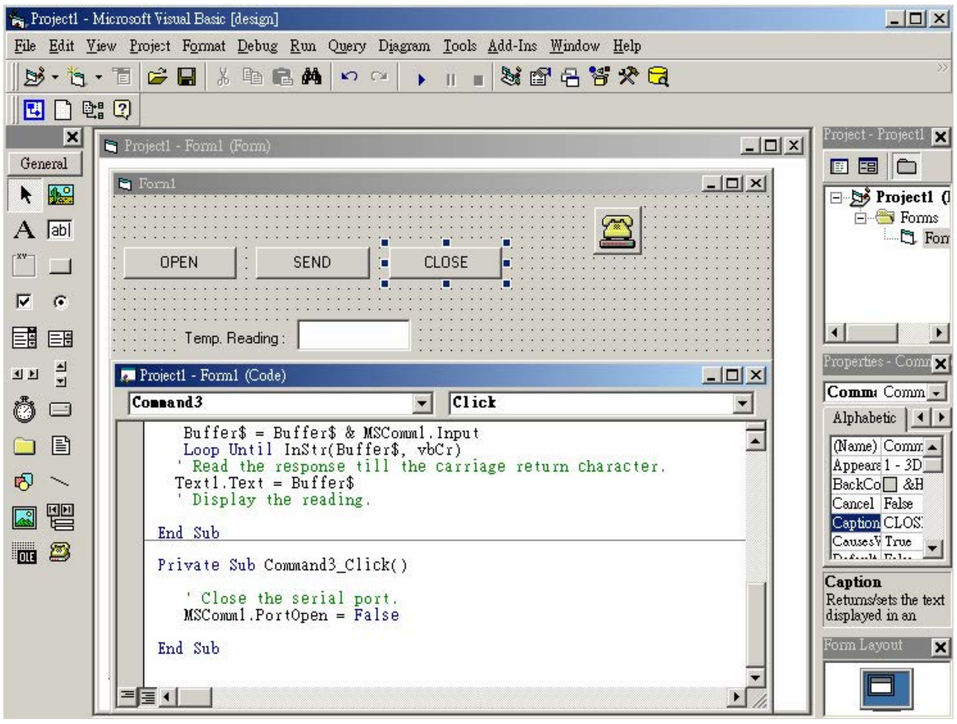


Project1 - Form1 (Code)

Command2 | Click

```
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
```

9. Click **CLOSE** and enter the following code (source code examples are listed at the end of this section).

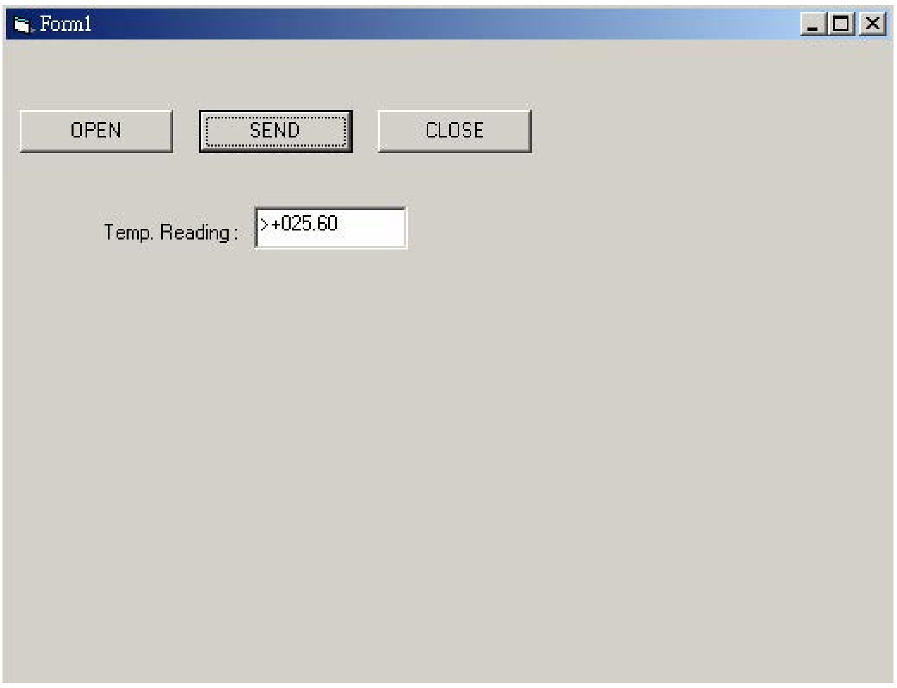


Project1 - Form1 (Code)

Command3 | Click

```
    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

Private Sub Command3_Click()

    ' Close the serial port.
    MSComm1.PortOpen = False

End Sub
```

10. Run the project, click **OPEN** to open COM1, then click **SEND** to send the get temperature reading command. Now, you will find the reading the same as the displayed format shown in the following image:



Temp. Reading : >+025.60

2.4 Program Source Code Examples

2.4.1 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
```

2.4.2 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
```

2.4.3 CLOSE Command Button

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

Chapter 3

I/O Modules

3.1 Common Specifications of ADAM-4100 Series Modules

Communication

- Speed: 1.2~115 Kbps
- Max. communication distance: 4,000 ft. (1.2 km) for RS-485
- 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 bits, 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
- Micro USB interface (B version)
- 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

Power Requirements

- Unregulated +10 ~ +48 V_{DC}

Mechanical

- PC with captive mounting hardware
- Plug-in screw terminal block accepts wire sizes of #14-22 AWG with a stripped length of 5 mm

3.2 ADAM-4117 8-ch Analog Input Module

The ADAM-4117 is a 16-bit, 8-ch 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. In addition to being able to endure harsh environments, it can also hold a more robust design. The detailed specification and enhancements are described in the following text.

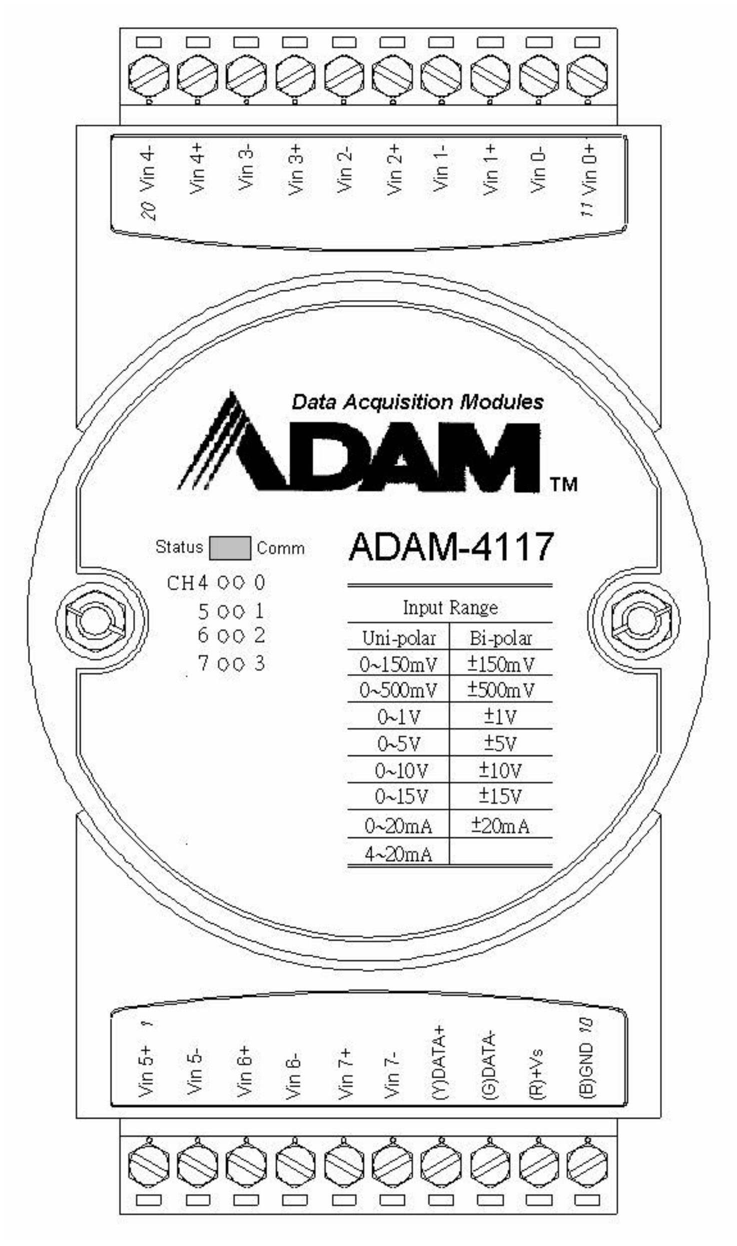


Figure 3.1 ADAM-4117-AE

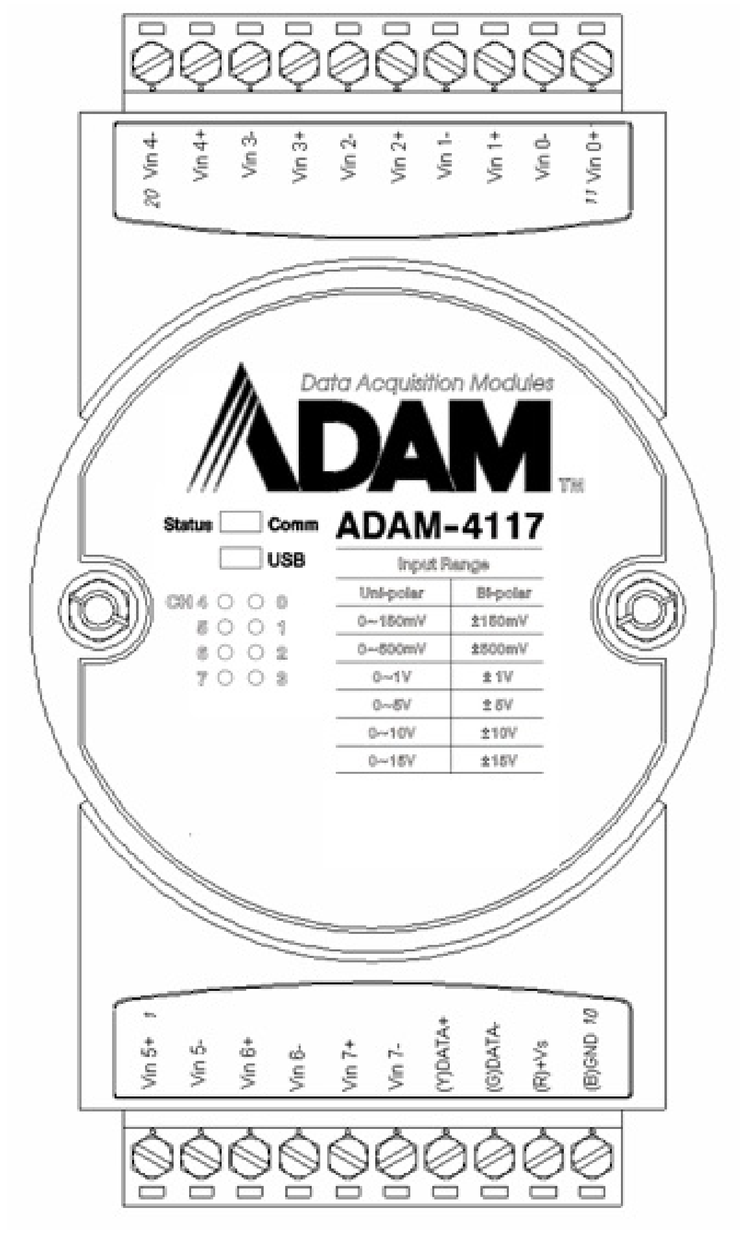


Figure 3.2 ADAM-4117-B

Specifications

Analog Input

- Effective resolution: 16-bit
- Channels: Eight differential and independent configuration channels
- High common mode: 200 V_{DC}
- ASCII command and Modbus protocol
- Input type: mV, V (supports unipolar and bipolar), mA
- Input range: 0~150 mV, 0~500 mV, 0~1 V, 0~5 V, 0~10 V, 0~15 V, ±150 mV, ±500 mV, ±1 V, ±5 V, ±10 V, ±15 V, 0~20 mA, ±20 mA, 4~20 mA
- Isolation voltage: 3000 V_{DC}
- Fault and overvoltage: Protection up to ±60 V
- Sampling rate: 10/100 Hz (selected in Adam/Apax .NET Utility)
- Input Impedance 20 MΩ

- Accuracy:
  - Voltage mode: ±0.1% or better
  - Current mode: ±0.2% or better
- Zero drift: ±6 μV/°C
- Span drift: ±25 ppm/°C (typical)
- CMR @ 50/60 Hz: 92 dB min
- Built-in dual watchdog timer
- Built-in TVS/ESD protection
- Power consumption 1.2 W @ 24 V_{DC}
- Protection (B version)
  - ESD (IEC 61000-4-2) 8 kV (air)
  - EFT (IEC 61000-4-4) 4 kV (power line)
  - Surge (IEC 61000-4-5) power 4 kV (power line)

Jumper/Switch Settings

To simplify the jumper settings, for the ADAM-4117 (B version), you can set whether the analog input type is voltage or current by adjusting the switch instead of opening the case.

Switch	**SW1**				**SW2**			
Analog input channel	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
Switch ON	Current input mode							
Switch OFF (default)	Voltage input mode							

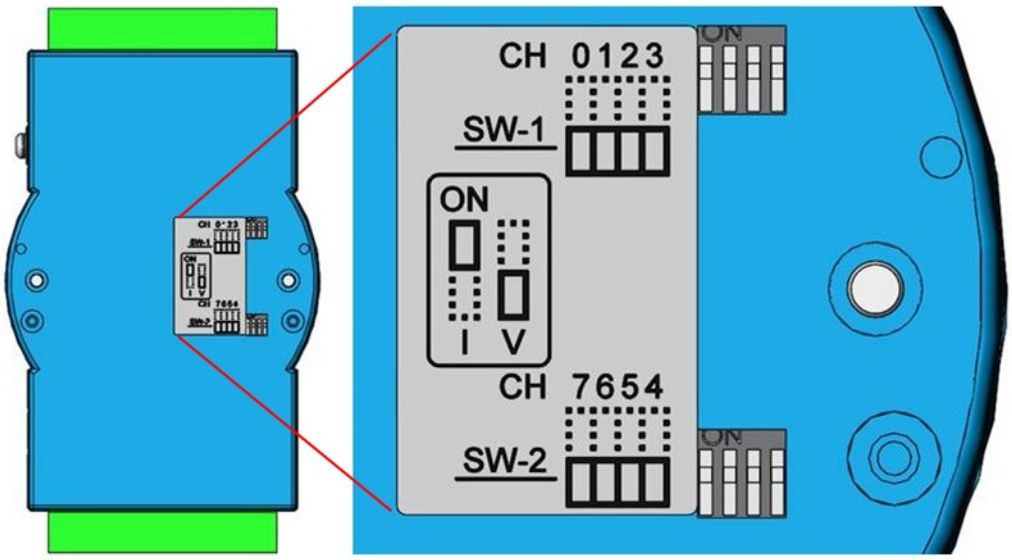


Figure 3.3 ADAM-4117-B Switch Settings

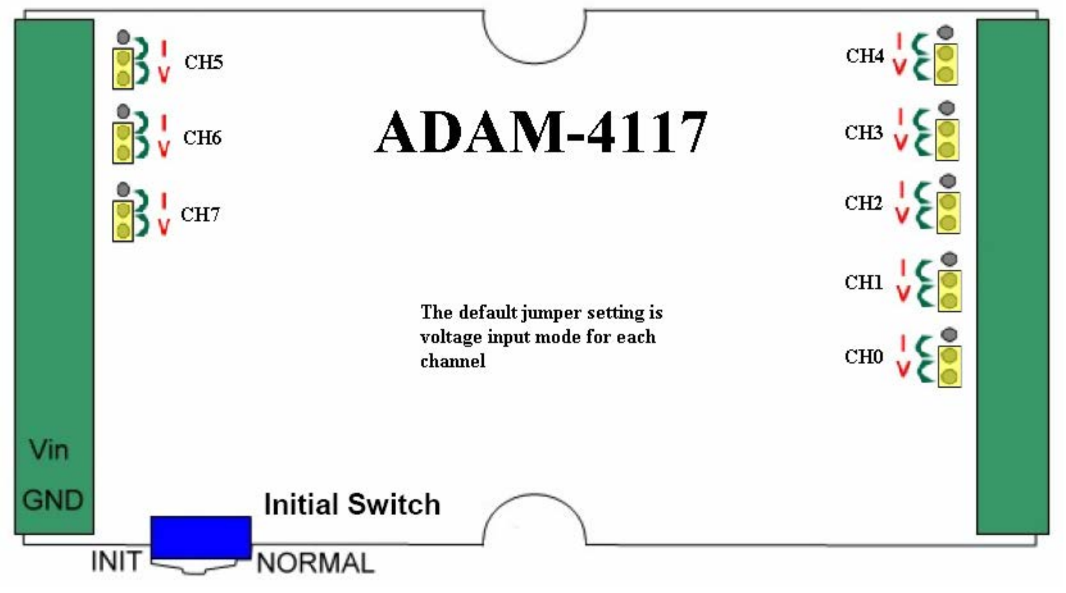
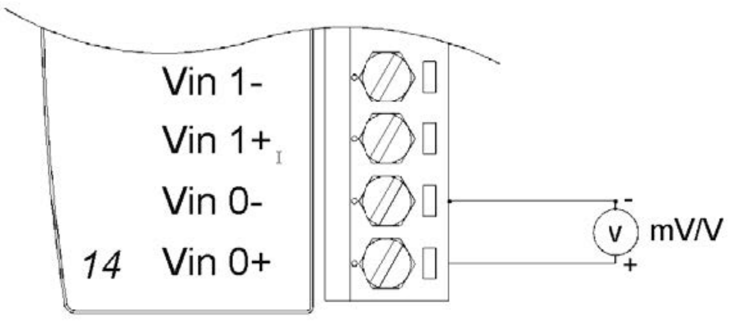

CH5
CH6
CH7
ADAM-4117
CH4
CH3
CH2
CH1
CH0
The default jumper setting is voltage input mode for each channel
Vin
GND
Initial Switch
INIT
NORMAL

Figure 3.4 ADAM-4117-AE Jumper Settings

Application Wiring



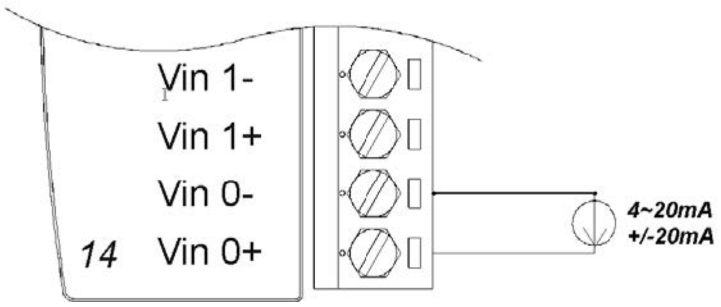


Figure 3.5 ADAM-4117 Wiring Application

3.3 ADAM-4118 8-ch Thermocouple Input Module

The ADAM-4118 is a 16-bit, 8-ch 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 (e.g., °C). To satisfy various temperature requirements in one module, each analog channel can be configured to have individual ranges, making a single module suitable for multiple applications.

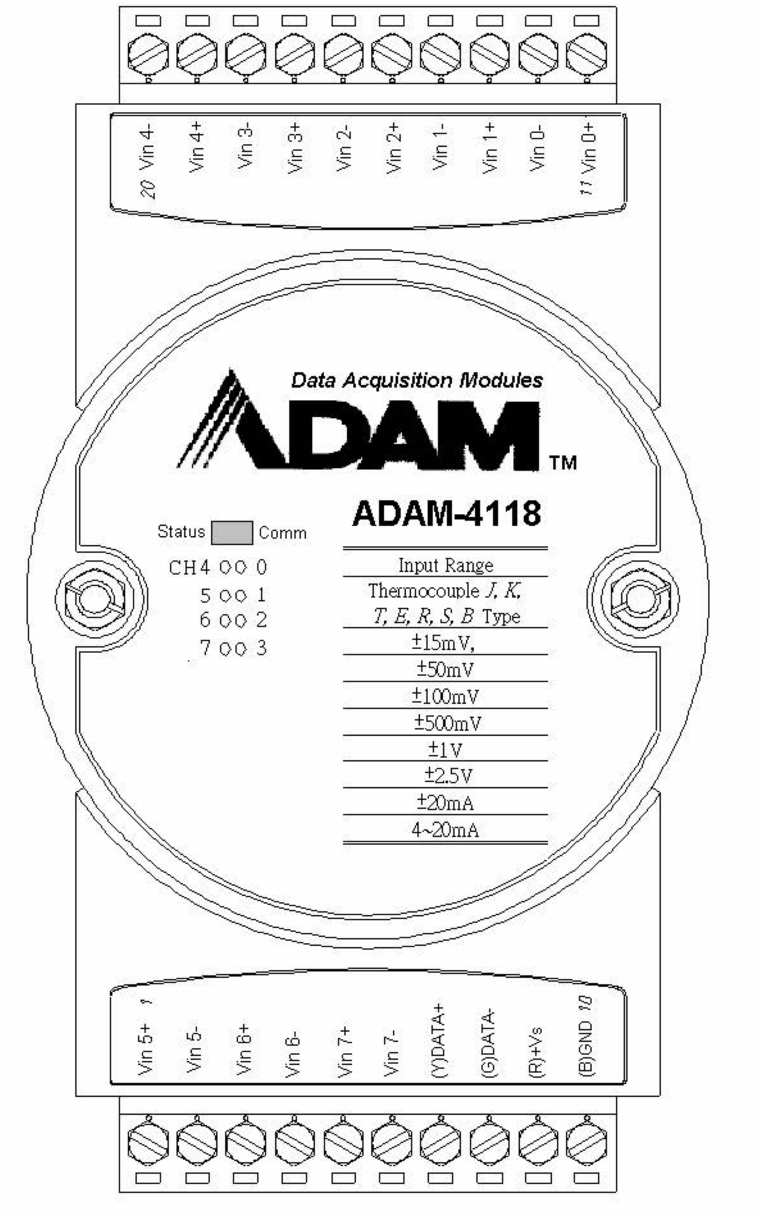


Figure 3.6 ADAM-4118-AE

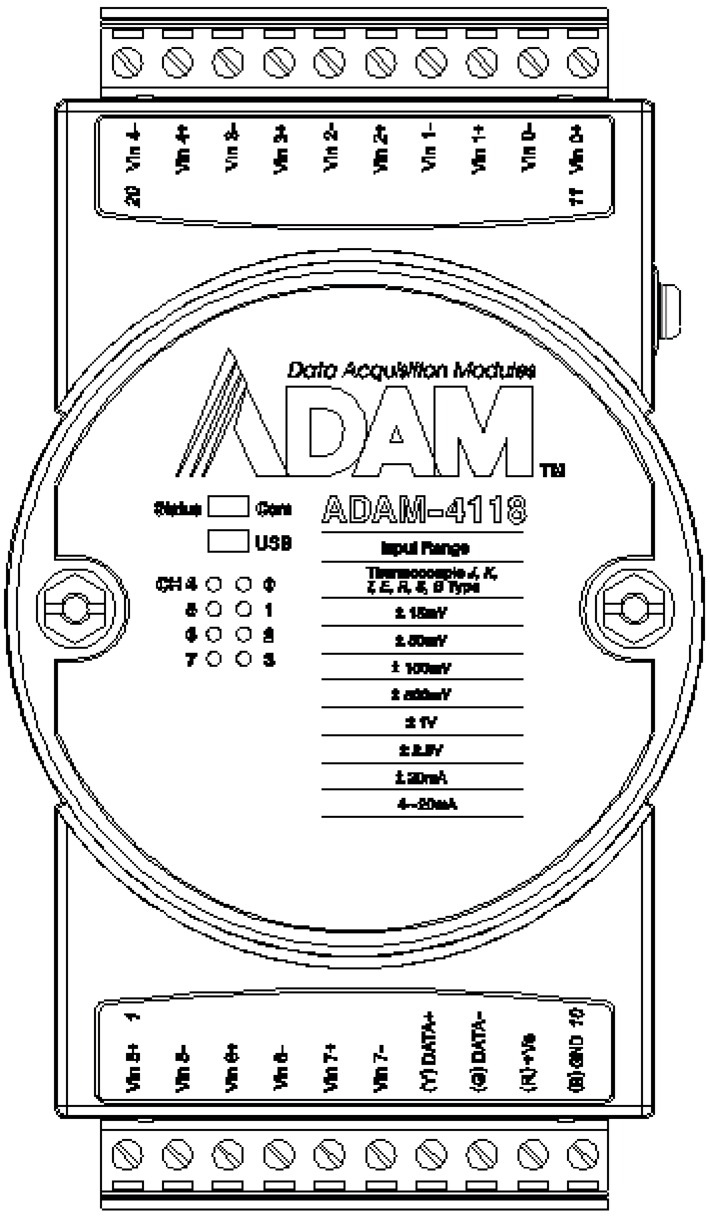


Figure 3.7 ADAM-4118-B

Specifications

Analog Input

- Effective resolution: 16-bit
- Channels: 8 differential
- ASCII command and Modbus protocol
- Input type and 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: ±15 mV, ±50 mV, ±100 mV, ±500 mV, ±1 V, ±2.5 V
- Current mode: ±20 mA, +4~20 mA
- Isolation voltage 3000 V_{DC}
- Fault and overvoltage: Protection up to ±60 V
- Sampling rate: 100 Hz (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: ±6 μV/°C
- Span drift: ±25 ppm/°C (typical)
- CMR @ 50/60 Hz: 92 dB min.
- Built-in dual watchdog timer
- Built-in TVS/ESD protection
- Power consumption: 1.2 W @ 24 V_{DC}

Jumper Settings

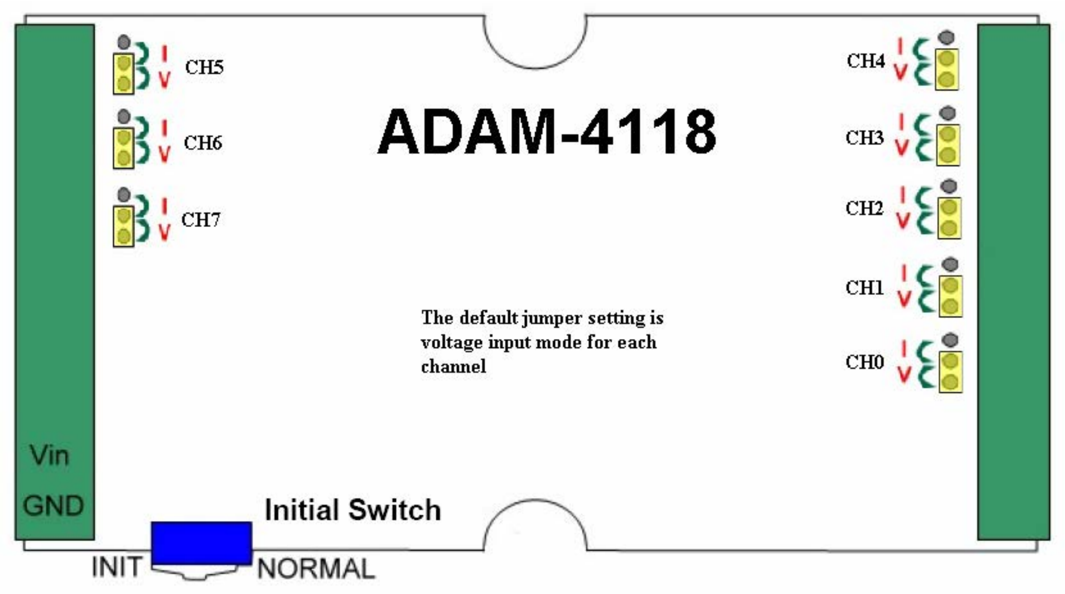

CH5
CH6
CH7
I
V
ADAM-4118
CH4
CH3
CH2
CH1
CH0
The default jumper setting is voltage input mode for each channel
Vin
GND
Initial Switch
INIT
NORMAL

Figure 3.8 ADAM-4118-AE Jumper Settings

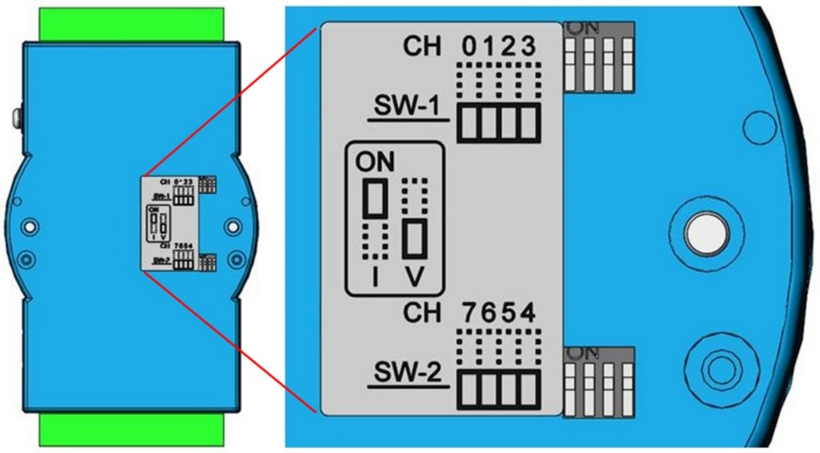


Figure 3.9 ADAM-4118-B Switch Setting

Switch	**SW1**				**SW2**			
Analog input channel	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
Switch ON	Current input mode							
Switch OFF (default)	Thermocouple (Voltage) input mode							

Application Wiring

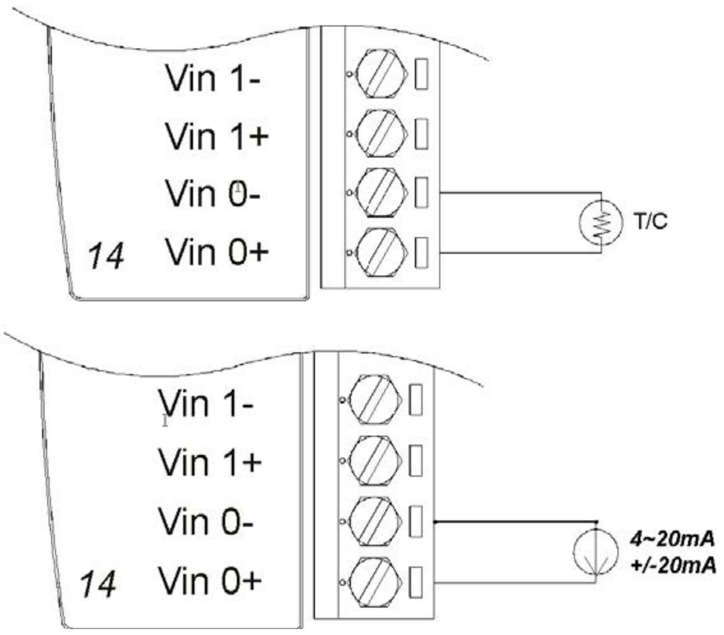


Figure 3.10 ADAM-4118 Wiring Application

*A resistor is built into the ADAM-4118 for the current input

Note! *1. Because the CJC sensor of the ADAM-4118 is located on the side of Channels 0~4, the measurement will have a difference of ±1°C between Channels 0~4 and Channels 5~7.*

2. The ADAM-4118 input range accuracy for thermocouple mode is shown in the following table:

Table 3.1: ADAM-4118 Input Range Accuracy (Thermocouple Mode)

Input Range	Typical Accuracy	Maximum Error	Unit
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 seven digital inputs and eight digital outputs. The outputs are open-collector transistor switches that can be controlled 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, the digital input channels can be used as a 3-kHz counter. Aside from its intelligent digital input functions, the digital outputs also support a 1-kHz pulse output function.

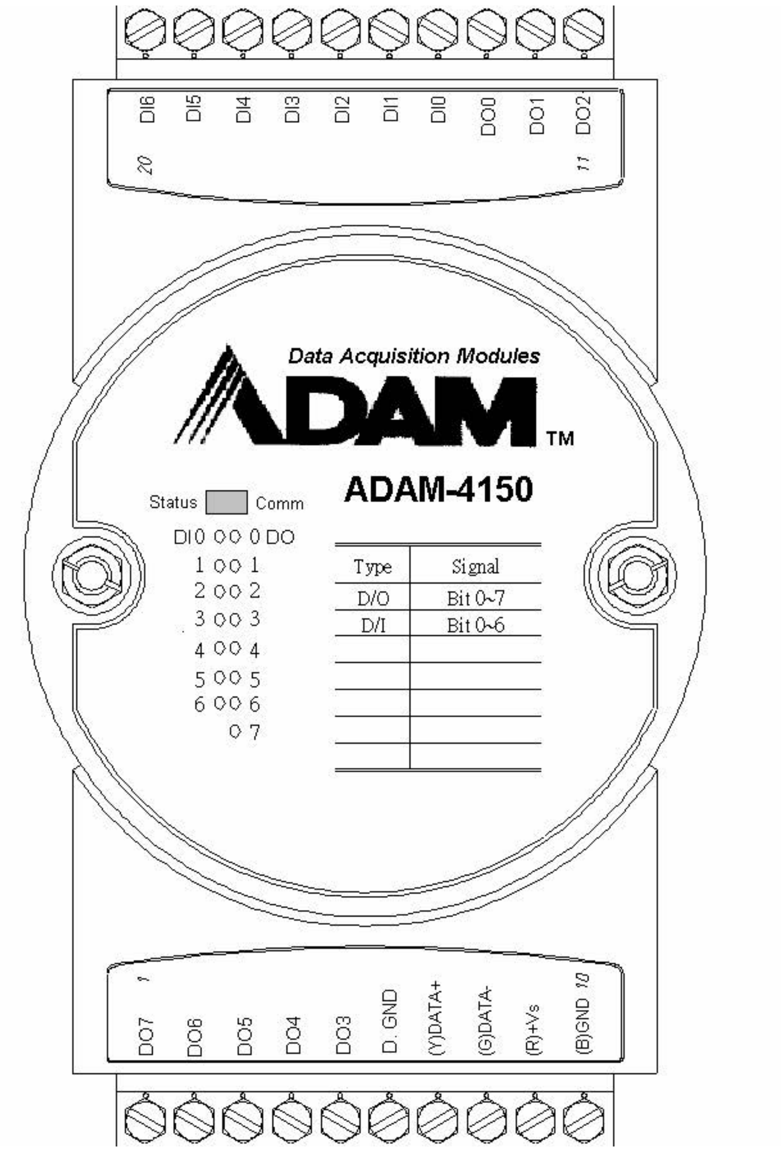


Figure 3.11 ADAM-4150-AE

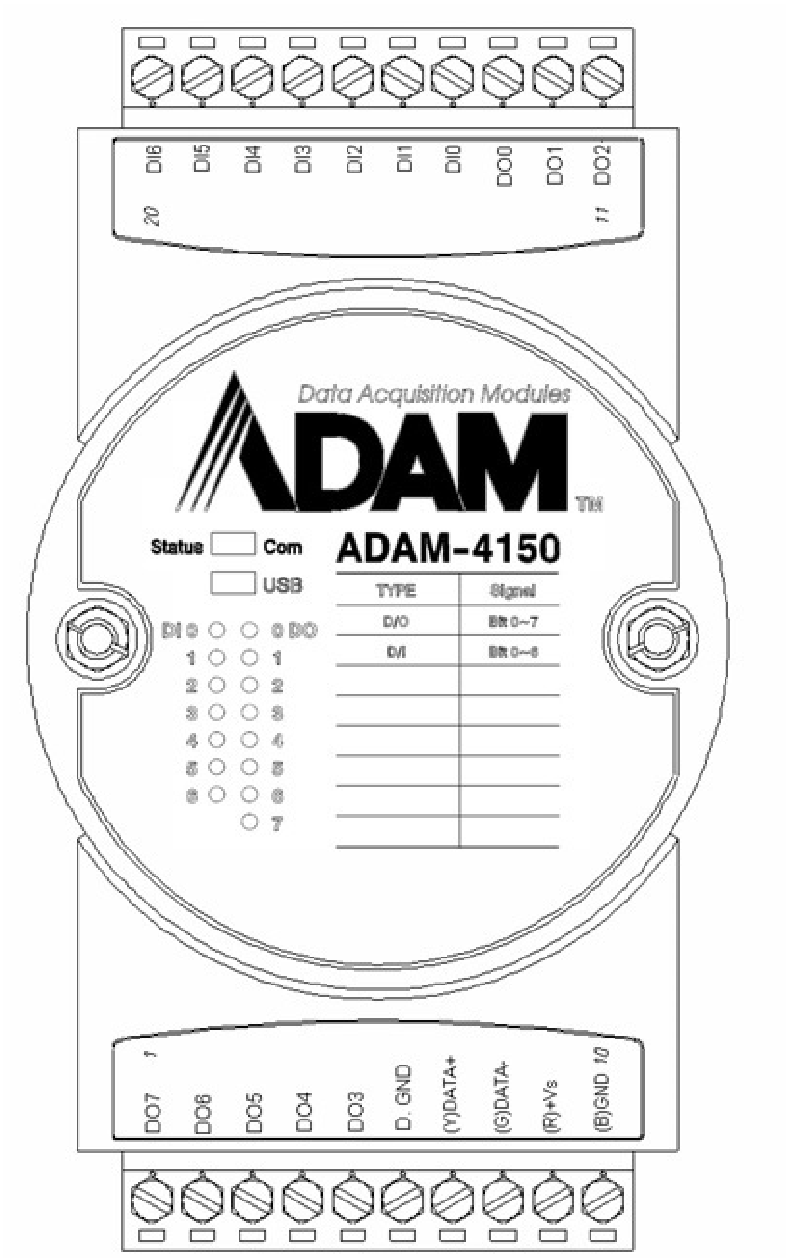


Figure 3.12 ADAM-4150-B

Specifications

- 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: +3 V (max.)
- Logic Level 1: +10 to +30 V
  - Isolation voltage: 3000 V_{DC}
  - Supports 3-kHz counter
  - Supports digital filter function

- Digital output:
  - Open drain to 40 V, 0.1A max. per channel
  - Maximum power dissipation: 1 W load
  - Ron maximum: 150 MΩ
  - Supports 1-kHz pulse output
  - Supports communication fail-safe value
- Power consumption:
  - 1.6 W (max.)
- Built-in dual watchdog timer
- Protection (B version)
  - ESD (IEC 61000-4-2) 8 kV (air)
  - EFT (IEC 61000-4-4) 4 kV (power line)
  - Surge (IEC 61000-4-5) power 4 kV (power line)

Note! *1. The digital filter function works on counter mode and can be used to set the minimum width of low and high signals in order to filter unwanted noise.*

2. Communication fail-safe values force the digital output channels to safety status when communication times out

Application Wiring

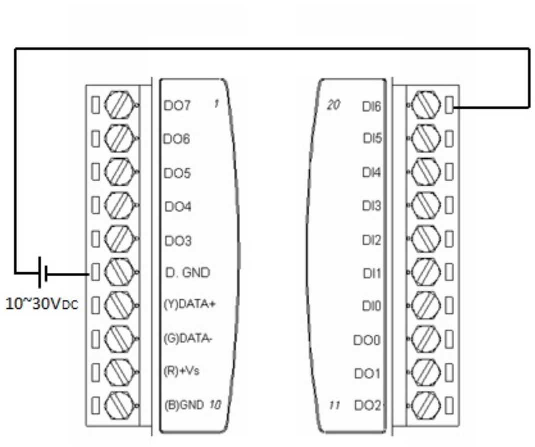


Figure 3.13 ADAM-4150 Digital Input Wet Contact Wiring

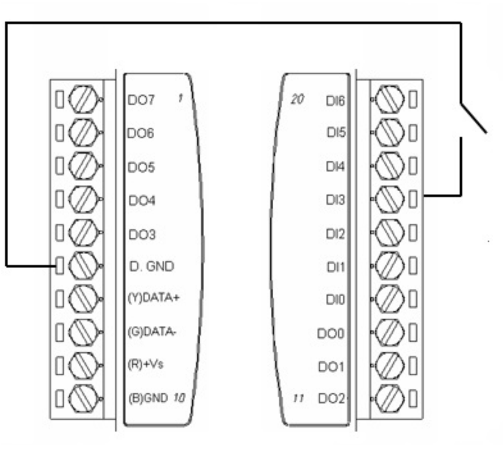


Figure 3.14 ADAM-4150 Digital Input Dry Contact Wiring

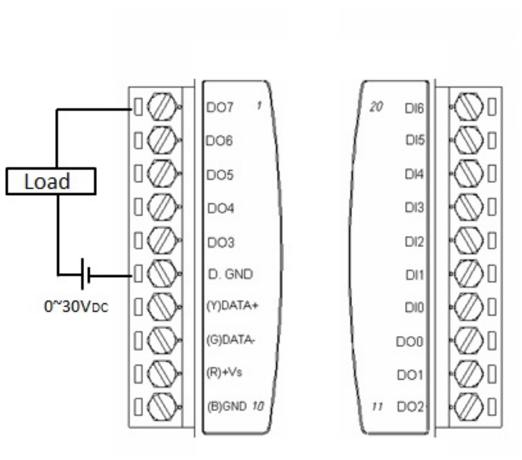


Figure 3.15 ADAM-4150 Digital Output Wiring

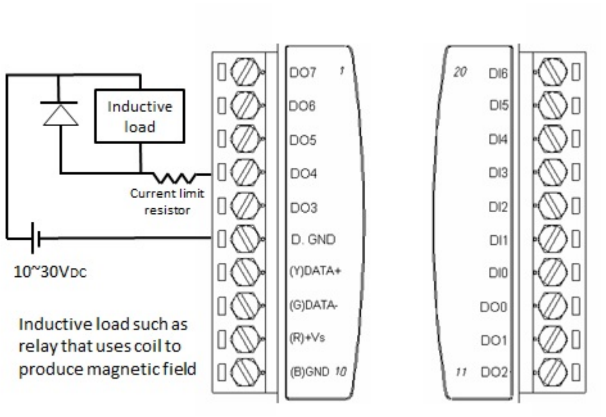


Figure 3.16 ADAM-4150 Digital Output with Inductive Load Wiring

3.5 ADAM-4168 8-ch 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.

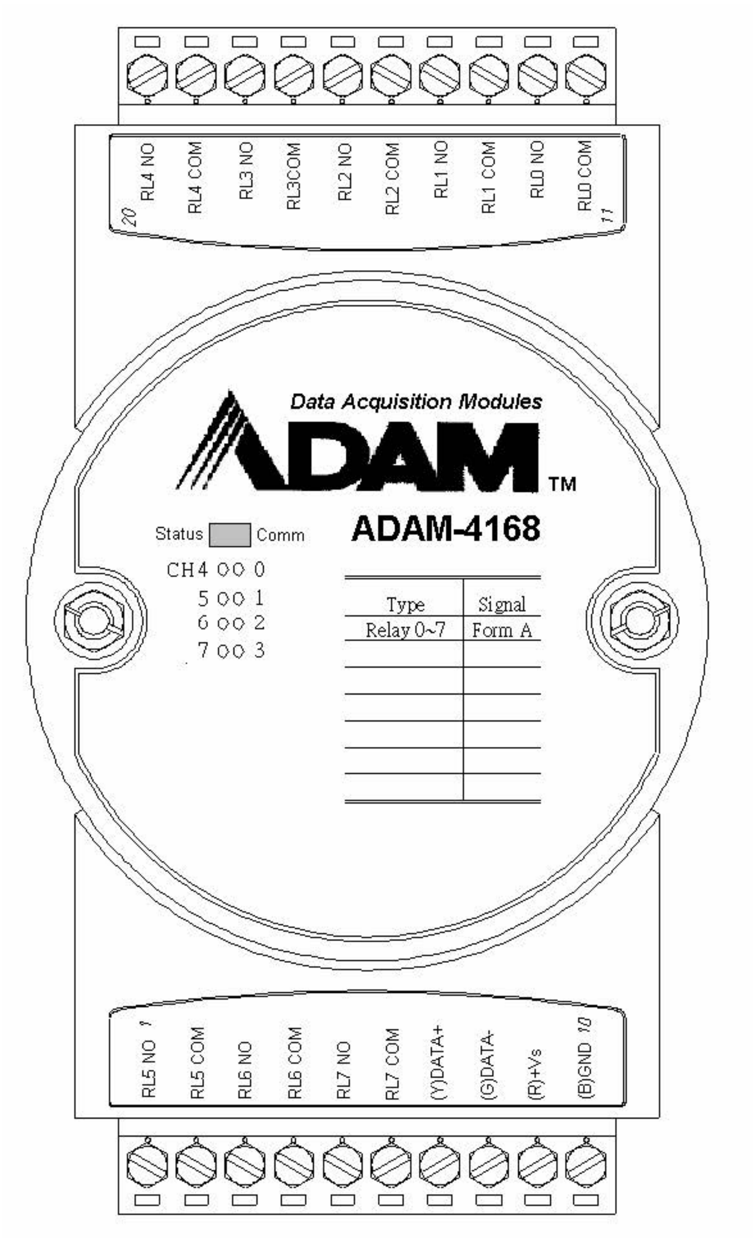


Figure 3.17 ADAM-4168-AE

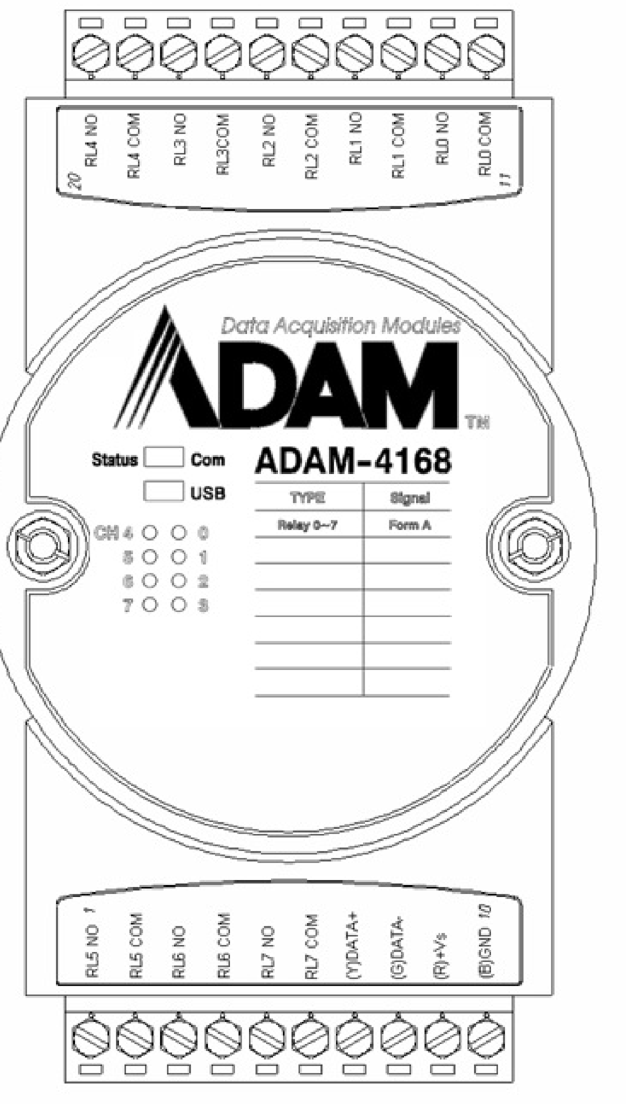


Figure 3.18 ADAM-4168-B

Specifications

- Number of Output Channel: 8 Form A
- Contact rating:
  - AC: 0.5 A @ 120 V; 0.25 A @ 240 V
  - DC: 1 A @ 30 V; 0.3 A @ 110 V
- Breakdown voltage: 750 V_{AC} (50/60 Hz)
- Insulation resistance: 1000 MΩ (min.) @ 500 V_{DC}
- Power consumption: 2.4 W (max.)
- Relay response time (typical): ON, 3 ms; OFF, 4 ms
- Total switching time: 10 ms
- Supports 100-Hz pulse output
- Supports communication fail-safe values
- Built-in dual watchdog timer
- Protection (B version)
  - ESD (IEC 61000-4-2) 8 kV (air)
  - EFT (IEC 61000-4-4) 4 kV (power line)
  - Surge (IEC 61000-4-5) power 4 kV (power line)

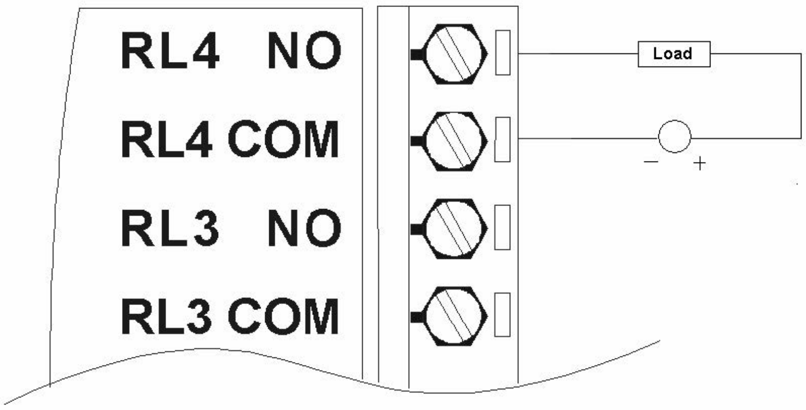


Figure 3.19 ADAM-4168 Form A Relay Output

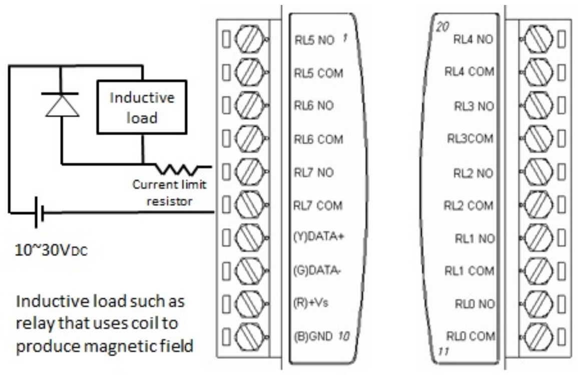


Figure 3.20 ADAM-4168 Relay Output with Inductive Load Wiring

3.6 ADAM-4115 6-ch RTD Input Module

A RTD module is popularly used for temperature measurement. Unlike the traditional design, the ADAM-4115 RTD Input Module provides six RTD input channels for different types of RTD signals.

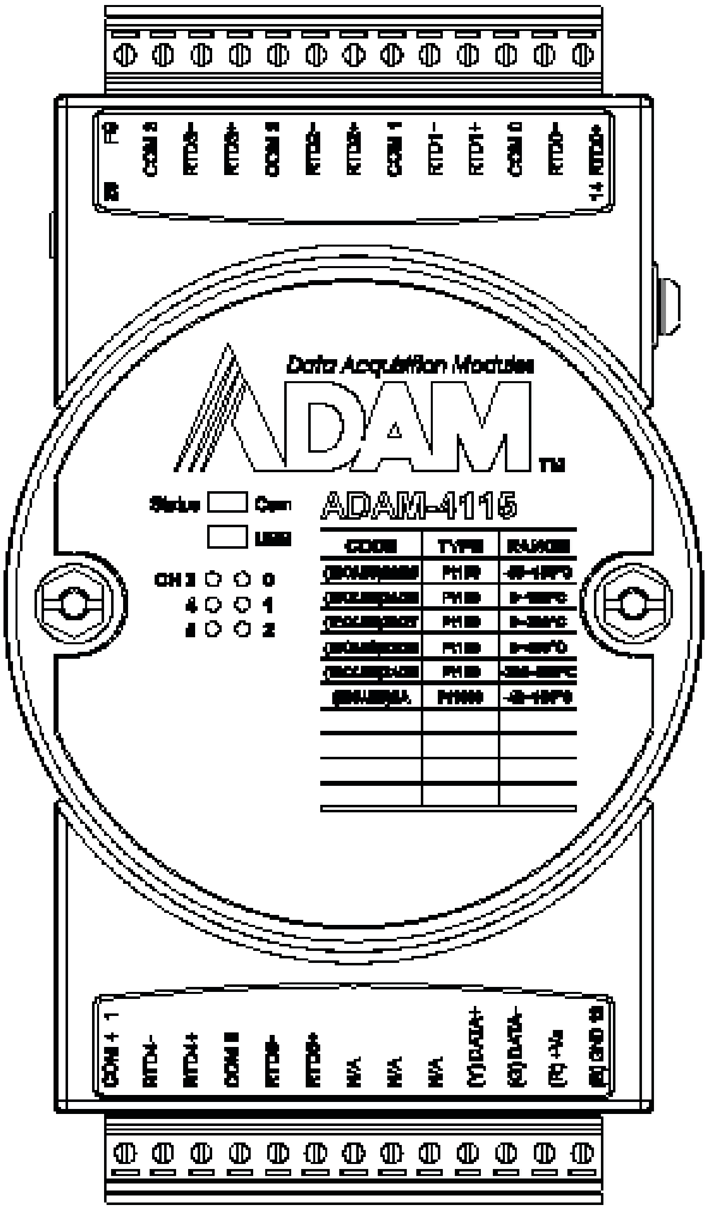


Figure 3.21 ADAM-4115-B

Specifications

RTD Input

- Resolution: 16-bit
- Channels: 6 differential
- ASCII command and Modbus protocol
- Input type and range:

  Pt100:
  - -50 ~ 150° C
  - 0 ~ 100° C
  - 0 ~ 200° C
  - 0 ~ 400° C
  - -200 ~ 200° C

Pt1000:

- -40 ~ 160° C
- Balco500:
- -30 ~ 120° C

Ni 50:

- -80 ~ 100° C

Ni 518:

- 0 ~ 100° C

- Isolation voltage 3000 VDC
- Sampling rate: 10/100 Hz total channels (selected by utility)
- Accuracy: ±0.1% or better
- Zero drift: ±6 μV/° C
- Span drift: ±25 ppm/°C (typical)
- Built-in dual watchdog timer
- Built-in TVS/ESD protection
- Power consumption: 1.2 W @ 24 VDC

Application Wiring

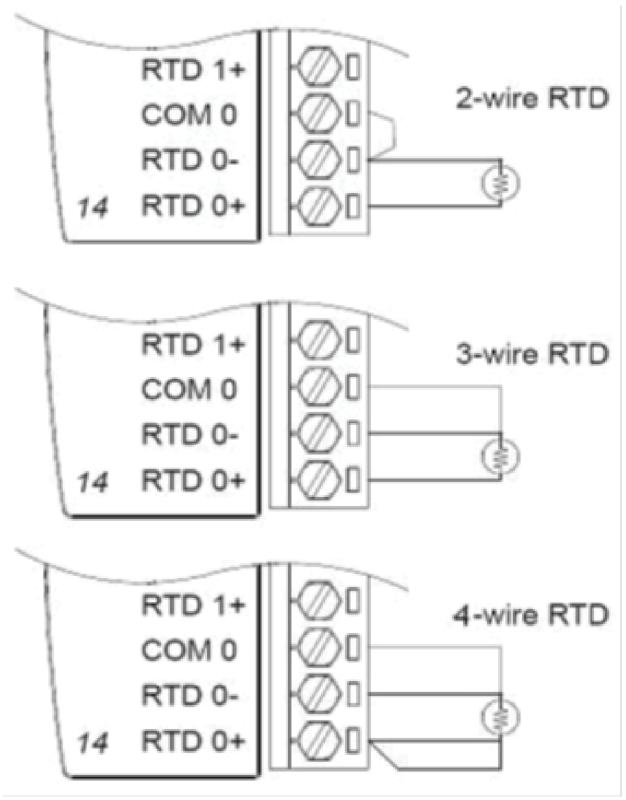


Figure 3.22 ADAM-4115 RTD Input Wiring

3.7 Digital Output Diagnostic Function

When a digital output is active, a circuit wire break or short to ground will result in an output function fail. To help rectify such a situation quickly, the digital outputs of ADAM-4100 modules (B version) have a digital output diagnostic function that can issue a notification when abnormalities are detected in the digital output. The diagnostic status is given according to the following groups:

Module	Diagnostic Group	Output Channel
ADAM-4150-B	Group 0	DO0, DO1
	Group 1	DO2, DO3
	Group 2	DO4, DO5
	Group 3	DO6, DO7

When the digital output is not active:

- A digital output circuit wire break has occurred (open load)
- A digital output connection is short to ground

When the digital output is active:

- An output has been exposed to an overcurrent (>1 A)

Note! *To ensure that the digital outputs operate normally, each digital output should be configured within the specification for the individual channels: 30 V, 100 mA (max.).*

3.7.1 Obtaining the Digital Output Diagnostic Status

Table 3.2: Obtaining the Digital Output Diagnostic Status With a Modbus Address Value

Address (4X)	Channels	Description	Attribute
40307	All	Digital output diagnostic status (for B version) 0=normal, 1=abnormal	Read

The following table shows the bit positions relative to the groups for the ADAM-4150-B. The status of the groups can thus be interpreted according to the value shown in each bit position. The group status values will be displayed as binary values, with Bit 1 being the right-most bit position and Bit 8 being the left-most bit position.

Bit Position for Modbus Address 40307	Relative Group for Interpreting the Digital Output Diagnostic Status Value (ADAM-4150-B)
Bit 1	Group 0
Bit 2	Group 1
Bit 3	Group 2
Bit 4	Group 3
Bit 5	Reserved
Bit 6	
Bit 7	
Bit 8	

Example (ADAM-4150-B)

The group status values are represented as "xxxx1110." Here, Bits 1~4 indicate the digital output diagnostic status of Groups 0~3, respectively. The group status can thus be interpreted as follows:

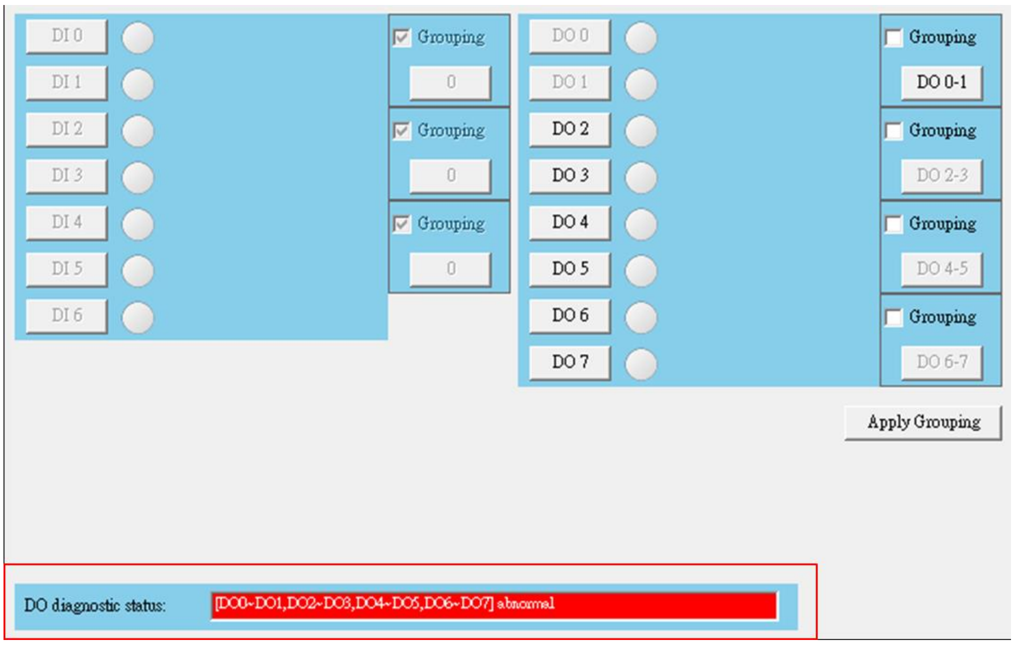
- Group 0 = 0 (normal)
- Group 1 = 1 (abnormal)
- Group 2 = 1 (abnormal)
- Group 3 = 1 (abnormal)

Obtaining the Digital Output Diagnostic Status With ASCII Commands

This example shows the ASCII command and response for requesting the status of digital outputs.

Syntax	$017
Response	!01(Group#n)…(Group #1)(Group#0)(cr)
Example	Command: $017 Response: !011110
Description	Because the ADAM-4150 has four digital output groups for the diagnostic status, the bit positions from right to left indicate the status of Groups 0~3, as follows: - Group 0 = 0 (normal) - Group 1 = 1 (abnormal) - Group 2 = 1 (abnormal) - Group 3 = 1 (abnormal)

The diagnostic status can be obtained using Adam/Apax .NET Utility. If a channel in the digital output diagnostic group is not wired or has a wire break or short to ground error, the digital output diagnostic column will show abnormal information. When all channels in the group are connected correctly before the digital outputs are activated, the field will show “all normal.”



DO diagnostic status: [DO0~DO1,DO2~DO3,DO4~DO5,DO6~DO7] abnormal

3.8 Reliability Function

ADAM-4100 series modules (B version) have been designed with reliability functions. The reliability function reduces the chance of failure due to unexpected events. You can enable/disable the function by setting in Adam/Apax .NET Utility.

Digital Input Reliability

Every digital input reliability group contains two digital input channels. When two channels in the group have the same status, the status of the group will change accordingly, as shown below. This function can be used to double-confirm a module is receiving a signal from digital sensors.

x = 0,1,2,3…	Logic Level	
DI (2x)	1	0
DI (2x + 1)	1	0
DI Reliability Group (x)	1	0

Detecting a Digital Sensor Abnormality in Simultaneous Applications

The reliability function can be used for applications where the output levels of two digital sensors are the same in normal cases. To detect a digital sensor abnormality, two sensors connect to two digital input channels in the digital input reliability group. If one of the sensors is defective, the output level will not be identical. As a result, a Modbus value of 40306 would be the 0800, indicating that an abnormality (refer to the Modbus mapping table in Section 3.7.1); this can immediately inform you of an abnormal situation so that prompt action can be taken.

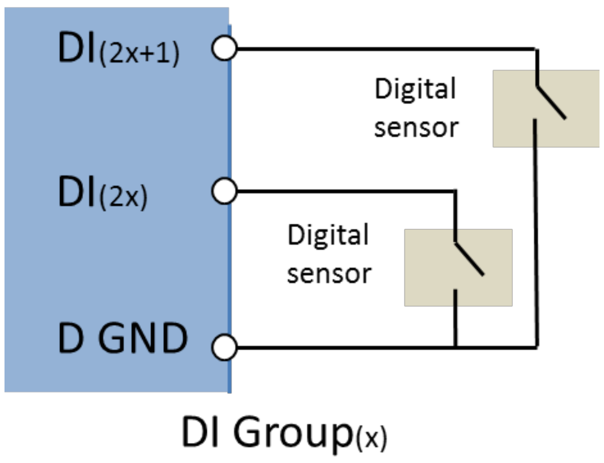


Figure 3.23 Detecting a Digital Sensor Abnormality in Simultaneous Applications

Digital Output Reliability

Every digital output reliability group contains two digital input channels. When the status of a digital output reliability group changes, the two digital output channels in the group will be turned on, as shown below.

	Logic Level	
DO (2x)	1	0
DO (2x + 1)	1	0
DO Reliability Group (x)	1	0

Note! *1. When the digital output reliability function is enabled, the FSV function will not work for the channels in the digital output reliability group.*

2. When the digital output reliability function is enabled, the group value will align with the value of the lower channel number. For example, assume that DO0 = high and DO1 = low before the reliability function is enabled; when the reliability function is enabled for Group 0, the value for that group will be "high".

Double-Confirming that an Actuator is Activated

This function can be used to double-confirm that a digital output has been applied successfully to an actuator. As shown in the following figure, when Group(x) is active, even if DO(2x + 1) has failed due to an unexpected event, such as a wire break, DO(2x) will still apply the digital signal to the actuator.

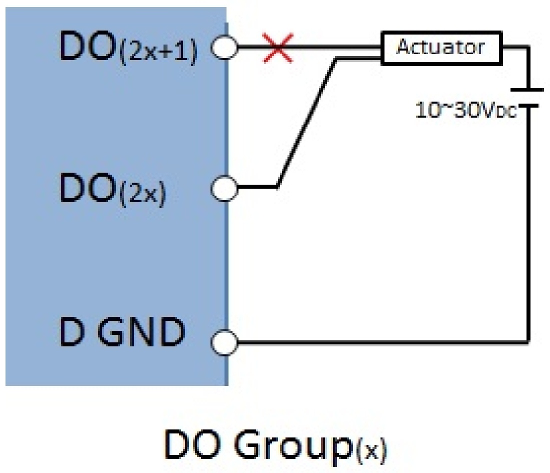


Figure 3.24 Double-Confirming an Actuator is Activated

Process to Trigger Two Actuators at the Same Time

By activating the group status, you can set two digital output channels at the same time. This function reduces the effort it would take to for programming to trigger two actuators at the same time. As shown in Fig 1.4, when activating the Group(x), DO(2x) and DO(2x + 1) will be activate at the same time to trigger Actuator 1 and Actuator 2.

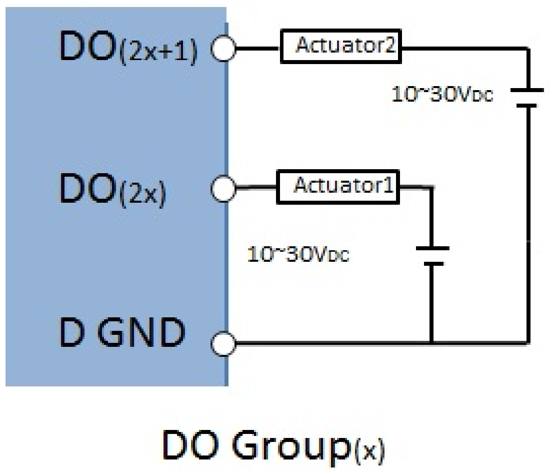


Figure 3.25 Process to Trigger Two Actuators at the Same Time

3.8.1 Modbus Mapping Table

Digital Output Reliability

Address	Channel	Item	Attribute
40305	0~7	Digital output reliability function enable/disable	R/W
40306	0~7	Digital output reliability status	R/W

Digital Output Reliability Function Enable/Disable (Enable: 1/Disable: 0)

Example: Modbus 40305 = 0x0E(hex) = 1110 (bin)

- DO reliability group 0 (DO0&1) = bit0:0 (group function disable)
- DO reliability group 1 (DO2&3) = bit1:1 (enable)
- DO reliability group 2 (DO4&5) = bit2:1 (enable)
- DO reliability group 3 (DO6&7) = bit3:1 (enable)

Digital Output Reliability Group Value

Example: Modbus 40306 = 0x0E(hex) = 1110 (bin)

- DO reliability group 0 (DO0&1) = bit0:0 (low),DO0&1 are low
- DO reliability group 1 (DO2&3) = bit1:1 (high), DO2&3 are high
- DO reliability group 2 (DO4&5) = bit2:1 (high), DO4&5 are high
- DO reliability group 3 (DO6&7) = bit3:1 (high), DO6&7 are high

Digital Input Reliability

Address	Channel	Item	Attribute
40309	DI Reliability Group 0 (DI0&1)	DI reliability status	R
40310	DI Reliability Group 1 (DI2&3)	DI reliability status	R
40311	DI Reliability Group 2 (DI4&5)	DI reliability status	R

(x=0,1,2,3…)	Modbus value			
DI (2x)	1	0	1	0
DI (2x+1)	1	1	0	0
DI Reliability Group(x)	1	0800		0

3.8.2 ASCII Commands

$01PDaaaa

Name	Enable/disable digital output reliability group
Syntax	$01PDaaaa a: function enable/disable, 0: disable, 1: enable
Response	!01
Example	command: $01PD1100 response: !01

Description	Group 0 status (DO0&1) = 1, enabled Group 1 status (DO2&3) = 1, enabled Group 2 status (DO4&5) = 0, disabled Group 3 status (DO6&7) = 0, disabled

$01PD

Name	Get digital output group function enabled/disabled status
Syntax	$01PD
Response	!01bbbb
Example	command: $01PD response: !011100
Description	Group 0 status (DO0&1) =1, enabled Group 1 status (DO2&3) =1, enabled Group 2 status (DO4&5) =0, disabled Group 3 status (DO6&7) =0, disabled

#01PCdddd

Name	Set the digital I/O reliability group value
Syntax	$01PCdddd d: value of the reliability group 1 = high level, 0 = low level
Response	!01
Example	command: $01PC1100 response: !01
Description	Group 0 status (DO0&1) =1, group value high Group 1 status (DO2&3) =1, group value high Group 2 status (DO4&5) =0, group value low Group 3 status (DO6&7) =0, group value low

$01PC

Name	Get the digital reliability group value
Syntax	$01PC
Response	!OOOOIII0 (ADAM-4150) !OOOO (ADAM-4168) O: Digital/relay output channel status I: Digital input channel status
Example	command: $01PC response: !11001800
Description	Group 0 status (DO0&1) =1, group value high Group 1 status (DO2&3) =1, group value high Group 2 status (DO4&5) =0, group value low Group 3 status (DO6&7) =0, group value low Group 0 (DI0&1) =1, both channel are high Group 1(DI2&3) = 8, channel status are different Group 2 (DI4&5) =0, both channel are low

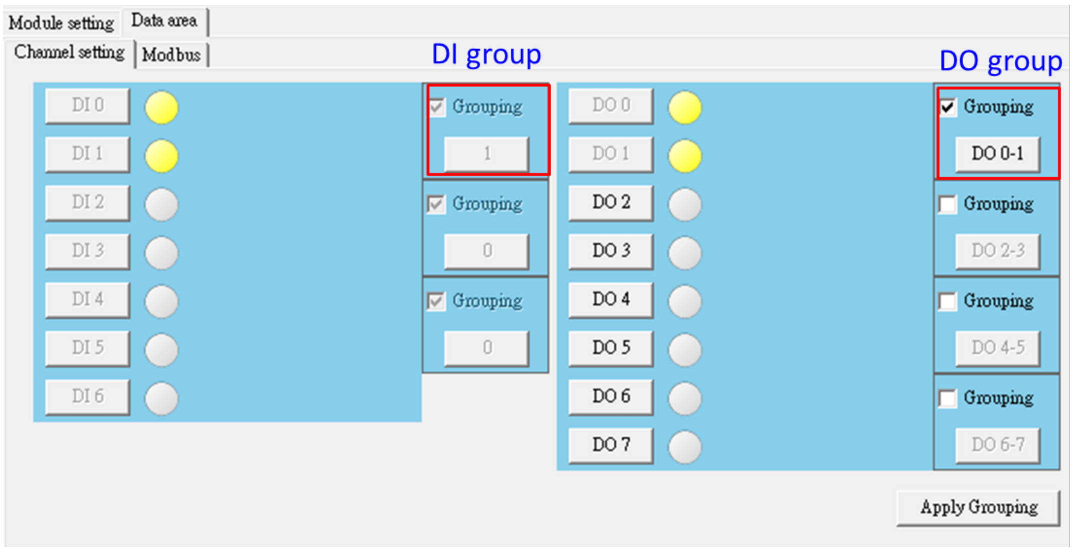
3.8.3 Configuration by ADAM/Apax .NET utility

Digital Output Reliability Function

The digital output reliability function can be enabled by checking Grouping and then clicking Apply Grouping.

Digital Input Reliability Function

The digital input group status can be observed using the utility. When all channels in a group turn to 1 (high level), the digital input reliability group status will also turn to 1(high level).



Chapter 4

Command Set

4.1 Introduction to ADAM Module Commands

To avoid communication conflicts among devices trying to send data simultaneously, all traffic is 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 control to the host.

Changing an ADAM module's configuration might require the module to perform auto calibration before the changes take effect. This is particularly the case when the range is modified. The module has to perform all stages of auto calibration, which is also performed during startup. When the calibration process is underway, the module will not respond to any other commands. The command set in the following pages includes the exact delays that might occur while a module is being reconfigured.

4.2 Command Syntax

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

Every command begins with a delimiter character. There are four valid characters:

- the dollar sign: $
- the pound sign: #
- the percentage sign: %
- the at sign: @

The delimiter character is followed by a two-character hex address that specifies the target module. The command string then follows the address. Depending on the command, an optional data segment may follow the command string, and an optional two-character checksum may be appended to the complete string. Every command is terminated by a carriage return (cr).

Note! *ALL COMMANDS SHOULD BE ISSUED IN UPPERCASE CHARACTERS!*

Before each command set is presented in the following text, an I/O module command search table is given 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 in the following text.

4.3 Analog I/O Module Command Search Table

Table 4.1: ADAM-4117 and ADAM-4118 Command Table

Command Syntax	Command Name	Command Description
%AANNTTC-CFF	Configuration	Sets a module's address, input range, baud rate, data format, checksum status, and/or integration time
#AAN	Read single analog input	Returns the input value of channel N
#AA	Read all analog inputs	Returns the input values of all channels
$AA0	Span calibration	Calibrates a module to correct for gain errors
$AA1	Offset calibration	Calibrates a module to correct for offset errors
$AA2	Configuration status	Returns a module's configuration data
$AA5VV	Set multiplexing status on all channels	Independently sets the multiplexing status of all channels to enable/disable
$AA6	Read multiplexing status of all channels	Reads the multiplexing status (enable/ disable) of all channels
$AAF	Read firmware version	Returns a module's firmware version
$AAM	Read module name	Return a module's name
$AA7CiRrr	Configure single channel type/range	Sets the input type/range of channel N
$AA8Ci	Read single channel type/range	Returns the input type/range of channel N
$AAXnnnn	Configure watchdog timer	Sets the watchdog timer communication cycle time
$AAY	Read watchdog timer	Returns the watchdog communication cycle time
$AAMC	Get auto-filter sample rate	Returns the auto-filter sample rate
#AAMKmm	Configure software filter multiplexing status	Set the software filter multiplexing status of all channels to enable/disable
$AAMD	Read software filter channel enable/ disable	Read Software Filter channel Enable/ Disable
#AAFQm	Locate module	Locates a module by activating its locate LED
$AA3	Read CJC sensor value*	Returns the CJC sensor value
$AA9SNNNN	CJC offset calibration*	Calibrates a module to adjust for offset errors in its CJC sensor

*These commands apply only to the ADAM-4118

Table 4.2: ADAM-4115 Command Table

Command Syntax	Command Name	Command Description
$AA6	Read Channel Status	Get the enable/disable status of all channels in an analog module
$AA0	Span Calibration	Calibrates an analog input module to correct for gain errors
$AA1	Zero Calibration	Calibrates an analog input module to correct for initial value errors
$AA0Ci	Single Channel Span Calibration	Calibrates a specified channel to correct for gain errors
$AA1Ci	Single Channel Offset Calibration	Calibrates a specified channel to correct for offset errors
$AA7CiRrr	Single Channel Range Configuration	Configure the input type and range of the specified channel in an analog input module
$AA8Ci	Read Single Channel Range Configuration	Get the input type and range of the specified channel in an analog input module
$AAXnnnn	Watchdog Timer Setting	Set communication WDT cycle time from 0000 ~ 9999 (unit: 0.1 second. if value is 0000, the communication WDT function will be disable)
$AAY	Read Watchdog Timer Setting	Read the cycle time setting of communication WDT %AANNTTCCFF
$AA2	Configuration status	Returns the configuration parameters for the specified analog input module
$AAF	Read firmware version	Returns the firmware version code from the specified analog input module
$AAM	Read module name	Returns the module name from the specified analog input module
#AA	Read all analog inputs	Returns the input value from a specified analog input module in the currently configured data format
#AAN	Read single analog input	Returns the input value from channel number n of the specified analog input module
$AA5VV	Set multiplexing status on all channels	Enables/disables multiplexing simultaneously for separate channels of the specified input module

4.4 Analog I/O Module Command Set

%AANNTTCCFF

Name	Configuration
Description	Sets a module's address, input range, baud rate, data format, checksum status, and/or integration time
Syntax	%AANNTTCCFF(cr) % is a delimiter AA(range 00-FF) is the 2-character hex address of the specified module NN represents the new hex address of the specified module (range 00h~FFh) TT represents 00 CC represents the baud rate code FF is a hex 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 2~5 are not used and are set to 0 (cr) is the terminating character, carriage return (0Dh)

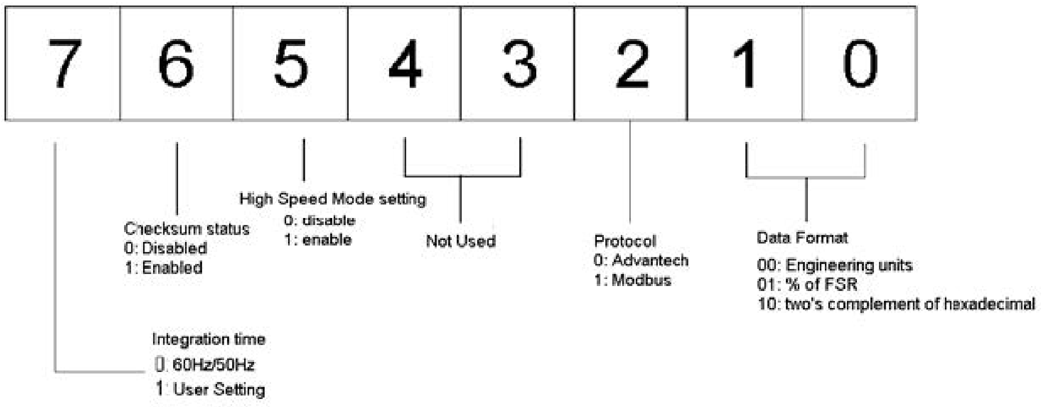


Figure 4.1 Data Format for 8-Bit Parameter

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 the baud rate or checksum settings There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hexadecimal address of the specified module (cr) is the terminating character, carriage return (0Dh)
Example	command: %2324050600(cr) response: !24(cr) The ADAM-4011 module with Address 23h is to have the new address of 24h as well as an input range of ±2.5 V, baud rate of 9600, and integration time of 50 ms (60 Hz) with data in engineering unit format and no checksum checking or generation. The response indicates that the command was received. Wait for 7 s for the new settings to take effect before issuing a new command to the module.

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

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 the baud rate and check-sum configuration details in Chapter 2.*

Table 4.3: Input Range Codes for the ADAM-4117 (Type Code)

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

Table 4.4: Input Range Codes for the ADAM-4118 (Type Code)

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 to 7600°C
0F	Type K thermocouple: 0 to 1370°C
10	Type T thermocouple: -1000 to 400°C
11	Type E thermocouple: 0 to 1000°C
12	Type R thermocouple: 500 to 1750°C
13	Type S thermocouple: 500 to 1750°C
14	Type B thermocouple: 500 to 1800°C

Table 4.5: Baud Rate Codes

Baud Rate Code (Hex)	Baud Rate
03	1,200 bps
04	2,400 bps
05	4,800 bps
06	9,600 bps
07	19.2 kbps
08	38.4 kbps
09	57.6 kbps
0A	115.2 kbps
0B	230.4 kbps

Table 4.6: ADAM-4115 Input Range code

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

#AAN

Name	Read single analog input
Description	Returns the input value of channel N in the currently configured data format
Syntax	#AAN(cr)# is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module N refers to the specific channel you want to read (range 0~7) (cr) is the terminating character, carriage return (0Dh)
Response	>(data)(cr) There is no response if the module detects a syntax, communication error, or if the address does not exist > is a delimiter (data) is the input value of channel N; data consist 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 with an input value of +1.4567 V for Channel 0.

#AA

Name	Read all analog inputs
Description	Returns the input values of all channels
Syntax	#AA(cr) # is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module (cr) is the terminating character, carriage return (0Dh)
Response	>(data)(cr) There is no response if the module detects a syntax, communication error, or if the address does not exist > is a delimiter (data) is the input value in the configured data format of the module (for data formats, see Appendix B) (cr) is the terminating character, carriage return (0Dh)
Example	command: #33(cr) response: >+5.8222(cr) The command accesses the analog input module at Address 33h for its input value. The analog input module responds with +5.8222 V (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, +7.2567, +7.3125, +7.1000, +7.4712, +7.2555, +7.1234, and +7.5678 V.
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

Note! *When modules measure thermocouple input values that are outside of their configured range, they will send data that implies input out-of-bounds. The next table shows the return values of the modules; however, it depends on the configured data format and the input value, which may fall under or exceed the configured range.*

An "input out-of-bounds" warning occurs only when modules are configured for thermocouples. 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 to have an input range of T/C type J (Input range: 0~760°C) and the data format is in engineering units. The module measures an input value of 820°C.

Example	command: #D1(cr) response: >+9999(cr) By returning a high value of +9999, the module at Address D1h indicates that the measured input value exceeds the configured range.

$AA0

Name	Span calibration
Description	Calibrates a module to correct gain errors
Syntax	$AA0(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 0 refers to the span calibration command (cr) is the terminating character, carriage return (0Dh)
Response	!AA(cr) if the command is valid ?AA(cr) if an invalid operation was entered There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module (cr) represents the terminating character, carriage return (0Dh) 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.

Note! *An analog input module requires up to 7 s to perform auto calibration and ranging after it has received a span calibration command. During this time, the module cannot be addressed to perform any other actions.*

$AA1

Name	Offset calibration
Description	Calibrates a module to correct for offset errors
Syntax	$AA1(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 1 refers to 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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) represents the 2-character hex address of the specified module (cr) represents the terminating character, carriage return (0Dh)

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.

Note: An analog input module requires up to 7s to perform auto calibration and ranging after it has received an offset calibration command. During this time, the module cannot be addressed to perform any other actions.

$AA2

Name	Configuration status
Description	Returns a module's configuration data
Syntax	$AA2(cr) $ is a delimiter AA (range 00-FF) represents the 2-character hex address of the specified module 2 refers to 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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module TT represents the type code that determines the input range CC represents the baud rate code FF is a hex number equal to an 8-bit parameter that represents the data format, checksum status, and integration time. The layout of this parameter is shown in Figure 4-1. Bits 2~5 are not used and are set to 0. (cr) is the terminating character, carriage return (0Dh) (also see %AANNTTCCFF)

Example	command: $452(cr) response: !45050600(cr) The command asks the analog input module at Address 45h to send its configuration data.

The module responds with an input range of 2.5 V, a baud rate of 9600 bps, and 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.

$AA5VV

Name	Set multiplexing status on all channels
Description	Independently sets the multiplexing status of all channels to enable/disable
Syntax	$AA5VV(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 5 refers to the set multiplexing status on all channels command VV are the 2 hexadecimal values interpreted as two binary words (4-bit). The first word represents the status of Channels 4~7, and the second word represents the status of Channels 0~3 (0 = disabled, 1 = 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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module (cr) is the terminating character, carriage return (0Dh)
Example	command: $00581(cr) response: 00(cr) 8 (hex) = 1000 (binary), which enables Channel 7 and disables Channels 4~6 1 (hex) = 0001 (binary), which enables Channel 0 and disables Channels 1~3

$AA6

Name	Read multiplexing status of all channels
Description	Reads the multiplexing status (enable/disable) of all channels
Syntax	$AA6(cr) AA (range 00-FF) is the 2-character hex address of the specified module 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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module VV are two hexadecimal values that are interpreted as two binary words (4-bit). The first word represents the status of Channels 4~7, and the second word represents the status of Channels 0~3 (0 = disabled, 1 = 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 its input channels. The module responds with FF (hex) = 1111 and 1111 (binary), meaning that all of its multiplex channels are enabled.

$AAF

Name	Read firmware version
Description	Returns a module's firmware version
Syntax	$AAF (cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module F refers to the read firmware version command (cr) is the terminating character, carriage return (ODh)
Response	!AA(Version)(cr) if the command is valid ! is a delimiter character indicating that a valid command was received There is no response if the module detects a syntax, communication error, or if the address does not exist AA (range 00-FF) is the 2-character hex address of the specified module (Version) is the firmware version code of the specified module (cr) is the terminating character, carriage return (ODh)

$AAM

Name	Read module name
Description	Returns a module's name
Syntax	$AAM (cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module M refers to the read module name command (cr) is the terminating character, carriage return (ODh)
Response	!AA(Module Name)(cr) if the command is valid ! is a delimiter character indicating that a valid command was received There is no response if the module detects a syntax, communication error, or if the address does not exist AA (range 00-FF) is the 2-character hex address of the specified module (Module Name) is the name of the specified module (cr) is the terminating character, carriage return (ODh)

$AA7CiRrr

Name	Configure single channel type/range
Description	Sets the input type/range of channel N
Syntax	$AA7CiRrr(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 7 refers to the configure single channel type/range command Ci represents the specified input channel you wish to configure Rrr represents the type and range you wish to set (please refer to Table 4-1 to check the range code) (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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) represents the 2-character hex address of the specified 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°C.

$AA8Ci

Name	Read single channel type/range
Description	Returns the channel type/range of channel N
Syntax	$AA8Ci(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 8 refers to the read single channel type/range command Ci represents the specific input channel you wish to read (cr) is the terminating character, carriage return (0Dh)
Response	!AACiRrr(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 indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) represents the 2-character hex address of the specified module Ci represents the specified input channel to be read Rrr represent the type/range setting of the specified channel (see Table 4-1 for range codes) (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.

$AAXnnnn

Name	Configure watchdog timer
Description	Sets the watchdog timer communication cycle time
Syntax	$AAXnnnn(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module X refers to the configure watchdog timer command nnnn (range 0000~9999) represents the value of the communication cycle (0000 = disabled) (cr) is the terminating character, carriage return (0Dh)
Response	!AA(cr) if the command is valid ?AA(cr) if an invalid operation was entered There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) represents the 2-character hex address of the specified module (cr) represents terminating character, carriage return (0Dh)
Example	command: $02X1234(cr) response: 02(cr) The command sets the watchdog timer cycle as 1234 from Address 02 of the input module

$AAY

Name	Read watchdog timer
Description	Returns the watchdog timer communication cycle time
Syntax	$AAY(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module Y represents the read watchdog timer command (cr) is the terminating character, carriage return (0Dh)
Response	!AAnnnn(cr) if the command is valid ?AA(cr) if an invalid operation was entered There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) represents the 2-character hex address of the specified module nnnn (range 0000~9999) represents the communication cycle value (cr) represents terminating character, carriage return (0Dh)
Example	command: $02Y(cr) response: !020030(cr) The command read the watchdog timer cycle as 0030 from Address 02 of the input module.

#AAMC

Name	Get auto-filter sample rate
Description	Returns the auto-filter sample rate
Syntax	#AAMC(cr) # is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module MC refers to the get auto-filter sample rate command (cr) is the terminating character, carriage return (0Dh)
Response	!AAmmm(cr) if the command is valid ?AA(cr) if an invalid operation was entered There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) represents the 2-character hex address of the specified module mmm represents the analog input 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.

#AAMKmm

Name	Configure software filter multiplexing status
Description	Sets the software filter multiplexing status of all channels to enable/ disable
Syntax	#AAMKmm(cr) # is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module MK refers to the configure software filter multiplexing status command mm are the two hexadecimal values interpreted as two binary words (4-bit). The first word represents the status of Channels 4~7, and the second word represents the status of Channels 0~3 (0 = disabled, 1 = 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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) represents the 2-character hex address of the specified module (cr) represents terminating character, carriage return (0Dh)
Example	command: $01MK23(cr) response: !01(cr) 2 (hex) = 0010 (binary), which enables the software filter of Channel 5 and disables Channels 4~7. 3 (hex) = 0011 (binary), which enables the software filter of Channels 0 and 1 and disables Channels 2 and 3.

#AAMD

Name	Read software filter multiplexing status
Description	Returns the software filter multiplexing status (enable/disable) of all channels
Syntax	#AAMD(cr) # is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module MD refers to the read software filter multiplexing status command (cr) is the terminating character, carriage return (0Dh)
Response	!AAmm (cr) if the command is valid ?AA(cr) if an invalid operation was entered There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) represents the 2-character hex address of the specified module mm are the 2 hex values that are interpreted as two binary words (4-bit). The first word represents the status of Channels 4~7, and the second word represents the status of Channels 0~3 (0 = disabled, 1 = enabled). (cr) represents terminating character, carriage return (0Dh)

Example	command: $01MD(cr) response: !0132(cr) 3 (hex) = 0011 (binary), which enables the software filter of Channels 4 and 5 and disables Channels 6 and 7. 2 (hex) = 0010 (binary), which enables the software filter Channel 1 and disables Channels 0, 2, and 3.

$AAFQm

Name	Locate module
Description	Locates a module by activating its locate LED
Syntax	#AAFQm(cr) # is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module FQ refers to the locate module command m: 1 LED Turn ON (5 minutes) 0 Clear Locate (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, communication error, or if the address does not exist > is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) represents the 2-character hex address of the specified module (cr) represents terminating character, carriage return (0Dh)
Example	command: $01FQ1(cr) response: >01 (cr) The Status LED of the module at Address 01 lights for 10 s.

$AA3

Name	Read CJC sensor value
Description	Returns the CJC sensor value
Syntax	$AA3(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 3 refers to the read CJC sensor value 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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module (data) is the CJC sensor value. The data format (unit: °C) consists of a "+" or "-" sign followed by five decimal digits and a fixed decimal point (increment: 0.1°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 module responds with 36.8°C.

$AA9SNNNN

Name	CJC offset calibration
Description	Calibrates a module to adjust for offset errors in its CJC sensor
Syntax	$AA9SNNNN(number of counts)(cr). $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 9 refers to the CJC offset calibration command S is a "+" or "-" sign, indicating whether to increase or decrease the CJC offset value NNNN(number of counts) is a 4-character hex "count" value in increments of approximately 0.009°C (range 0000~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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified 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 by 42 (hex) = 66 (decimal) counts, which is approximately 0.6°C

Note! *An analog input module requires up to 2s to perform auto calibration and ranging after it has received a CJC offset calibration command. During this time, the module cannot be addressed to perform any other actions.*

$AA0Ci

Name	Single Channel Span Calibration command
Description	The command calibrates a specified channel to correct for gain errors.
Syntax	$AA0Ci(cr) $ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be calibrated. 0 is the Single Channel Span Calibration command. Ci represents the specified input channel you want to calibrate. (cr) is the terminating character, carriage return (0Dh).
Response	!AA(cr) if the command was valid. ?AA(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the specified address does not exist. ! delimiter character indicates a valid command was received. ? delimiter character indicates the command was invalid. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module. (cr) represents terminating character, carriage return (0Dh). In order to successfully calibrate an analog input module's input range, a proper calibration input signal should be connected to the analog input module before and during the calibration. (See also Chapter 8, Calibration)

$AA1Ci

Name	Single Channel Offset Calibration command
Description	The command calibrates a specified channel to correct for offset errors.
Syntax	$AA1Ci(cr) $ is a delimiter character. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module which is to be calibrated. 1 is the Single Channel Offset Calibration command. Ci represents the specified input channel you want to calibrate. (cr) is the terminating character, carriage return (0Dh).
Response	!AA(cr) if the command was valid. ?AA(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the specified address does not exist. ! delimiter character indicates a valid command was received. ? delimiter character indicates the command was invalid. AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module. (cr) represents terminating character, carriage return (0Dh).
Example	command: $021C5(cr) response: !02(cr) The command calibrates channel 5 of the analog input module at address 02 for correcting offset errors.

4.5 Digital I/O Module Command Search Table

Table 4.7: ADAM-4150 and ADAM-4168 Command Table

Command Syntax	Command Name	Command Description
%AANNTTCCFF	Configuration	Sets a module's address, input range, baud rate, data format, checksum status, and/or integration time
$AA2	Configuration status	Returns a module's configuration data
$AA6	Digital data in	Returns the statuses of a module's digital input channels and the values of its digital output channels
#AABB(data)	Digital data out	Writes a value to either one or all digital output channels
$AAF	Get firmware version	Return a module's firmware version
$AAM	Get module name	Return the module name
$AAX0TTTTDD	Write safety value	Writes the safety status and time-out value for all channels
$AAX1	Read safety value	Returns the safety status and time-out period of all channels
$AAX2	Read safety flag	Returns a module's safety flag status to determine whether the safety value has been executed since the last write safety value command was set
$AACIIIIIIIIIIIIIOOOOOOOOOOOOOOOO	Set all digital I/O channel statuses	Forces a module's digital I/O channels to a different mode
$AAC	Read all digital I/O channel statuses	Returns the statuses of all digital I/O channels
$AACICjII	Set single digital input channel status*	Forces a digital input channel to a different mode
$AACICj	Read single digital input channel status*	Returns the status of a digital input channel
$AACOCjOO	Set single digital output channel status	Forces a digital output channel to a different mode
$AACOCj(cr)	Read single digital output channel status	Returns the status of a digital output channel
$AA0CjLLLLLLLLHHHHHHHH	Set digital input filter width*	Sets the width of a module's digital input filter
$AA0Cj	Read digital input filter width*	Returns the width of a module's digital input filter
$AA9n(lw)(hw)(ld)(hd)	Set single pulse output width	Sets the pulse output width for channel N
$AA9n	Read pulse output input width	Returns the pulse output width of channel N
#AAN	Read counter/frequency value*	Returns a module's counter or frequency value from Counter 0 or 1
#aaERFFccvvvvvvvv	Set pulse output count	Sets the pulse output count
$aaERFFcc	Read pulse output count	Returns the pulse output count
@AACACj	Clear latch alarm*	Both the high and low alarm states of the counter module are set to OFF
$AA5NS	Start/stop counter*	Requests the counter/frequency module to start or stop counting for Counter 0 or 1

Table 4.7: ADAM-4150 and ADAM-4168 Command Table

$AA5N	Read counter start/ stop status*	Requests the counter/frequency module to indicate whether Counter 0 or 1 is active
$AA6N	Clear counter*	Clear counter
#AAFQm	Locate module	Locate the module

*These commands apply only to the ADAM-4150

4.6 Digital I/O Module Command Set

%AANNTTCCFF

Name Basic module Configuration

Description Sets a module's address, baud rate, and/or checksum status

Syntax %AANNTTCCFF(cr)
% is a delimiter
AA (range 00-FF) is the current 2-character hex address of the specified module
NN is the new hex address you wish to assign to the module (range 00h~FFh)
TT represents the type code (always 40 for digital I/O modules)
CC represents the baud rate code (see Table 4-3)
FF is a hex number equal to an 8-bit parameter that represents the checksum status and protocol
Bits 3~5 as well as Bits 0, 1, and 7 are not used and are set to 0 (see Figure 4-2)
Bit 6 is the checksum and Bit 2 is the protocol (0: Advantech, 1: Modbus)
(cr) is the terminating character, carriage return (0Dh)

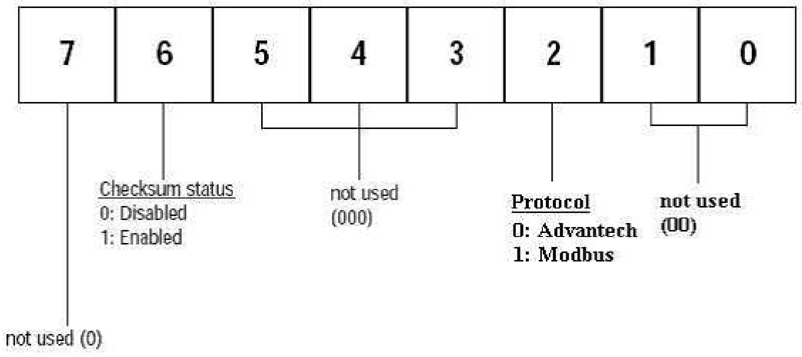


Figure 4.2 Checksum and Protocol

Response !AA (cr) if the command is valid
?AA(cr) if an invalid parameter was entered or if the INIT* terminal was not grounded when attempting to changing the baud rate or checksum settings
There is no response if the module detects a syntax, communication error, or if the address does not exist
! is a delimiter indicating that a valid command was received
? is a delimiter indicating that the command was invalid
AA (range 00-FF) is the 2-character hex address of the specified 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 4.8: 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

Note! *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.*

$AA2

Name	Configuration status
Description	Returns a module's configuration data
Syntax	$AA2(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 2 refers to 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 command has been issued There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module TT represents the type of code (always 40) CC represents the baud rate code FF is a hexadecimal number equal to an 8-bit parameter that represents the checksum status and protocol Bits 3~ 5 as well as Bits 0, 1, and 7 are not used and are being set to 0 (see Figure 4-3) Bit 6 is the checksum and Bit 2 is the protocol (0: Advantech, 1: Modbus) (cr) is the terminating character, carriage return (ODh)

Example	command: $452 (cr) response: !45400600 (cr) The command asks the digital I/O module at Address 45h to send its configuration data. The module responds with a baud rate of 9600 and no checksum function. The response also indicates that the module supports the Advantech protocol.

$AA6

Name	Digital data in
Description	Returns the statuses of a module's digital input channels and the values of its digital output channels
Syntax	$AA6(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 6 refers to the digital data in command (cr) is the terminating character, carriage return (0Dh)
Response	!(dataOutput)(dataInput)00(cr) if the command is valid (ADAM-4150) !(dataOutput)0000(cr) if the command is valid (ADAM-4168) ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module (dataOutput) is a 2-character hex value that is the feedback of either a digital output channel or a relay (dataInput) is a 2-character hex value representing the input values of the specified module (cr) is the terminating character, carriage return (0Dh)
Example	command: $336(cr) response: !112200(cr) This example is for the ADAM-4150. The first two characters of the response, 11 (hex) = 00010001 (binary), indicate that Channels 0 and 4 are ON whereas Channels 1~3 and 5~7 are OFF. The following two characters of the response, 22 (hex) = 00100010 (binary), indicate that Channels 1 and 5 are both HIGH whereas channels 0, 2~4, 6, and 7 are LOW.

#AABB

Name	Digital data out
Description	Writes a value to one or all digital output channels

Syntax

#AABB(data)(cr)
# is a delimiter
AA (range 00-FF) is the 2-character hex address of the specified module
BB indicates whether one or all channels will be set (the last character indicates which channel)
- Writing to all channels (write a byte): both characters should be equal to zero (BB=00)
- Writing to one channel (write a bit): first character is 1, and the second character is the channel number (range 0~B)

(data) is the hex representation of the digital output value(s)
- Writing to all channels (byte): both characters are significant (range 00h~FFh); the decimal equivalent of these 2 hex characters represents the channel values
- Writing to one channel (bit): the first character is always 0, and the second character is either 0 or 1

As an example, the value 7A can be converted as follows to represent the 8 channels of the 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

Response

>(cr) if the command is valid
?AA(cr) if an invalid command was issued
There is no response if the module detects a syntax, communication error, or if the address does not exist
> is a delimiter indicating that a valid command was received
? is a delimiter indicating that the command was invalid
AA (range 00-FF) is the 2-character hex address of the specified module
(cr) is the terminating character, carriage return (0Dh)

Example

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.

Example

command: #151201(cr)
response: >(cr)
An output bit with a value of 1 is sent to Channel 2 of a digital I/O module at Address 15h. Channel 2 of the digital I/O module is set to ON.

$AAX0TTTTDD

Name Write safety value

Description Writes the safety status and time-out period for all channels

Syntax

$AAX0TTTTDD(cr)
$ is a delimiter
AA (range 00-FF) is the 2-character hex address of the specified module
X0 refers to the write safety value command
TTTT is the time (increment: 100 ms); the watchdog timer is off when TTTT = 0
DD is a 2-character hex value representing the desired input safety value
As an example, 7A would be interpreted as follows:

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

Here, 7A (hex) means that the status is ON for Channels 1, 3, 4, 5, 6; the other channels are OFF
(cr) is the terminating character, carriage return (0Dh)

Response

>(cr) if the command is valid
?AA(cr) if an invalid command was issued
There is no response if the module detects a syntax, communication error, or if the address does not exist
> is a delimiter indicating that a valid command was received
? is a delimiter indicating that the command was invalid
AA (range 00-FF) is the 2-character hex address of the specified module

$AAX1

Name Read safety value

Description Returns the safety status and time-out period of all channels

Syntax

$AAX1(cr)
$ is a delimiter
AA (range 00-FF) is the 2-character hex address of the specified module
X1 refers to the read safety value command

Response

! TTTTDD(cr) if the command is valid
DD is a 2-character hex value representing the desired input safety value
As an example, the meaning of 7A is as follows:

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

Thus, 7A means Channel 1, 3, 4, 5, and 6 are ON; 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)

$AAX2

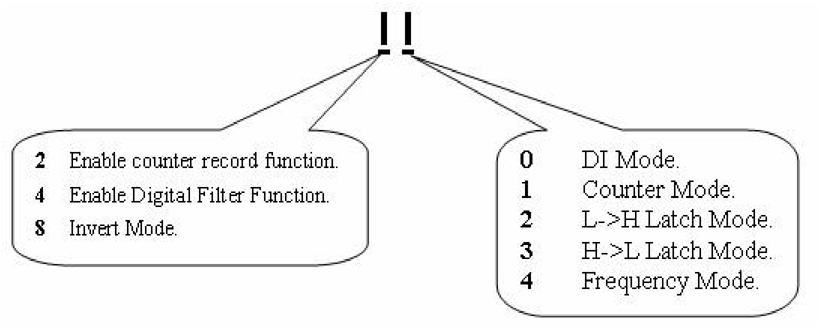
Name Read safety flag

Description Returns a module's safety flag status to determine whether the safety value has been executed since the last write safety value command was set

Syntax	$AAX2(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module X0 is the read safety flag command
Response	!XX (cr) if the command is valid XX is a 2-character hexadecimal value (00: OFF, 01: ON) ?AA(cr) if an invalid command was issued ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid (cr) is the terminating character, carriage return (ODh)

$AACIIIIIIIIIIIIIIOOOOOOOOOOOOOOOO

Name	Set all digital I/O channel statuses
Description	Forces a module's digital I/O channels to a different mode
Syntax	$AACIIIIIIIIIIIIIIOOOOOOOOOOOOOOOO $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module C refers to the set all the digital I/O channel statuses command IIIIIIIIIIIIII denotes the 7 digital input channels (every "II" pair refers to one channel)



OOOOOOOOOOOOOOOO denotes the 8 digital output channels (every "OO" pair refers to 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)

Response

>(cr) if the command is valid
?AA(cr) if an invalid command was issued
There is no response if the module detects a syntax, communication error, or if the address does not exist
> is a delimiter indicating that a valid command was received
? is a delimiter indicating that the command was invalid
AA (range 00-FF) is the 2-character hex address of the specified module

Example

command: $02C00210440000000020000000000010203(cr)
response: >(cr)
The digital I/O channels at Address 2 are set as follows:

II Character Mode	II Character	Mode
0	00	DI mode
1	21	Counter mode + enable counter record
2	04	Frequency mode
3	40	Enable digital filter
4	00	DI mode
5	00	DI mode
6	00	DI mode

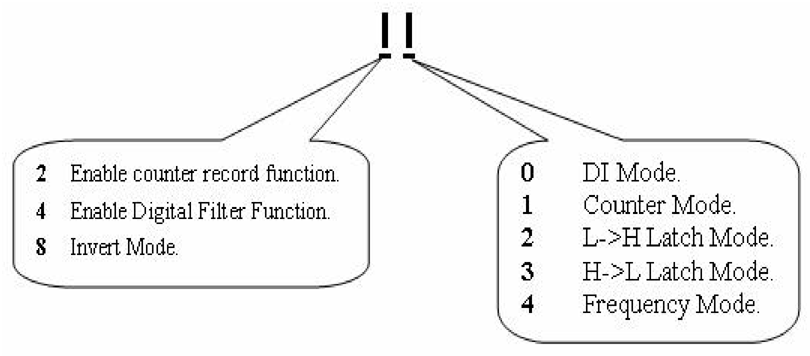
DO channel	OO Character	Mode
0	02	L>H delay mode
1	00	DO mode
2	00	DO mode
3	00	DO mode
4	00	DO mode
5	01	Pulse output mode
6	02	L>H delay mode
7	03	H>H delay mode

$AAC

Name	Read all digital I/O channel statuses
Description	Returns the statuses of all digital I/O channels
Syntax	$AAC(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module C refers to the read all digital I/O statuses command (cr) is the terminating character, carriage return (0Dh)
Response	!AAIIIIIIIIIIIIIIOOOOOOOOOOOOOOOO(cr) if the command is valid ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received IIIIIIIIIIIIIIOOOOOOOOOOOOOOOO is defined the same as $AACIIIIIIIIIIIIIIOOOOOOOOOOOOOOOO ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module IIIIIIIIIIIIII is to set the 7 DI channels (every "II" pair refers to one channel). Refer to command $AACIIIIIIIIIIIIIIOOOOOOOOOOOOOOOO for further information. OOOOOOOOOOOOOOOO is for setting the 8 digital output channels OO=00 DO mode OO=01 Pulse output mode OO=02 L->H delay mode OO=03 H->L delay mode

$AACICjII

Name	Set single digital input channel status
Description	Forces a digital input channel to a different mode
Syntax	$AACICjII(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module CI refers to the set single digital input channel status command Cj denotes channel j II is two characters for setting the digital input mode



Response	(cr) is the terminating character, carriage return (0Dh) >(cr) if the command is valid ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist > is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module
Example	command: $02CIC202(cr) response: >(cr) Channel 2 is set to low-to-high latch mode

$AACICj

Name	Read single digital input channel status
Description	Returns the status of a digital input channel
Syntax	$AACICj(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module CI refers to the read single digital input channel status command Cj denotes channel j (cr) is the terminating character, carriage return (0Dh)
Response	!AAII if the command is valid ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received II is defined the same as $AACICjII ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module

$AACOCjOO

Name	Set single digital output channel status
Description	Forces a digital output channel to a different mode
Syntax	$AACOCjII(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module CO refers to the set single digital output channel status command Cj denotes channel j II is two characters for setting the digital output 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 is valid ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist > is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module
Example	command: $02COC201(cr) response: >(cr) Channel 2 is set to pulse output mode

$AACOCj

Name	Read single digital output channel status
Description	Returns the status of a digital output channel
Syntax	$AACOCj(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module CO refers to the read single digital output channel status command Cj denotes channel j (cr) is the terminating character, carriage return (0Dh)
Response	!AAOO if the command is valid ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received OO is defined the same as $AACOCjOO ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module

$AA0CjLLLLLLLLHHHHHHHH

Name	Set digital input filter width
Description	Sets the width of a module's digital input filter
Syntax	$AA0CjLLLLLLLLHHHHHHHH(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 0 is the DI filter input command Cj denotes channel j LLLLLLLL is the low-level time (range 0x0~0xffffffff) HHHHHHHH is the high-level time (range 0x0~0xffffffff) Increment: 0.1 ms The digital filter function works on counter mode and can set the minimum width of the low/high signal to filter unwanted noise (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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module

$AA0Cj

Name	Read digital input filter input width
Description	Returns the width of a module's digital input filter
Syntax	$AA0Cj (cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 0 refers to the read digital input filter width command Cj denotes channel j (cr) is the terminating character, carriage return (0Dh)
Response	!AALLLLLLLLHHHHHHHH(cr) if the command is valid ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received. LLLLLLLL is the low-level time (range 0x0~0xffffffff) HHHHHHHH is the high-level time (range 0x0~0xffffffff) Increment: 0.1 ms ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module

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

Name	Set single pulse output width
Description	Sets the pulse output width for channel N
Syntax	$AA9n(lw)(hw)(ld)(hd)(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 9 refers to the set single pulse output width command n denotes channel N (lw) is an 8-character hex value for low width=>0x0~0xffffffff (hw) is an 8-character hex value for high width=>0x0~0xffffffff (ld) is an 8-character hex value for low delay=>0x0~0xffffffff (hd)is an 8-character hex value for high delay=>0x0~0xffffffff Increment: 0.1 ms (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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module

$AA9n

Name	Read single pulse output width
Description	Returns the pulse output width of channel N
Syntax	$AA9n (cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 9 refers to the read single pulse output width command n denotes the channel number (cr) is the terminating character, carriage return (0Dh)
Response	!AA(lw)(hw)(ld)(hd)(cr) if the command is valid ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received (lw) is an 8-character hex value for low width=>0x0~0xffffffff (hw) is an 8-character hex value for high width=>0x0~0xffffffff (ld) is an 8-character hex value for low delay=>0x0~0xffffffff (hd) is an 8-character hex value for high delay=>0x0~0xffffffff Increment: 0.1 ms ? is a delimiter character indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module

#AAN

Name	Read counter/frequency value
Description	Returns a module's counter or frequency value from Counter 0 or 1
Syntax	#AAN(cr) # is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module N denotes 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, communication error, or if the address does not exist ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module (data) is the module's counter value. The data format consists of 8 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 Counter 0 and to return the data. The module responds with value of 000002FE (hex) = 766 (decimal).

#AAERFFccvvvvvvvv

Name	Set pulse output count
Description	Sets the pulse output count
Syntax	#AAERFFccvvvvvvvv (cr) # is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module ERFF refers to the set pulse output count command cc is the channel number (00~07 means Channels 0~7) vvvvvvvv is an 8-hex character representing the pulse output count (0 = continuous pulse output) (cr) is the terminating character, carriage return (0Dh)
Response	!AA if the command is valid ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was received ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2-character hex address of the specified module

$AAERFFcc

Name	Read pulse output count
Description	Returns the pulse output count
Syntax	$AAERFFcc (cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module ERFF refers to the read pulse output count command cc is the channel number (00~07 means Channels 0~7) (cr) is the terminating character, carriage return (0Dh)
Response	>AAmvvvvvvvv if the command is valid ?AA(cr) if an invalid command was issued There is no response if the module detects a syntax, communication error, or if the address does not exist > is a delimiter indicating that a valid command was received m = 0 means non-continue mode m = 1 means continue mode vvvvvvvv is an 8-character hex value for the pulse output count (0 = continuous pulse output) ? is a delimiter indicating that the command was invalid AA (range 00-FF) is the 2- character hex address of the specified module

@AACACj

Name	Clear latch alarm
Description	Both the high and low alarm states of the counter module are set to OFF
Syntax	@AACACj(cr) @ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module CA refers to the clear latch alarm command Cj denotes channel j (cr) represents 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, communication error, or if the address does not exist ! is a delimiter indicating that a valid command was valid AA is the 2-character hex address of the specified 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.

$AA5NS

Name	Start/stop counter
Description	Requests the counter/frequency module to start or stop counting for Counter 0 or 1
Syntax	$AA5NS(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 5 refers to the start/stop counter command N notes whether the counter should be enabled or disabled (0 = Counter 0, 1 = Counter 1) S represents the counter status (0 = stop counting, 1 = start 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, communication error, or if the address does not exist ! is a delimiter indicating that the command was valid AA (range 00-FF) is the 2-character hex address of the specified 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.

$AA5N

Name	Read counter start/stop status
Description	Requests the counter/frequency module to indicate whether Counter 0 or 1 is active
Syntax	$AA5N(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module N indicates which counter is active (0 = Counter 0, 1 = 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, communication error, or if the address does not exist ! is a delimiter indicating that the command was valid AA (range 00-FF) is the 2-character hex address of the specified module S represents the counter status (0 = counting, 1 = 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 module replies that Counter 0 is counting.

$AA6N

Name	Clear counter
Description	Requests the counter/frequency module to clears Counter 0 or 1
Syntax	$AA6N(cr) $ is a delimiter AA (range 00-FF) is the 2-character hex address of the specified module 6 refers to the clear counter command N determines which the counter should be cleared (0 = Counter 0, 1 = 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, communication error, or if the address does not exist ! is a delimiter indicating that the command was valid AA (range 00-FF) is the 2-character hex address of the specified 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.

Chapter 5

Hardware Installation Guide

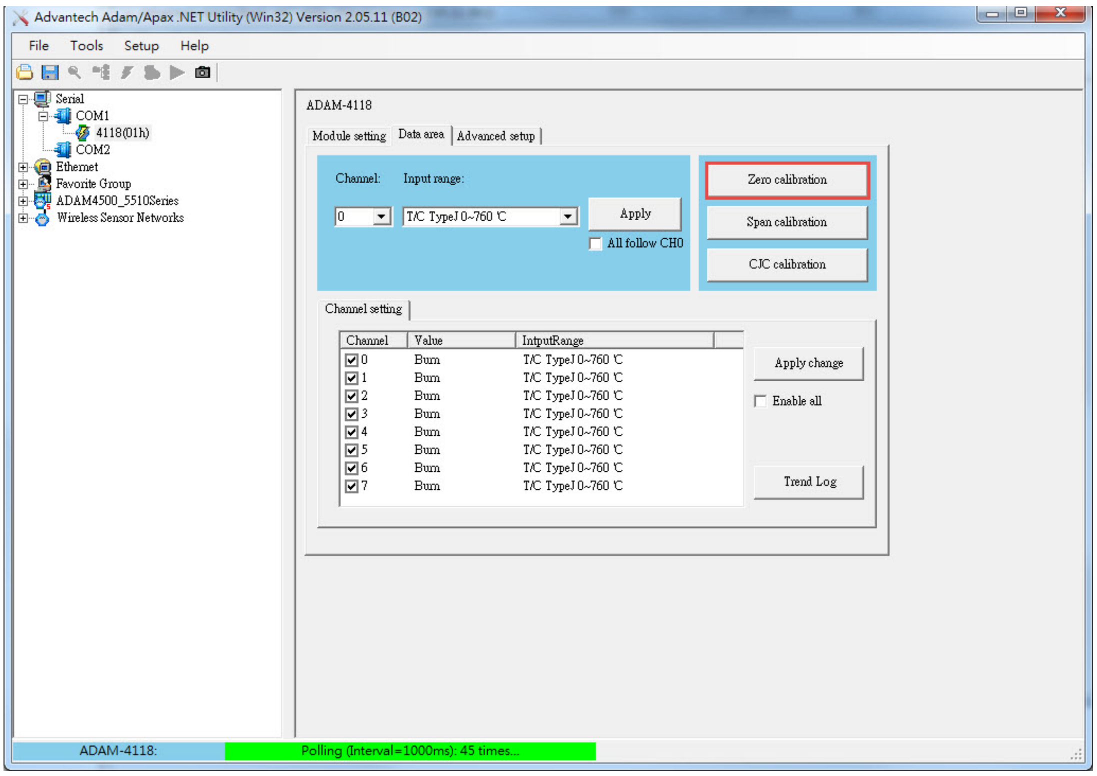
Analog input modules are already calibrated when you receive them. However, calibration is occasionally required and you can do this via software. Calibration parameters are stored in the ADAM module's onboard EEPROM.

ADAM modules come with utility software that allows you to calibrate a module's inputs and outputs. Aside from the calibration that is carried out through the software, the modules incorporate automatic zero calibration and automatic span calibration upon 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 min.
2. Ensure that the module is correctly installed and is properly configured for the input range that you want to calibrate. You can do this /Apax .NET Utility (see Appendix A)
3. Enable the auto calibration function.



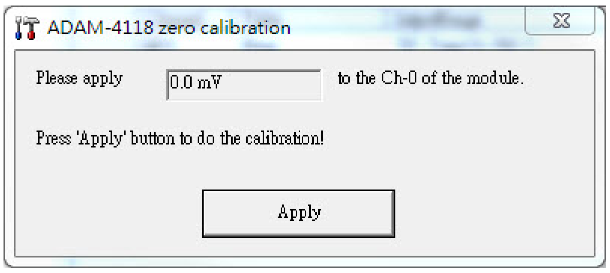
Channel: 0

Input range: T/C Type J 0~760 ℃

Channel	Value	IntputRange
☑ 0	Burn	T/C Type J 0~760 ℃
☑ 1	Burn	T/C Type J 0~760 ℃
☑ 2	Burn	T/C Type J 0~760 ℃
☑ 3	Burn	T/C Type J 0~760 ℃
☑ 4	Burn	T/C Type J 0~760 ℃
☑ 5	Burn	T/C Type J 0~760 ℃
☑ 6	Burn	T/C Type J 0~760 ℃
☑ 7	Burn	T/C Type J 0~760 ℃

ADAM-4118: Polling (Interval=1000ms): 45 times...

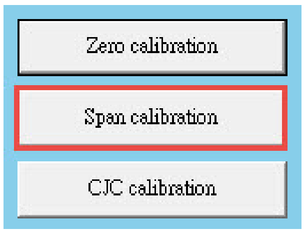
4. Execute the zero calibration command. This is also done through Adam/Apax .NET Utility. Apply the indicating signal to the input channel then save the exact value.

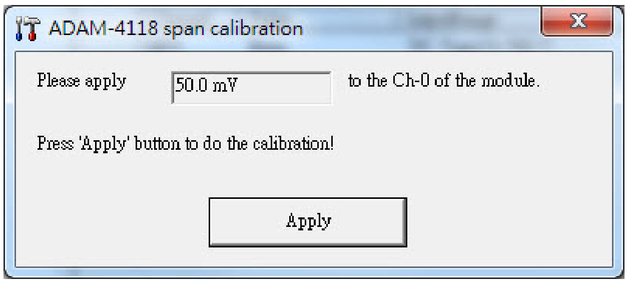


Please apply 0.0 mV to the Ch-0 of the module.

Press 'Apply' button to do the calibration!

5. Execute the span calibration command. This can be done with Adam/Apax .NET Utility. Apply the indicating signal to the input channel then save the exact value.

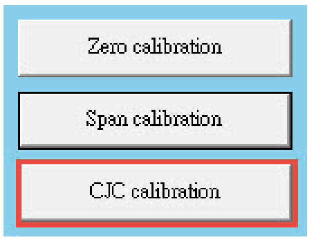


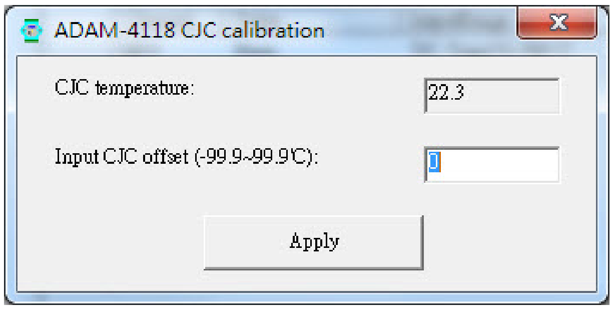

ADAM-4118 span calibration

Please apply 50.0 mV to the Ch-0 of the module.

Press 'Apply' button to do the calibration!

6. For the ADAM-4118 only, execute the CJC calibration command. This is also done through Adam/Apax .NET Utility. You can use CJC offset to adjust the exact temperature. For example, if the input signal is 24°C but the reading is 23.8°C, CJC offset can be set to +0.2°C to compensate.




ADAM-4118 CJC calibration

CJC temperature: 22.3

Input CJC offset (-99.9~99.9℃): 0

Input Range	Typical Accuracy	Maximum Error	Unit
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

Appendix A

Utility Software Overview

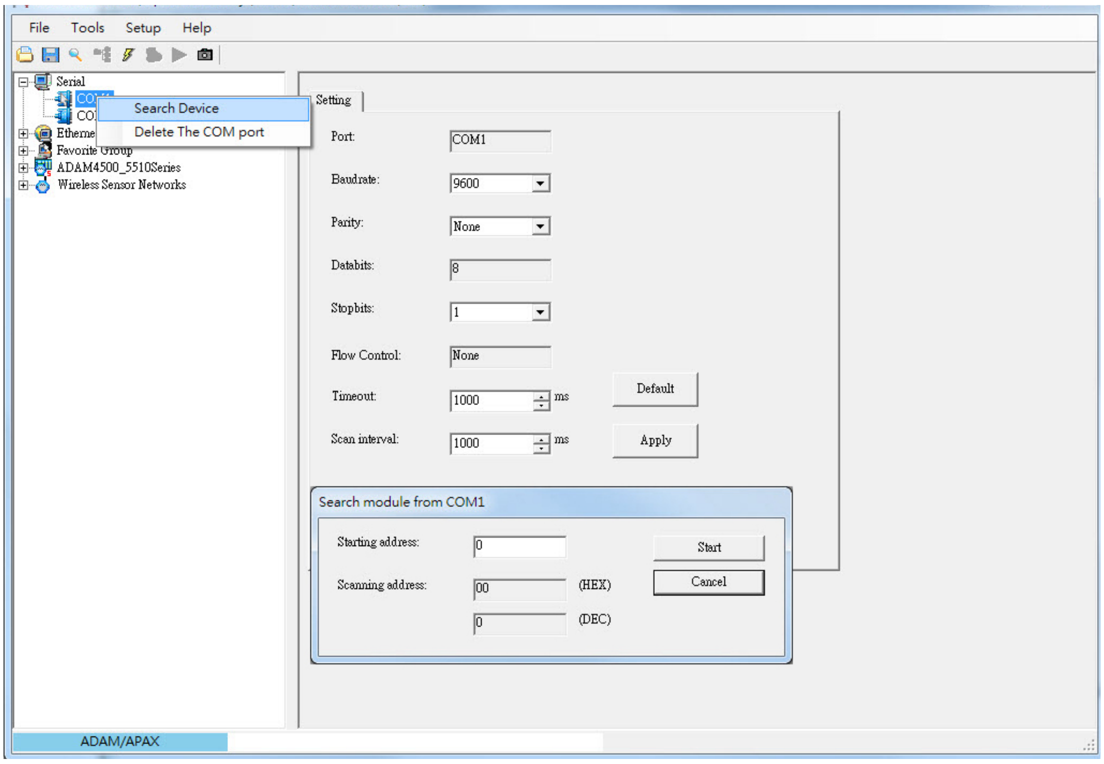
ADAM modules can be configured /Apax .NET Utility, a configuration utility software package that gives you the following capabilities:

- Module configuration
- Module calibration
- Data I/O monitoring
- Auto-scan of connected modules
- Terminal emulation

The following text provides brief instructions on how to use the software.

A.1 Searching for Installed Modules

The main screen this window has four main areas: 1) the Menu Bar, 2) the Toolbar, 3) the Module Tree Display Area, and 4) the Status Display Area. The Status Display Area is the main area that shows information about the connected modules. Once your module is connected, start the program and search for the module by selecting **Search Device** from the **Tools** menu, by clicking the **Search Modules** icon 🔍 on the Toolbar, or by right-clicking on the COM port in the Module Tree Display Area and then selecting **Search Device**. You should check whether the COM port and related settings are correct.



Port: COM1
Baudrate: 9600
Parity: None
Databits: 8
Stopbits: 1
Flow Control: None
Timeout: 1000 ms
Scan interval: 1000 ms

Search module from COM1
Starting address: 0
Scanning address: 00 (HEX)
0 (DEC)

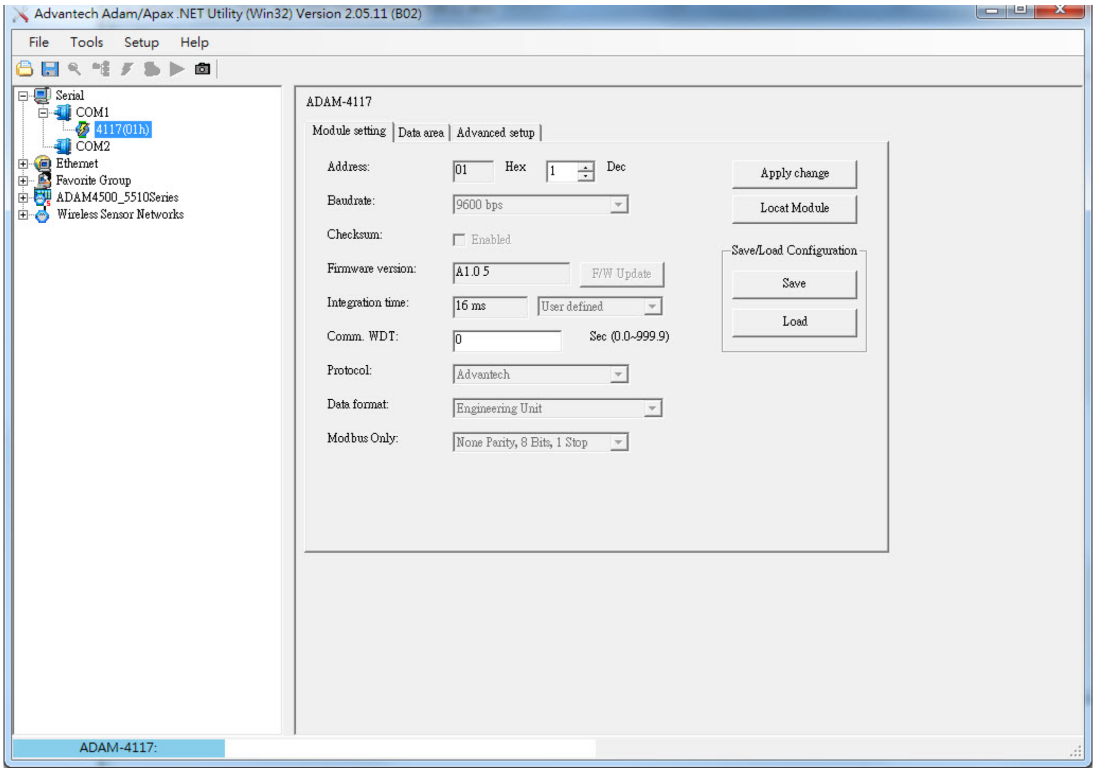
Figure A.1 Search Device Screen

Note! *Whenever you configure or calibrate a module or its alarm parameters, a window will appear to confirm the changes to the target module.*

An asterisk “*” before a module's address indicates that the module is in the INIT* state.

A.2 Module Configuration

From the Module Tree Display Area, select a module that you would like to configure. A setup page and related settings will appear in the Status Display Area. The example in Figure A-2 is for an ADAM-4117.



Address: 01 Hex 1 Dec

Baudrate: 9600 bps

Checksum: Enabled

Firmware version: A1.05

Integration time: 16 ms User defined

Comm. WDT: 0 Sec (0.0~999.9)

Protocol: Advantech

Data format: Engineering Unit

Modbus Only: None Parity, 8 Bits, 1 Stop

Figure A.2 Configuration Screen

There are three major areas in the status field: General Settings, Calibration Area, and Channel Settings. To apply any changes to settings, click **Apply change**.

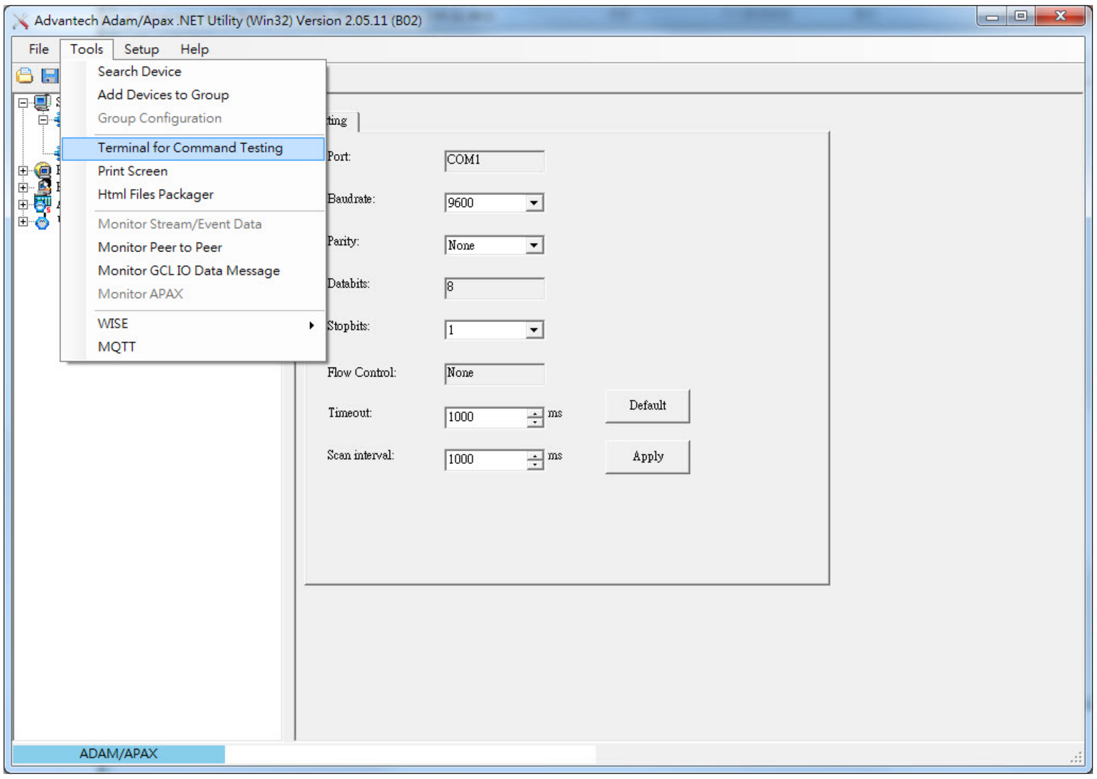
The **Checksum** and **Baud rate** fields require special attention since they can only be changed when an ADAM module is in the INIT state. After you have made all necessary changes to a module, the utility will automatically display the processed data.

The data format is as follows:

- 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)

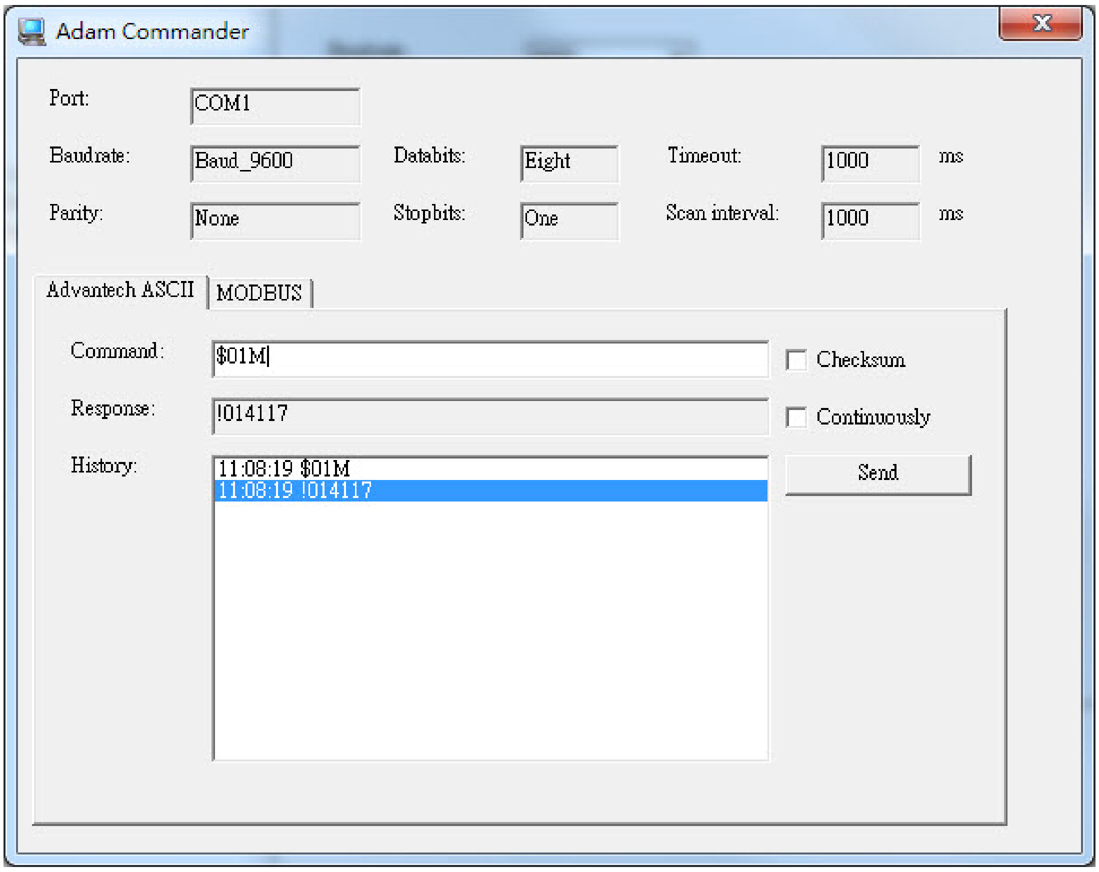
A.3 Terminal Function

When you would like to send and receive commands via RS-485, you can use the Terminal function, which can be accessed by pressing **Terminal for Command Testing** under the **Tools** menu.



Port: COM1
Baudrate: 9600
Parity: None
Databits: 8
Stopbits: 1
Flow Control: None
Timeout: 1000 ms
Scan interval: 1000 ms

You can enter an ASCII command in the **Command** box and click **Send** to test the commands, which are listed in Chapter 4.



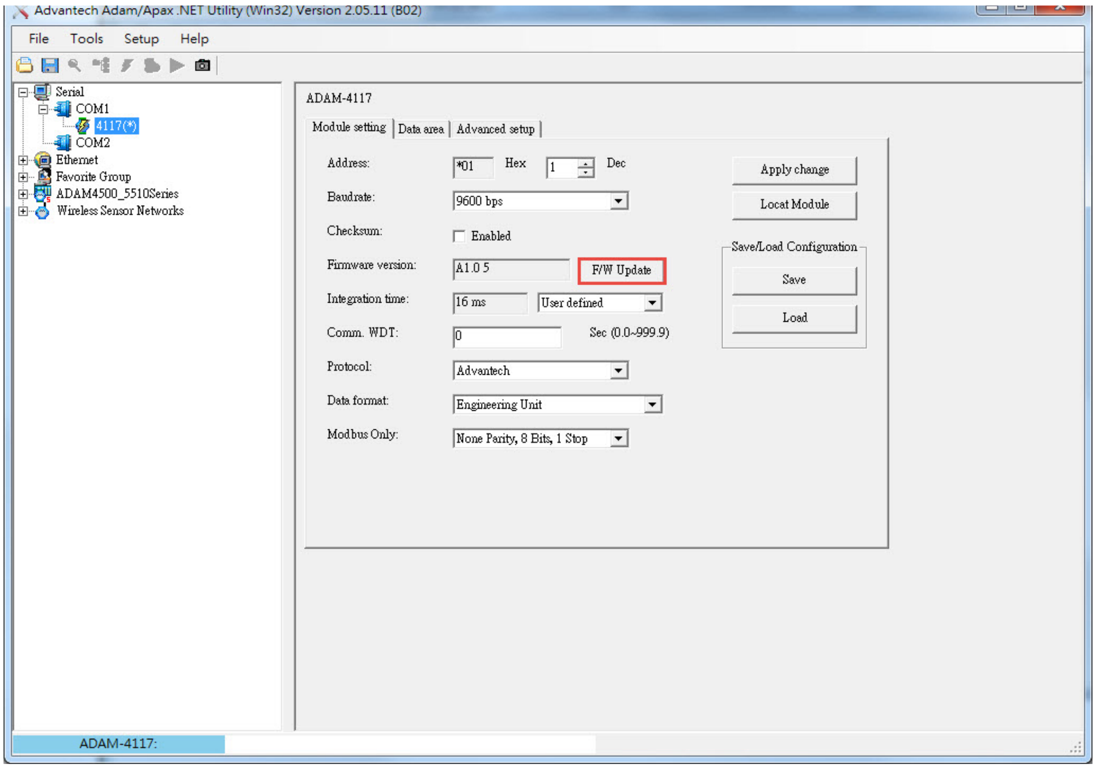
Port: COM1
Baudrate: Baud_9600
Databits: Eight
Timeout: 1000 ms
Parity: None
Stopbits: One
Scan interval: 1000 ms

Command: $01M
Checksum
Response: !014117
Continuously
History:
11:08:19 $01M
11:08:19 !014117

A.4 Firmware Updates

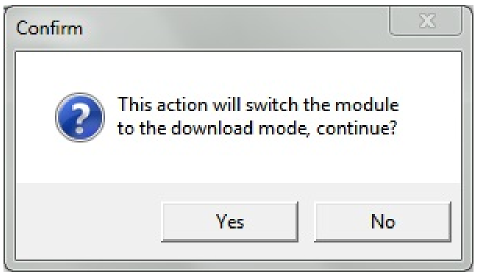
In contrast to older ADAM modules, the firmware of newer ADAM-4100 series modules can be updated online. Simply follow these steps:

1. Configure the ADAM module to be in the Initial state and then click **F/W Update**

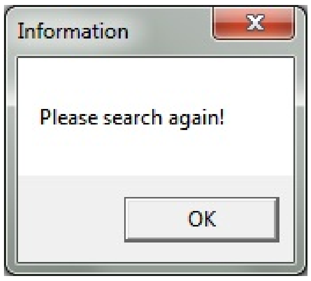


Address: *01 Hex 1 Dec
Baudrate: 9600 bps
Checksum: Enabled
Firmware version: A1.0 5
Integration time: 16 ms User defined
Comm. WDT: 0 Sec (0.0~999.9)
Protocol: Advantech
Data format: Engineering Unit
Modbus Only: None Parity, 8 Bits, 1 Stop

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

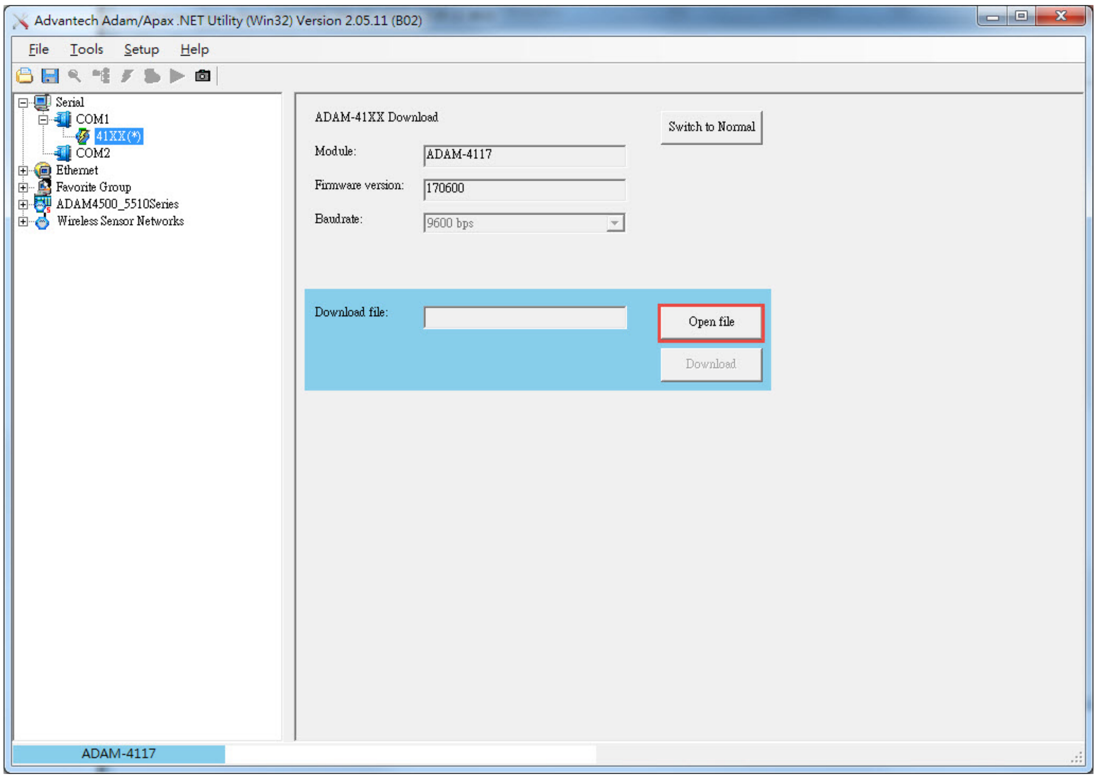


This action will switch the module to the download mode, continue?



Please search again!

3. A new window will appear with the firmware download option. The model name in the Module Tree Display Area will change to 41XX, and the Status Display Area will show a field for you to select the baud rate for the download. Below the baud rate selection, you can choose the firmware file by pressing **Open file**, selecting the file, and then clicking **Download** to download the file to the hardware.



ADAM-41XX Download

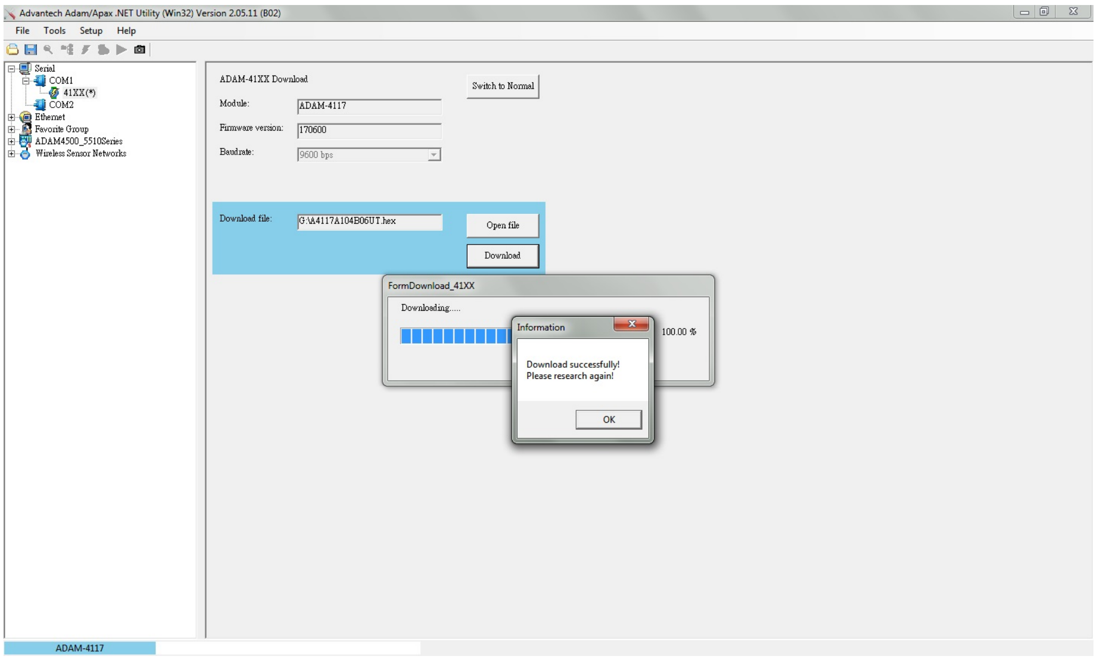
Module: ADAM-4117

Firmware version: 170600

Baudrate: 9600 bps

Download file:

4. Once the download is successful, click **OK**



ADAM-41XX Download

Module: ADAM-4117

Firmware version: 170600

Baudrate: 9600 bps

Download file: G:\A4117A104B06UT.hex

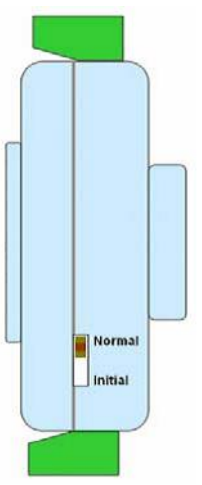
Downloading.....

100.00 %

Download successfully!
Please research again!

A.5 Operating Modes

ADAM-4100 series modules have three status modes: Normal, Initial, and Address mode. This is set by adjusting the switch on the side of the module:



Normal Mode

Once set to normal mode, the module will apply the user-defined settings during operation. A power reset will not alter these settings.

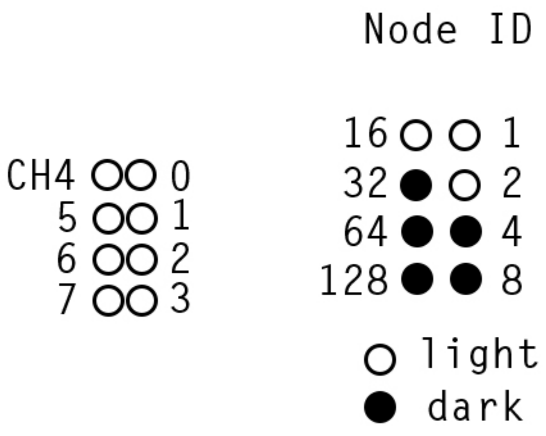
Initial Mode

Once set to initial mode, the module will use factory settings (Address, 0; data format 9600, N, 8, 1).

Note! *If you need to change between Initial and Normal modes, the module needs to be reset for the changes to take effect.*

Address Mode

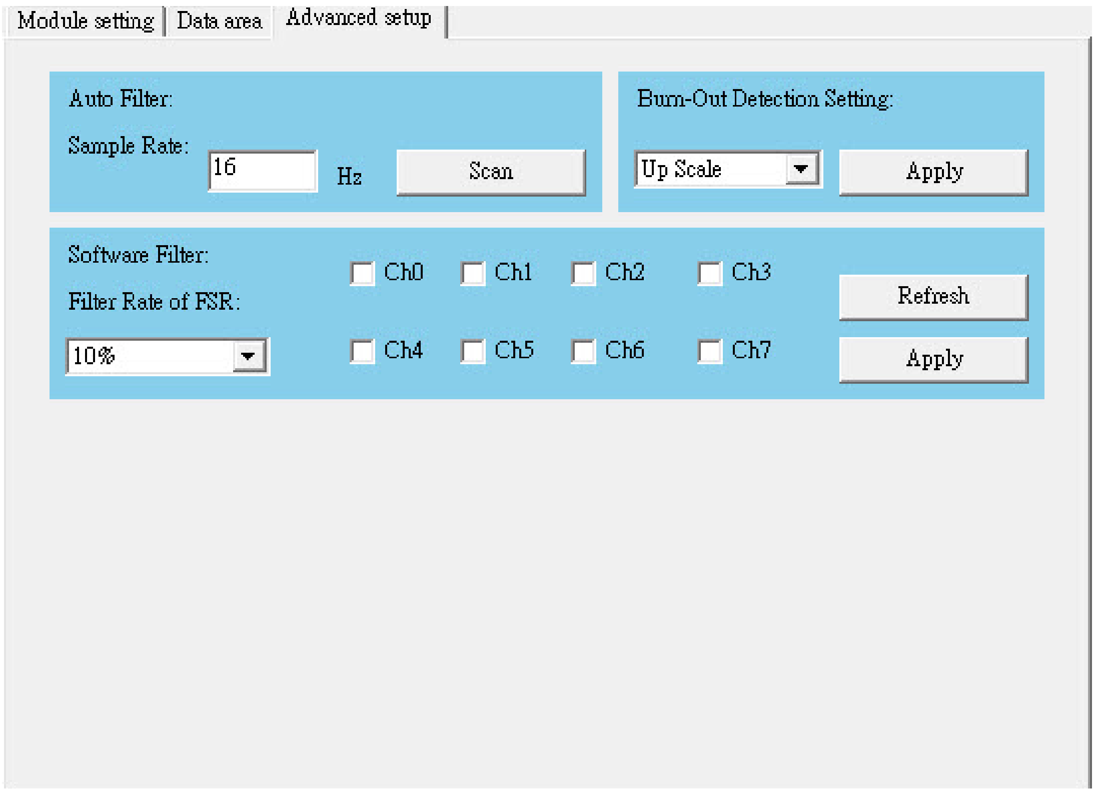
For address mode, turn the switch directly from normal to initial mode without turning off the power. The module will apply the user-defined settings during operation, and the LEDs will show the node ID. These LEDs are commonly used for determining the channel status and in address mode.



This image shows that ID = 19 (13h). Previously, you would have had to have used the utility to check the node ID. Now, address mode can help you read the module address directly.

A.6 Software Filter

Click the **Advanced setup** tab in the Status Display Area to open the filter settings.



Sample Rate: 16 Hz

Burn-Out Detection Setting: Up Scale

Filter Rate of FSR: 10%

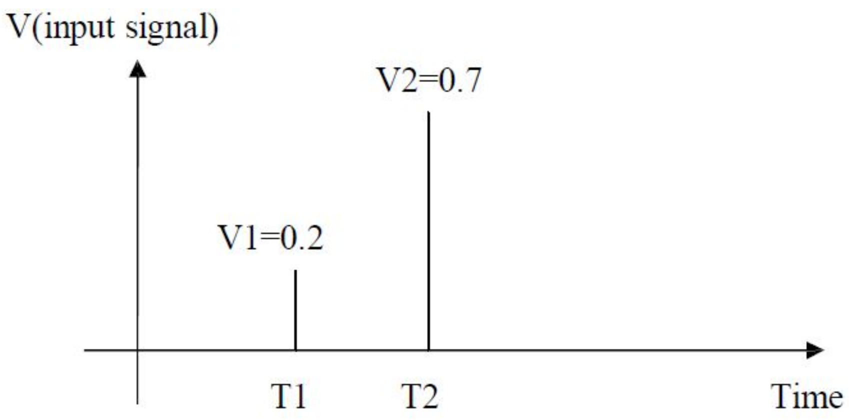
Auto Filter

When integration time is selected, the auto-filter will automatically scan for noise and filter it. The systems will respond with the appropriate setting. If the appropriate setting cannot be found, it will return to 50/60 Hz.

- 50/60 Hz: When the system finds this base, noise at 50/60 Hz will be filtered.
- 100 Hz: When system finds this base, noise at 100 Hz will be filtered.
- Auto Filter: When a fixed voltage is set, for example 5 V, users can find a base which is 10 Hz and filter it. If it cannot be found, 16 Hz will be displayed.

Software Filter

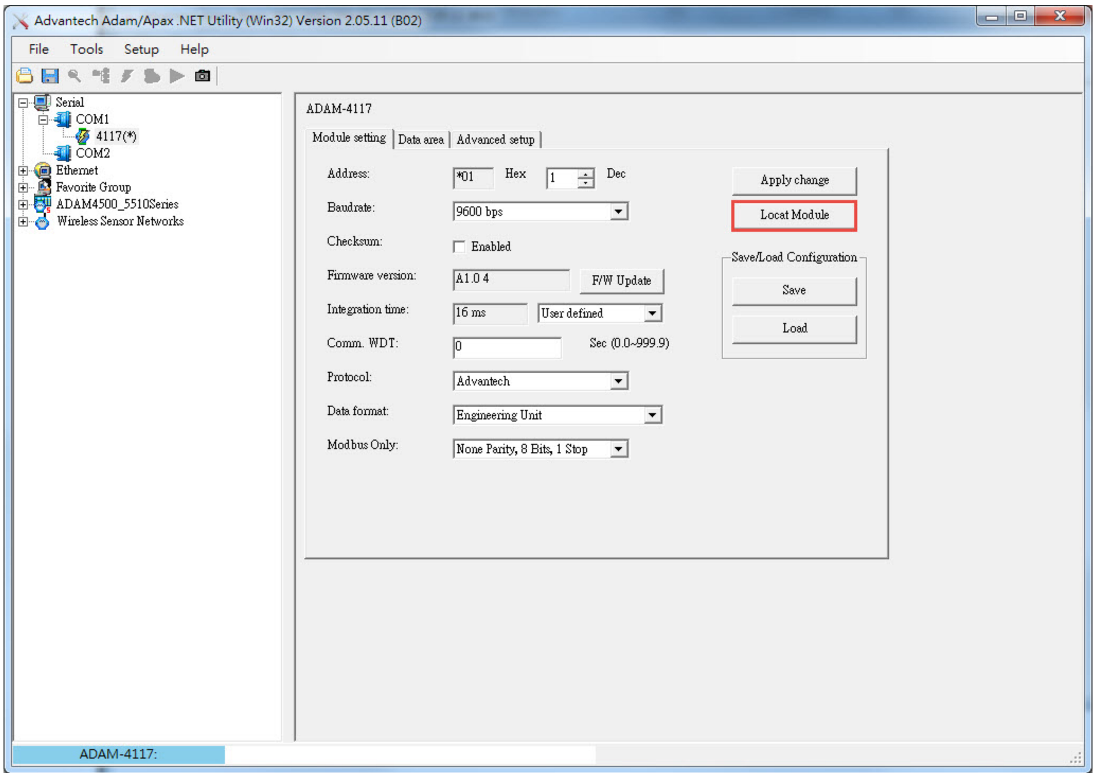
This is for ignoring sudden noise. The following illustration shows this concept for the input signal of Channel 1.



- The condition of Channel 1 is input range = ±1 V and filter = 20%
- 20% of the full-scale range (FSR) means 0.4 V
- The time interval T1~T2 is equal to the sampling time
- Conclusion: the difference between V1 @ T1 and V2 @ T2 is 0.7 - 0.2 = 0.5 V > 0.4 V; thus, signal V2 will be ignored

A.7 Locate Mode

If you want to locate a specific ADAM-4100 module, the configuration utility provides a locate function to assist you. When you select a specific device, the Status LED of that module will flash for 8 min. If **Locate** is clicked, the Status LED will remain on; if you click the button again, the Status LED will return to its default status.


Advantech Adam/Apax .NET Utility (Win32) Version 2.05.11 (B02)

ADAM-4117

Address: *01 Hex 1 Dec
Baudrate: 9600 bps
Checksum: Enabled
Firmware version: A1.0 4
Integration time: 16 ms User defined
Comm. WDT: 0 Sec (0.0~999.9)
Protocol: Advantech
Data format: Engineering Unit
Modbus Only: None Parity, 8 Bits, 1 Stop

Appendix B

Modbus Mapping Table

ADAM-4100 I/O Modbus Mapping Table

The following ADAM-4100 I/O series modules support the Modbus protocol:

- ADAM-4117 8-ch analog input module
- ADAM-4118 8-ch thermocouple input module
- ADAM-4150 Digital I/O module
- ADAM-4168 Relay output module
- ADAM-4115 thermocouple input module

Table B.1: ADAM-4117 Modbus Mapping Table

ADDR 4X	Channel	Item	Attribute	Memo
00201	0	Burnout	R	
00202	1	Burnout	R	
00203	2	Burnout	R	
00204	3	Burnout	R	
00205	4	Burnout	R	
00206	5	Burnout	R	
00207	6	Burnout	R	
00208	7	Burnout	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

Table B.2: ADAM-4118 Modbus Mapping Table

ADDR 4X	Channel	Item	Attribute	Memo
00201	0	Burnout	R	
00202	1	Burnout	R	
00203	2	Burnout	R	
00204	3	Burnout	R	
00205	4	Burnout	R	
00206	5	Burnout	R	
00207	6	Burnout	R	
00208	7	Burnout	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

Table B.3: ADAM-4150 Modbus Mapping Table 1 (0X)

ADDR 0X	Channel	Item	Attribute	Memo
00001	0	Digital input signal	R	
00002	1	Digital input signal	R	
00003	2	Digital input signal	R	
00004	3	Digital input signal	R	
00005	4	Digital input signal	R	
00006	5	Digital input signal	R	
00007	6	Digital input signal	R	
00017	0	Digital output signal	W	
00018	1	Digital output signal	W	
00019	2	Digital output signal	W	
00020	3	Digital output signal	W	
00021	4	Digital output signal	W	
00022	5	Digital output signal	W	
00023	6	Digital output signal	W	
00024	7	Digital output 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	

Table B.3: ADAM-4150 Modbus Mapping Table 1 (0X)

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

Table B.4: ADAM-4150 Modbus Mapping Table 2 (4X)

ADDR 4X	Channel	Item	Attribute	Memo
40001~40014	0~6	For counter/frequency Frequency= ADDR/10 Hz 7-ch, 32-bit	R	
40015~40030	0~7	For pulse output L level Increment: 0.1 ms 8-ch, 32-bit	R/W	
40031~40046	0~7	For pulse output H level Increment: 0.1 ms 8-ch, 32-bit	R/W	
40047~40062	0~7	Set absolute pulse (0 = continue mode) 8-ch, 32-bit	R/W	
40063~40078	0~7	Set incremental pulse 8-ch, 32-bit	R/W	
40079~40085	0~6	Reference	R/W	
40086~40093	0~7	Digital output mode	R/W	
40094~40107	0~6	Digital input filter low width	R/W	
40108~40121	0~6	Digital input filter high width	R/W	
40122~40137	0~7	Digital output low delay width	R/W	
40138~40155	0~7	Digital output high delay width	R/W	
40211		Module Name 1	R	0x41 0x50
40212		Module Name 2	R	0x00 0x00
40213		Version 1	R	0xa2 0x00
40214		Version 2	R	0xB0 0x01
40215		Comm safety enable	R	Enable: 0x00 0x01
40216		Comm safety flag	R	Occur: 0x00 0x01
40301		Digital input data in word	R	
40302		Reserved		
40303		Digital output data in word	R/W	

Reference:

- II&0x07 = 00 Digital input mode
- II&0x07 = 01 Counter mode
- II&0x07 = 02 Low-to-high latch mode
- II&0x07 = 03 High-to-low latch mode
- II&0x07 = 04 Frequency mode
- II&0x20 = 20 Digital input enable counter record function
- II&0x40 = 40 Digital input enable digital filter function
- II&0x80 = 80 Digital input invert mode

Table B.5: ADAM-4168 Module Mapping Table 1 (0X)

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	

Table B.6: ADAM-4168 Module Mapping Table 2 (0X)

ADDR 0X	Channel	Item	Attribute	Memo
40001~40016	0~7	For pulse output L level Increment: 0.1 ms 8-ch, 32-bit	R/W	0xa2 0x00
40017~40032	0~7	For pulse output H level Increment: 0.1 ms 8-ch, 32-bit	R/W	0xB00x01
40033~40048	0~7	Set absolute pulse (0 = continue mode) 8-ch, 32-bit	R/W	Enable: 0x00 0x01
40049~40064	0~7	Set incremental pulse 8-ch, 32-bit	R/W	Occur: 0x00 0x01
40065~40072	0~7	Digital output mode	R/W	
40073~40088	0~7	Digital output low delay width	R/W	
40089~40104	0~7	Digital output 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		Digital output data in word	R/W	

Table B.7: ADAM-4115-B Modbus Mapping Table

ADDR0X	Channel	Item	Attribute	Memo
00201	0	Burnout	R	
00202	1	Burnout	R	
00203	2	Burnout	R	
00204	3	Burnout	R	
00205	4	Burnout	R	
00206	5	Burnout	R	
ADDR4X	**Channel**	**Item**	**Attribute**	**Memo**
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	
40201	0	Type code	R/W	
40202	1	Type code	R/W	
40203	2	Type code	R/W	

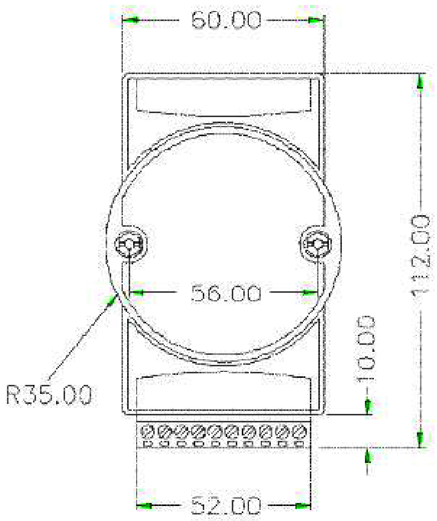
Table B.7: ADAM-4115-B Modbus Mapping Table

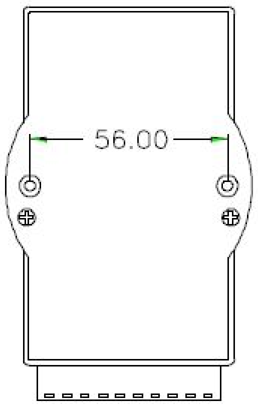
40204	3	Type code	R/W
40205	4	Type code	R/W
40206	5	Type code	R/W
40211		Module Name1	R
40212		Module Name2	R
40213		Version1	R
40214		Version2	R
40221		Channel Enable	R/W

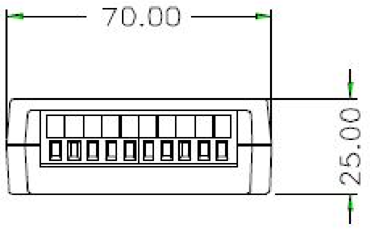
Appendix C

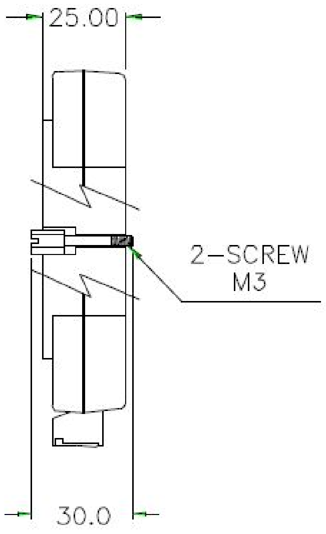
Technical Diagrams

C.1 ADAM Dimensions









C.2 Installation - DIN Rail

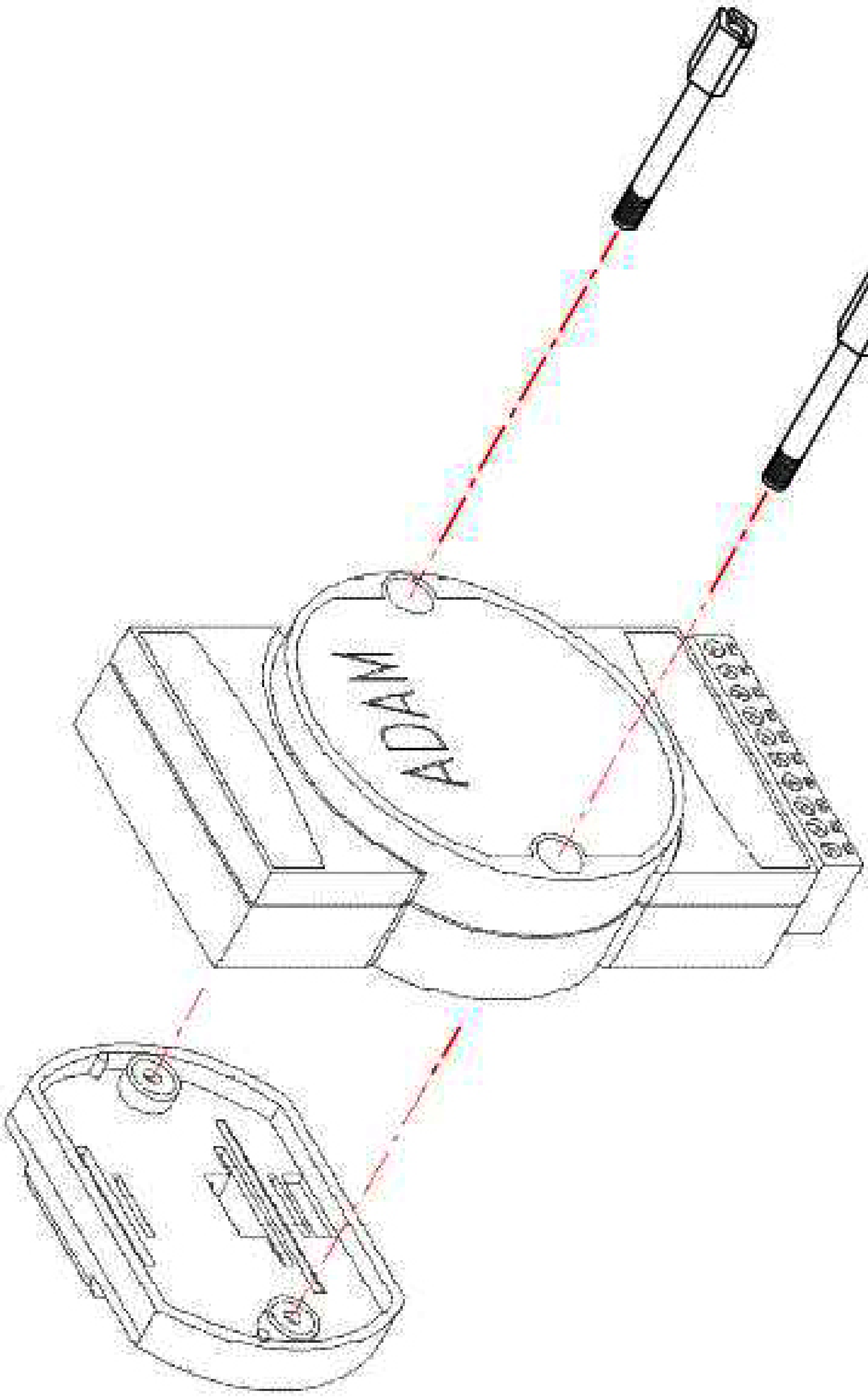


Figure C.1 Installation onto the DIN Rail Adapter

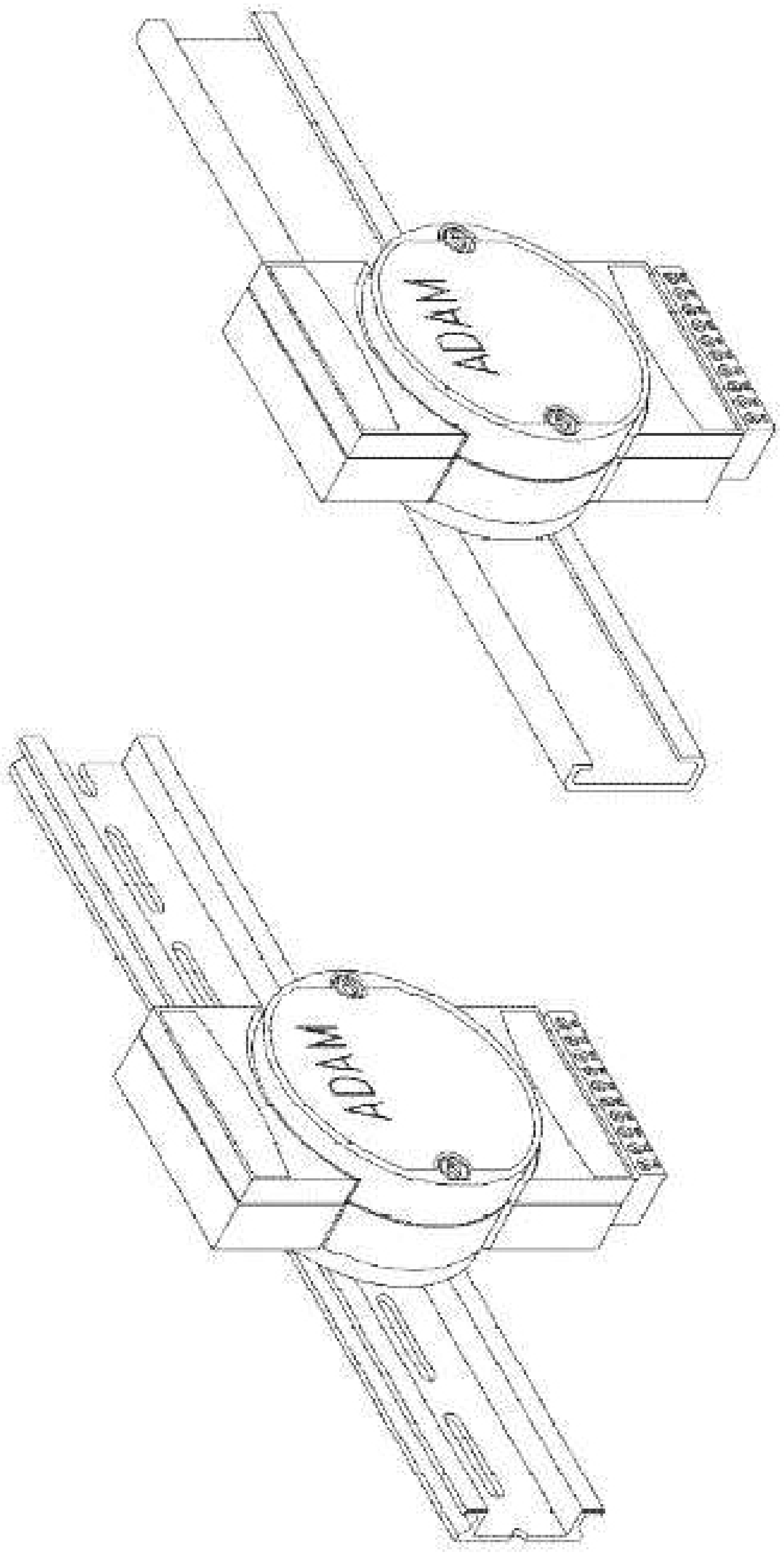


Figure C.2 Installation onto the DIN Rail

C.3 Installation - Panel Mounting

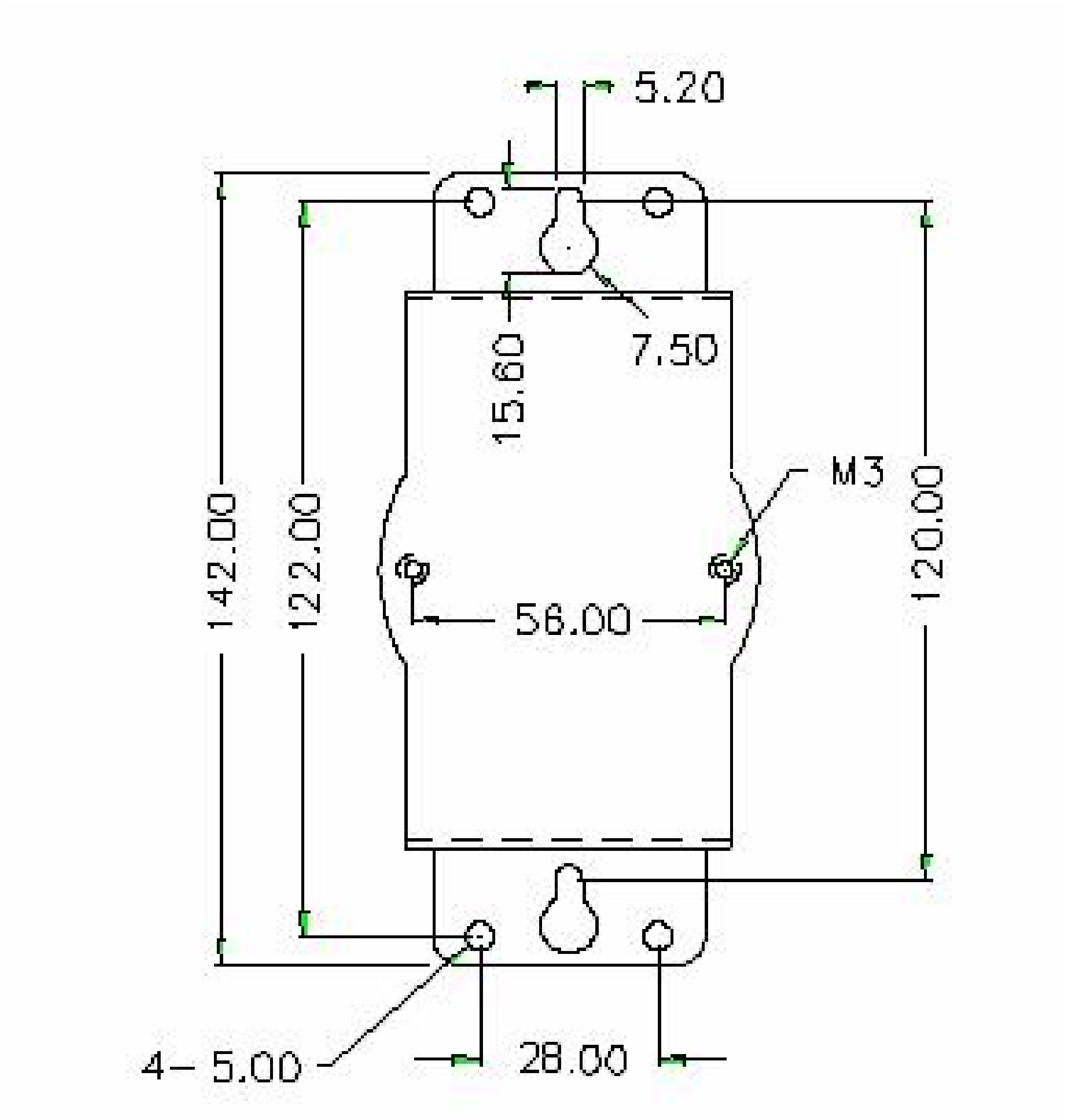


Figure C.3 Panel Mounting Bracket Dimensions

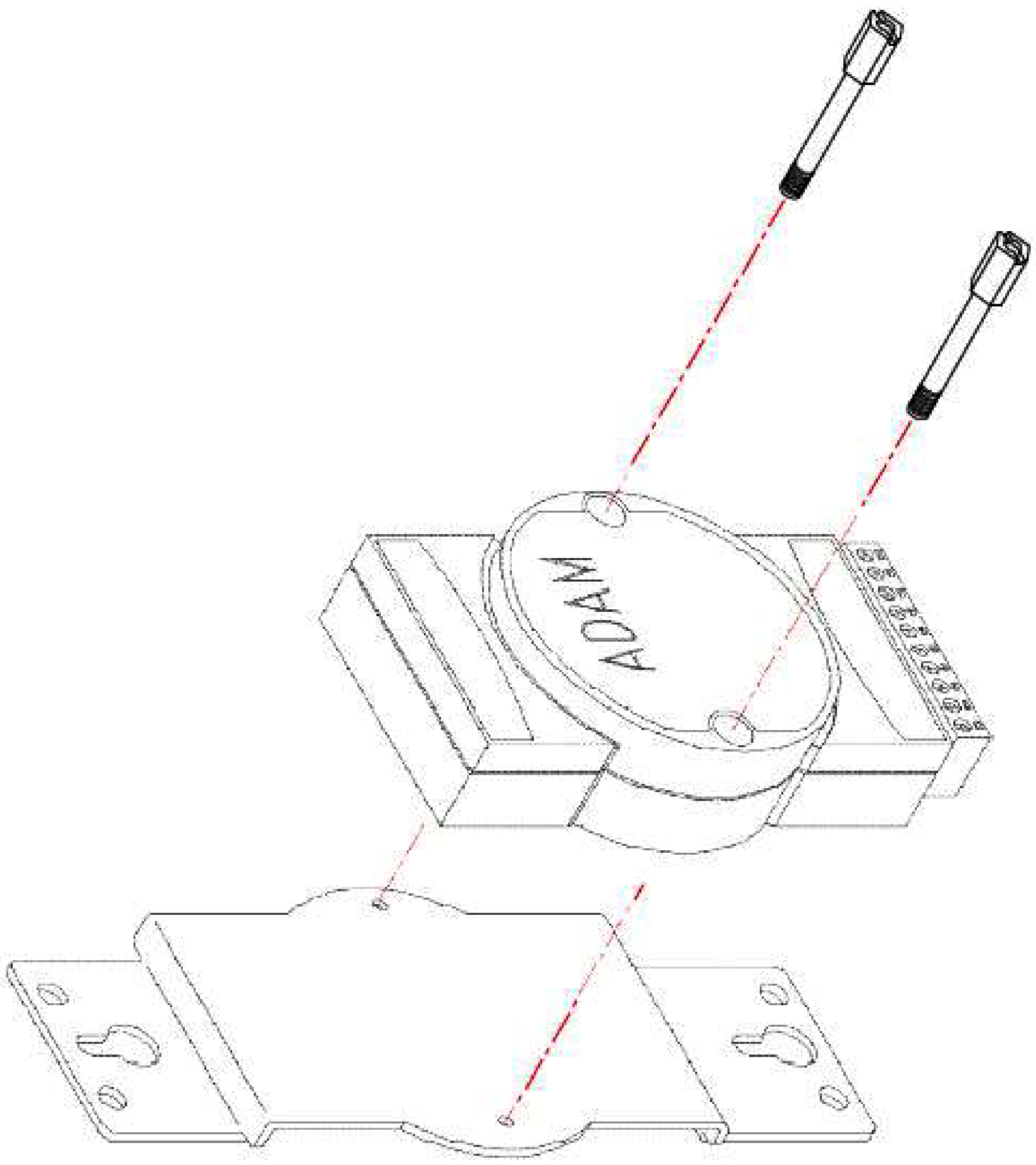


Figure C.4 Installation onto the Panel Mount

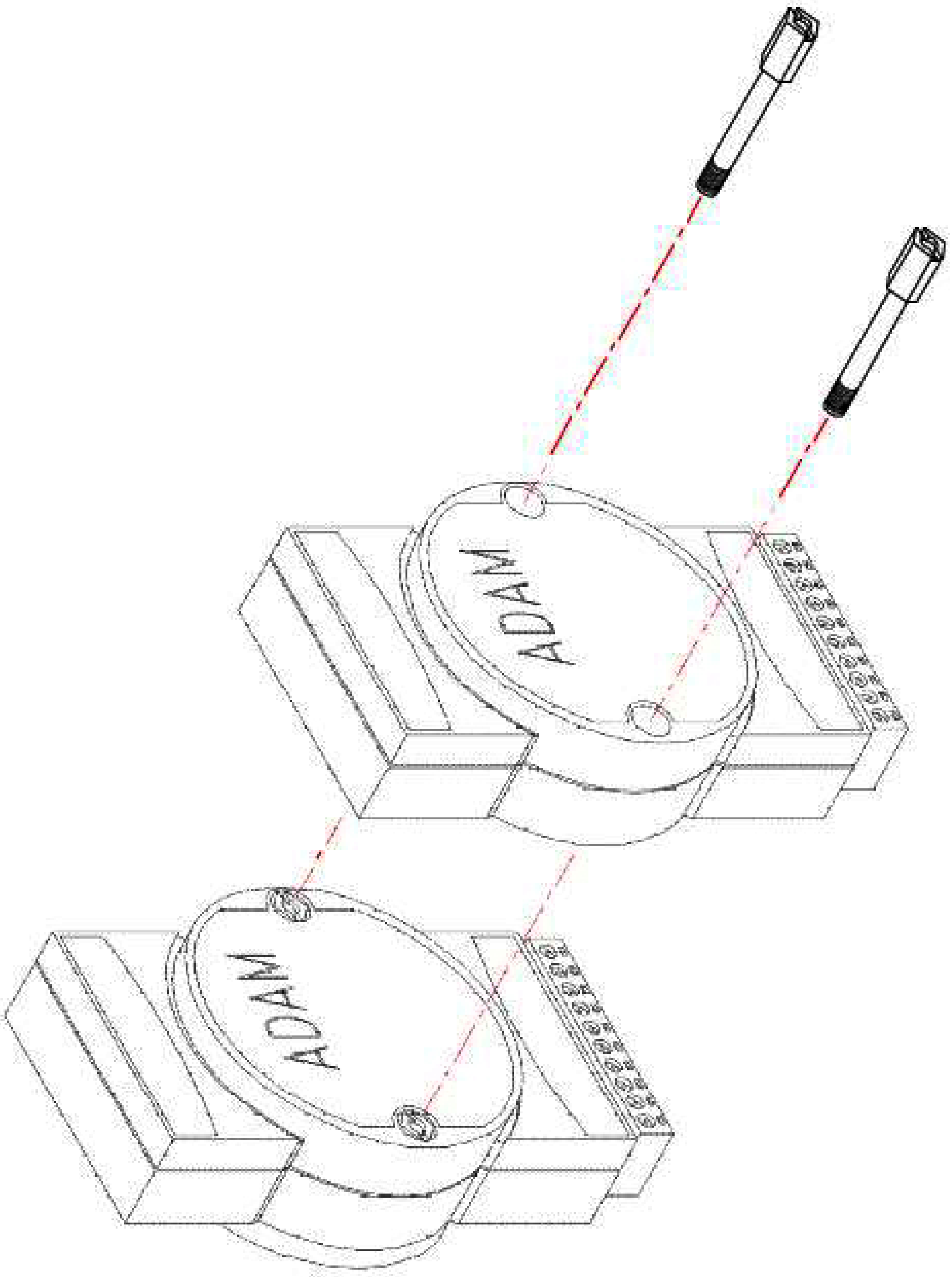


Figure C.5 Piggyback Stack Configuration

Appendix D

Data Formats and I/O Ranges

D.1 Analog Input Formats

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 a value of 00. This format presents data in standard units such as degrees, volts, millivolts, and milliamps. When the value in engineering format is converted to computer language, it is presented as seven characters that may include sign and decimals. However, the number of characters cannot exceed seven.

Data are grouped into a plus (+) or minus (-) sign, followed by five decimal digits and a decimal point. The input range that 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 0~760°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 range of ±5 V. When the engineering unit format is used, ADAM series analog input modules 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 FSR.

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 +040.00 (cr).

The full calibrated voltage range ranges from -100% to 100% because voltage input ranges are always bipolar. Thus, an input of ±5 V would range from -5 V (-100%) to 5 V (100%). In this example the input is represented by +40% of the FSR, which is equal to +(40/100) x 5 V = +2.0 V, which is the actual input value.

Example 2

The input value is 652.5°C, and a type E thermocouple (0~1,000°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 input value (652.5°C) is 65.25% of the FSR (1,000°C).

Thermocouple input ranges are always assumed to be bipolar with zero as the point of symmetry. This holds true regardless of the specified range of operation. For example, when a type J thermocouple (0~760°C) is used, 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 the point of symmetry. For instance, a type B thermocouple is specified for operation at 500~1800°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 to 100°C. Then, the lower value of -80°C is equal to 0% of the span and the upper value of 100°C is equal to 100% of the span.

In FSR mode, an over-range feature is automatically invoked by ADAM analog input modules if the value exceeds the uppermost value of the input range. For instance, an analog module that is configured for a range of ±5 V has one of the values reading + 5.5 V. 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 that 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 hex form, providing a rapid communication, high-resolution, and easy conversion to computer-compatible integer format.

To indicate twos complement hexadecimal, Bits 0 and 1 of the data format/checksum/integration time parameter must be set to 10. This format displays data in the form of a 4-character hexadecimal string.

The 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 range of ±5 V. The value returned is E069 (cr), which is equivalent to the signed integer -8087. Input ranges with voltage and milliamp values are used with the full calibrated voltage range of 8000~7FFF. For instance, an ADAM-4118 module is given an input range of ±2.5 V. In this case, -2.5 V is denoted as 8000h and +2.5 V as 7FFFh. When thermocouple input ranges are used, an input range that is bipolar and symmetric at zero is assumed. The following table provides several examples.

Thermocouple Type	Temperature Range	Temperature Range (Hex)
J	0°C to 760°C	0000~7FFF
T	-100°C to 400°C	E000~7FFF
R	500°C to 1750°C	2492~7FFF

Appendix E

RS-485

EIA RS-485 is the most widely used bidirectional, balanced transmission line standard. It is specifically developed for industrial multi-drop systems that should be able to transmit and receive data at high rates or over long distances.

The specifications of the EIA RS-485 protocol are as follows:

- Maximum line length per segment: 1200 m (4000 ft)
- Throughput of 10 Mbaud and beyond
- Differential transmission (balanced lines) with high resistance against noise
- Maximum 32 nodes per segment
- Bidirectional master-slave communication over a single set of twisted pair cables
- Parallel connected nodes, multi-drop capability

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

When nodes communicate through the network, no conflicts will occur during transmission since only a simple command/response sequence is used. There is always one initiator (with no address) and many slaves (with an address). In this case, the master is a PC connected through its serial RS-232 port to an ADAM RS-232/485 converter. The slaves are the ADAM I/O modules.

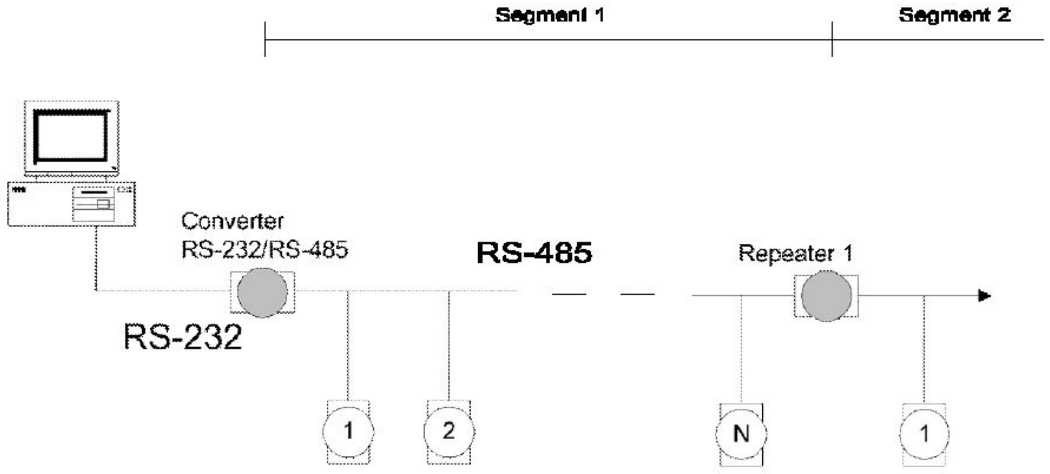
When the modules are not transmitting data, they are in listening mode. The host PC initiates a command/response sequence with one of the modules. Commands normally contain the address of the module that the host wants to communicate with. The module will respond back to the host once a match occurs 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 a node does not 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 ASCII-based computers and terminals. The basic layouts that can be used for an RS-485 network are described in the following text.

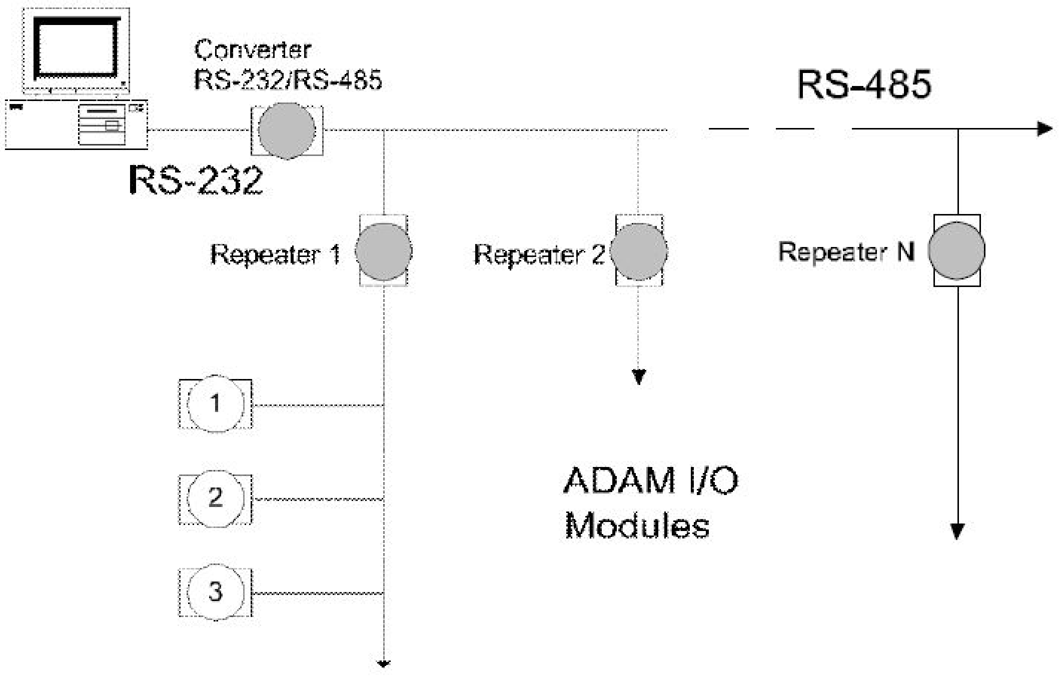
E.1.1 Daisy Chain Topology

The last module of a segment is a repeater and is directly connected to the main wires. Therefore, it acts as a medium that repeats the signals between two segments. However, there is a limitation in 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-digit hexadecimal representation. The maximum representation of two-digit hexadecimal representation is 256. The ADAM converter, repeaters, and host PC are non-addressable units; therefore, they are not included in these numbers.



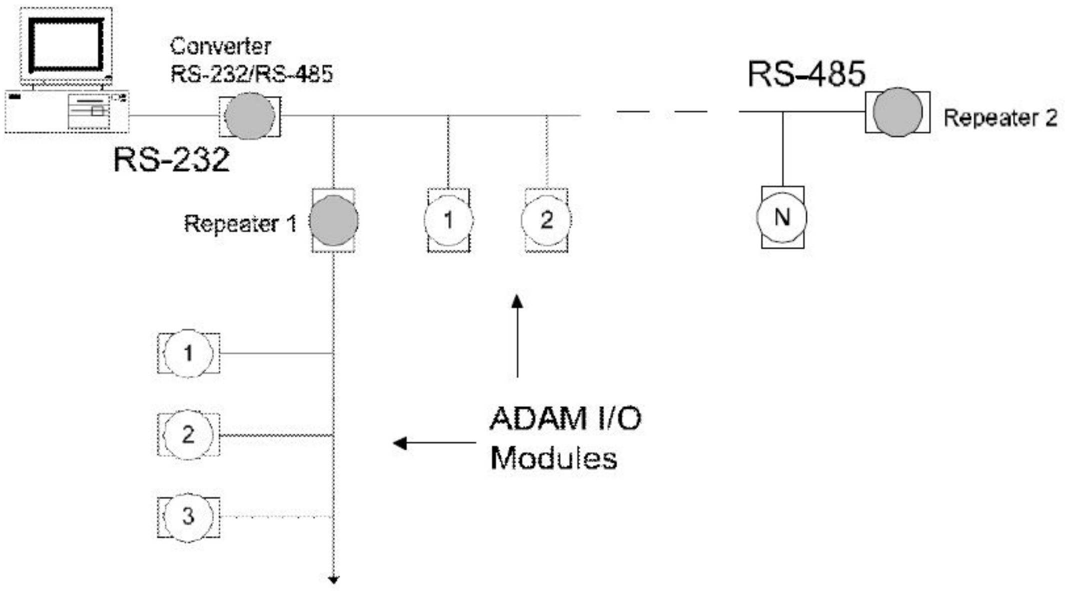
E.1.2 Star Topology

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



E.1.3 Random Topology

This is a combination of daisy chain and hierarchical topologies.



E.1.4 Line Termination

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

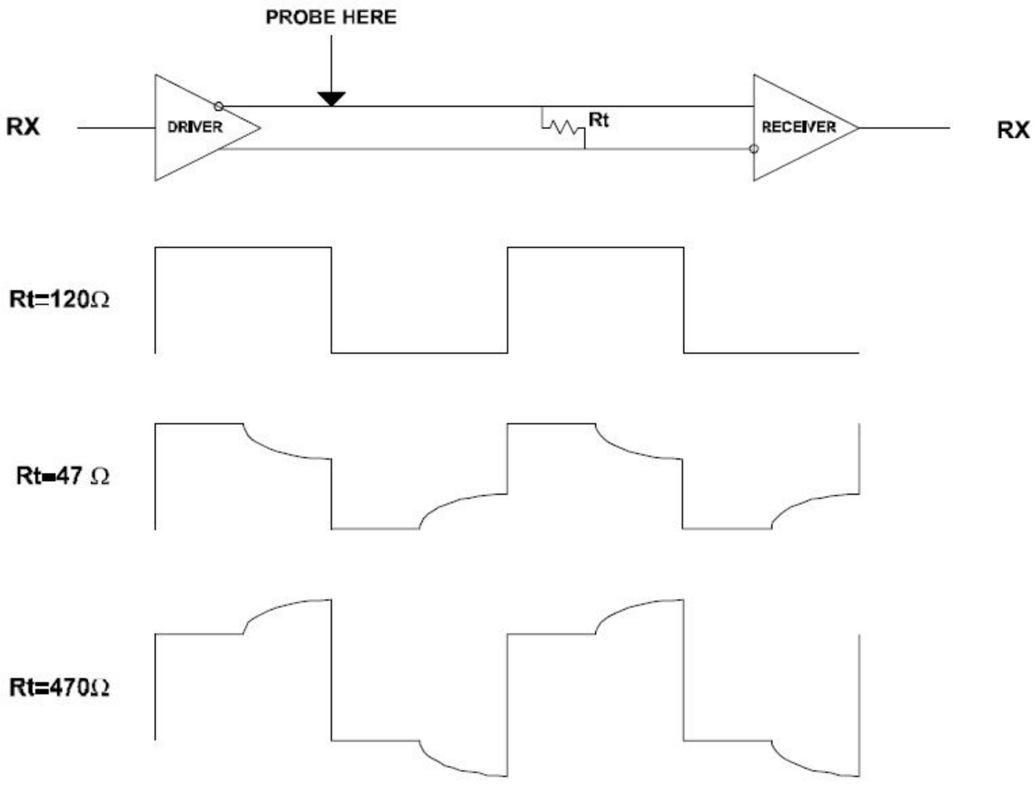


Figure E.1 Signal Distortion

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 equal to the characteristic impedance of the line should be sufficient.

Example

Each receiver input 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 that provides the large common range of the receiver required for RS-485 systems (see Figure E-5).

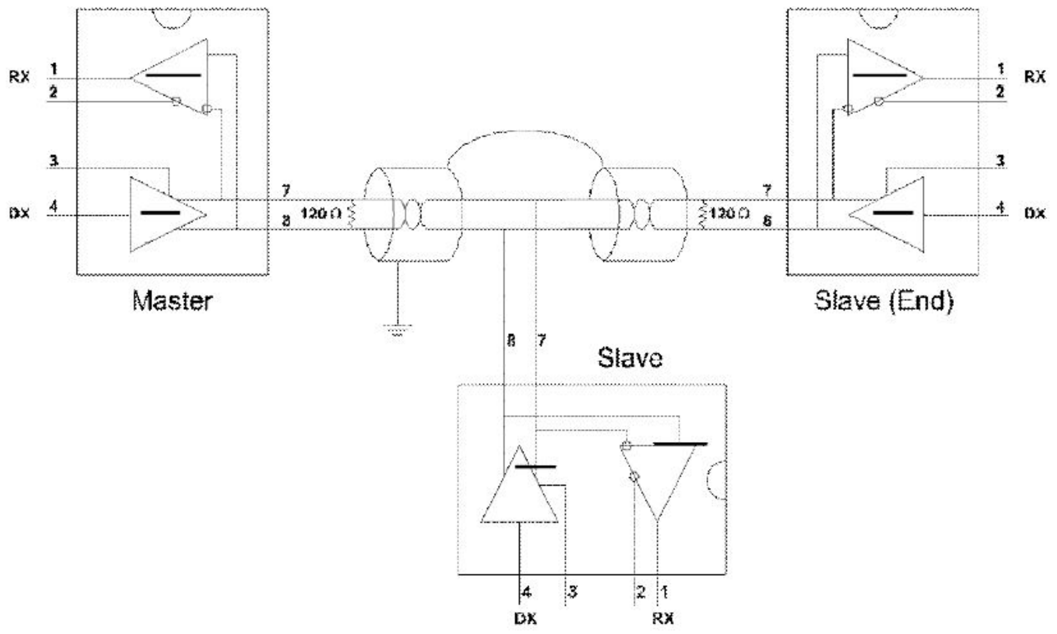


Figure E.2 Termination Resistor Locations

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

The effective parallel receiver resistance RP will therefore be equal to

RP = 36 x 10 / 30 = 1200 W

While the termination receiver RT will equal

RT = RO / [1 - RO / RP]

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

RT = [1 - 100 / 1200] = 110 Ω

since this value lies within 10% of the line characteristic impedance. Thus, as already stated, the line termination resistor RT will normally equal the characteristic impedance ZO.

The star connection causes a multitude of these discontinuities since there are several transmission lines; it is therefore not recommended.

Note! The recommended wiring method that causes the minimum amount of reflection is daisy chaining, where all receivers tap from one transmission line and need to be terminated only twice.

E.2 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 (i.e., request-to-send) and CTS (i.e., clear-to-send) are the most commonly used methods.

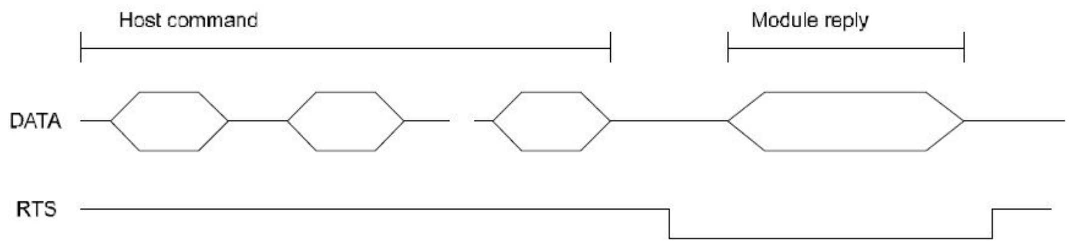


Figure E.3 RS-485 Data Flow Control with RTS

E.3 Intelligent RS-485 Control

The ADAM-4510 and ADAM-4520 are both equipped with an I/O circuit that can automatically sense the direction of the data flow. No handshaking with the host (like with RTS, for example) is necessary. Any software that 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.

Appendix F

Using the Checksum Feature

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

F.1 Checksum Enable/Disable

To enable configuration of a module's checksum feature, its INIT* terminal should be shorted to its ground terminal. Then, the module should be rebooted. The checksum feature is enabled by setting Bit 6 of the data format/checksum parameter to 1; conversely, the checksum is disabled by setting the parameter to 0. Whenever the checksum feature is used, all connected devices-including the host computer-should be in enable mode.

The checksum is represented in 2-character ASCII hexadecimal format and is transmitted just prior to the carriage return. The checksum is equal to the result after performing modulus -256 (100h) of the sum of all the ASCII values 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 characters "#", "0", and "5". The response checksum (9Dh) is the sum of the ASCII values for the 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 follows:

25h = (24h+ 30h + 37h + 52h + 48h) MOD 100h

The hexadecimal ASCII codes for "$", "0", "7", "R", and "H" are 24h, 30h, 37h, 52h, and 48h, respectively. The sum of these ASCII codes is 125h, and the result is equal 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	}		

Appendix G

Switching to Modbus

ADAM-4100 Modbus modules may come from the factory with the ADAM ASCII protocol set as the default protocol. In such cases, if the module is connected to a Modbus network, the network might not recognize the module. This may be due to incorrect settings. ADAM-4100 modules should be set up for the Modbus protocol instead of the ADAM ASCII protocol.

Follow these steps to configure an ADAM-4100 module to the Modbus protocol:

1. Download and install Adam/Apax .NET Utility (the latest version can be found at www.advantech.com).
2. Initialize the ADAM-4100 on an RS-485 network (the preferred method is to initialize one module at a time).
3. With the module powered off, move the switch to the "Init" position.
4. Power up the module.
5. Wait 10 s for the module to initialize.
6. Use Adam/Apax .NET Utility to search for the module to change the protocol (Initial COM settings: 9600 baud, N-8-1).
7. Once the utility has identified the module, the serial data protocol can be changed to the Modbus protocol.
8. If necessary, change the address and COM port settings.
9. To access the module, click on the module icon in the utility.
10. Update the settings by pressing **Update.**
11. Power off the module.
12. Move the switch back to the NORMAL* position.
13. The module is now ready to be placed in the Modbus network.

Enabling an Intelligent Planet