ADAM-5510M

Data Acquisition Modules
User's Manual

Acknowledgments

IBM and PC are trademarks of International Business Machines Corporation.

Part No. 2001500001 2'nd Edition

Printed in Taiwan Oct. 2001

Product Warranty

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

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

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

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

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

Contents

Figures

Tables

Chapter 1
System Overview

1.1 Introduction

Standalone Data Acquisition and Control System

As the growth of PC-based technology, Advantech PC-based controllers have been widely applied in variety of industrial automation applications. Enhanced from the ADAM-5510, the ADAM-5510M is a new standalone controller features high memory capacity, multi communication interfaces, user-friendly configuration tool and so on. Apply the ADAM-5510M, the C programmers would be able to handle any complex task easily.

1.2 Features

The ADAM-5510M system consists of two major components: the main unit and I/O modules. The main unit includes a CPU card, a power regulator, a 4-slot base, three serial communication ports and a programming port. It has the following major features:

1.2.1 Control flexibility with C programming

The ADAM-5510M is a compact PC in its own right and includes an 80188 CPU and a built-in ROM-DOS operating system. It can be used in a way similar to how one uses an x86 PC in the office. Programmers in C/C++ can write and compile applications in Inprise (Borland) Turbo C and download to the ADAM-5510M. The communication protocol of the ADAM-5510 is an ASCII-based command and response protocol. Given the prevalence of C language programming tools, this is a distinct advantage for many users and can result in a very short learning curve and very modest training expense requirements.

1.2.2 RS-232/485 communication ability

The ADAM-5510M has four serial communication ports, giving it excellent communication abilities. This facilitates its ability to control networked devices. Refer to Figure 1-1, COM1 is a dedicated RS-232 port, COM2 is a dedicated RS-485 port, and COM4 is a RS-232/485 selectable port. These three ports allowed the ADAM-5510M to satisfy diverse communication and integration demands. COM3 is a spare programming port for downloading or transferring executable programs from a host PC. It can also be used as an RS-232 communication port.

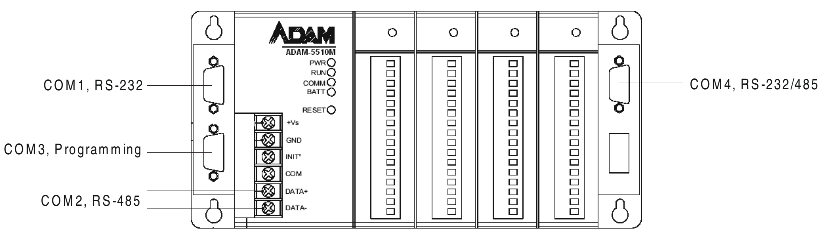


Figure 1-1 ADAM-5510M Communication Ports

1.2.3 Complete set of I/O modules for total solutions

The ADAM-5510M uses a convenient backplane system common to the ADAM-5000 series. Advantech's complete line of ADAM-5000 modules integrates with the ADAM-5510M to support your applications. Following table is the I/O modules support list we provided for user's choice.

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

Table 1-1 I/O Module Support List

A full range of digital modules support 10 to 30 V_{DC} I/O and relay outputs. A set of analog modules provide 16-bit resolution and programmable input and output (including bipolar) signal ranges. For details, refer to Chapter 4: I/O Module Introduction.

A complete set of C language I/O subroutines are included in the ADAM-5511M's function library to reduce programming effort. Users can easily call these subroutines to execute the ADAM-5510M's I/O functions while programming in Borland C 3.0 languages. For a detailed description, refer to Chapter 5: Function Library.

1.2.4 Built-in ROM and RAM disk for programming

The ADAM-5510M has built-in Flash Memory and SRAM for file downloading, system operation and data storage. It provides 512 KB file system, 400 KB free for users to download programs. There are also 640 KB SRAM to provide the memory needed for efficient application operation and file transfer. Moreover, users are allowed to decide the battery backup memory size up to 512 KB in the SRAM.

1.2.5 Built-in real-time clock and watchdog timer

The micro-controller also includes a real-time clock and watchdog timer. The real-time clock records events while they occur. The watchdog timer is designed to automatically reset the microprocessor if the system fails. This feature greatly reduces the level of maintenance required and makes the ADAM-5510M ideal for use in applications which required a high level of system stability.

1.3 ADAM-5510M Specification

1.3.1 System

- **CPU:** 80188-40, 16-bit microprocessor
- **Operating system:** Boot ROM-DOS
- **Flash Memory:** 1 MB
  256 KB System Flash
  256 KB Flash Memory
  512 KB File System (400 KB for users' application)
- **SRAM:** 640 KB (Up to 512 KB battery backup)
- **Timer BIOS:** Yes
- **Real-time clock:** Yes
- **Watchdog timer:** Yes
- **COM1:** RS-232, DB-9 connector
- **COM2:** RS-485, DB-9 connector
- **COM3:** Programming port (RS-232 interface, DB-9 connector): Tx, Rx, GND
- **COM4:** RS-232/485 selectable, DB-9 connector
- **I/O capacity:** 4 modules (One ADAM-5024 allowed)
- **CPU power consumption:** 1.0 W
- **Status display:** Power, CPU, Communication, Battery

1.3.2 RS-232 interface (COM1)

- **Signals:** TxD, RxD, RTS, CTS, DTR, DSR, DCD, RI, GND
- **Mode:** Asynchronous full duplex, point to point
- **Connector:** DB-9 pin
- **Transmission speed:** Up to 115.2 Kbps
- **Max transmission distance:** 50 feet (15.2 m)

1.3.3 RS-485 interface (COM2)

- **Signals:** DATA+, DATA-
- **Mode:** Half duplex, multi-drop
- **Connector:** Screw terminal
- **Transmission speed:** Up to 115.2 Kbps
- **Max transmission distance:** 4000 feet (1220 m)

1.3.4 RS-232 programming port (COM3)

- **Signals:** Tx, Rx, GND
- **Mode:** Asynchronous, point to point
- **Connector:** DB-9 pin
- **Transmission speed:** Up to 115.2 Kbps
- **Max transmission distance:** 50 feet (15.2 m)

1.3.5 RS-232/485 interface (COM4)

- RS-232/485 Mode Selectable (Select by jumper setting, refer to Figure 1-2)

  RS-485 Signal: DATA+, DATA-
- **RS-232 Mode:** Asynchronous full duplex, point to point

  Signals: TxD, RxD, RTS, CTS, DTR, DSR, DCD, RI, GND
- **RS-485 Mode:** Half duplex, multi-drop

  RS-485 Signal: DATA+, DATA-
- **Connector:** DB-9 pin
- **Transmission speed:** Up to 115.2 Kbps
- **Max transmission distance:**
  RS-232: 50 feet (15.2 m)
  RS-485: 4000 feet (1220 m)

1.3.6 Isolation

- **Power:** 3000 V_{DC}
- **Input/Output:** 3000 V_{DC}
- **Communication:** 2500 V_{DC} (COM2 only)

1.3.7 Power

• Unregulated +10 to +30 VDC

• Protected against power reversal

• Power consumption: 2.0 W

1.3.8 Mechanical

• Case: ABS with captive mounting hardware

• Plug-in screw terminal block:

Accepts 0.5 mm2 to 2.5 mm2, 1 - #12 or 2 - #14 to #22 AWG

1.3.9 Environment

• Operating temperature: -10° to 70° C (14° to 158° F)

• Storage temperature: -25° to 85° C (-13° to 185° F)

• Humidity: 5 to 95 %, non-condensing

• Atmosphere: No corrosive gases

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

1.3.10 Dimensions

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

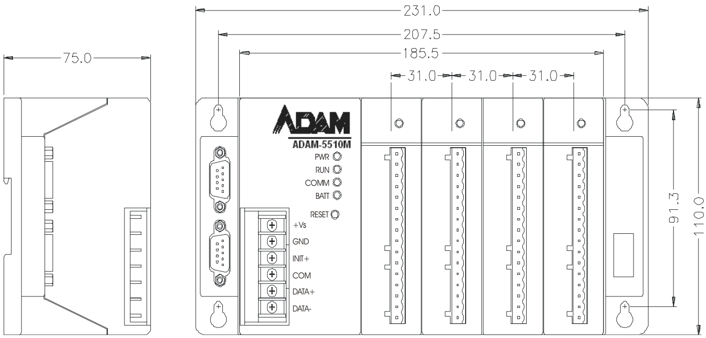


Figure 1-2: ADAM-5510M System IO and Module Dimension

1.3.11 LED Status of the ADAM-5510M

There are four LEDs on the ADAM-5510M front panel. The LED's indicate ADAM-5510M's operating status, as explained below:

(1) PWR: power indicator. This LED is on whenever the ADAM-5510M is powered on.

(2) RUN: program execution indicator. This LED is regularly blinks whenever the ADAM-5510M is executing a program.

(3) COMM: communication indicator. This LED blinks whenever the host PC and the ADAM-5510M are communicating. Please notice: if the host COM port is connected to the ADAM-5510M's RS-232 port, this LED will normally be off. On the other hand, if the host COM port is connected to the ADAM-5510M's RS-485 port, this LED will normally be on.

(4) BATT: battery status indicator. This LED will be on whenever the SRAM backup battery is low.

Chapter 2
Installation Guidelines

This chapter explains how to install an ADAM-5510M stand-alone controller. A quick hookup scheme is provided that lets you easily configure your system before implementing it into your application.

2.1 System Requirements

Before you start installing the ADAM-5510M, make sure the system requirements are met:

2.1.1 Host Computer Requirements

1. IBM PC compatible computer with 486 CPU (Pentium is recommended).
2. Microsoft 95/98/NT 4.0 (SP3 or SP4) or higher versions.
3. DOS version 3.31 or higher.
3. Borland Turbo C 3.0 for DOS
4. At least 32 MB RAM.
5. 20 MB of hard disk space available
6. VGA color monitor.
7. 2x or higher speed CD-ROM.
8. Mouse or other pointing devices.
9. At least one standard RS-232 port (e.g. COM1, COM2).
10. One RS-485 card or RS-232 to RS-485 converter (e. g. ADAM-4520) for system communication.

2.1.2 ADAM-5510M Requirements

1. One ADAM-5510M main unit.
2. One ADAM-5510M Quick Start Book
3. One core clamp for power supply connection.
4. One ADAM Products Utilities CD.
5. Power supply for ADAM-5510M (+10 to +30 VDC unregulated)
6. One RS-232 straight through DB-9 cable

2.1.3 I/O Module Requirements

At least one I/O module is needed to use the system. A variety of I/O modules are available to meet different application requirements. Table 2-1 gives a current listing of these modules for your reference.

Module	Name	Specification	Reference
Analog I/O	ADAM-5013	3-ch. RTD input	Isolated
	ADAM-5017	8-ch. AI	Isolated
	ADAM-5017H	8-ch. High speed AI	Isolated
	ADAM-5018	7-ch. Thermocouple input	Isolated
	ADAM-5024	4-ch. AO	Isolated
Digital I/O	ADAM-5050	7-ch. D I/O	Non-isolated
	ADAM-5051	16-ch. DI	Non-isolated
	ADAM-5051D	16-ch. DI W/ LED	Non-isolated
	ADAM-5052	8-ch. DI	Isolated
	ADAM-5056	16-ch. DO	Non-isolated
	ADAM-5056D	16-ch. DO W/LED	Non-isolated
Relay Output	ADAM-5060	6-ch. Relay output	Isolated
	ADAM-5068	8-ch. Relay output	Isolated
Counter/Frequency	ADAM-5080	4-ch. Counter/Frequency	Isolated
Serial I/O	ADAM-5090	4-port RS232	Non-isolated

Table 2-1 I/O Module Support List

2.2 Hardware Installation

2.2.1 Selecting I/O Module

To organize an ADAM-5510M data acquisition & control system, you need to select I/O modules to interface the main unit with field devices or processes that you have previously determined. There are several things should be considered when you select the I/O modules.

What type of I/O signal is applied in your system?

How many I/O is required to your system?

How will you place the controller for concentrate the I/O points of an entire process?

How many ADAM-5000 main units are required for distributed I/O points arrangement.

What is the required voltage range for each I/O module?

What isolation environment is required for each I/O module?

What are the noise and distance limitations for each I/O module?

Refer to table 2-2 as I/O module selection guidelines

Choose this type of I/O module:	For these types of field devices or operations (examples):	Explanation:
Discrete input module and block I/O module	Selector switches, pushbuttons, photoelectric eyes, limit switches, circuit breakers, proximity switches, level switches, motor starter contacts, relay contacts, thumbwheel switches	Input modules sense ON/OFF or OPENED/CLOSED signals. Discrete signals can be either ac or dc.
Discrete output module and block I/O module	Alarms, control relays, fans, lights, horns, valves, motor starters, solenoids	Output module signals interface with ON/OFF or OPENED/CLOSED devices. Discrete signals can be either AC or DC.
Analog input module	Thermocouple signals, RTD signals, temperature transducers, pressure transducers, load cell transducers, humidity transducers, flow transducers, potentiometers.	Convert continuous analog signals into input values for ADAM-5510M
Analog output module	Analog valves, actuators, chart recorders, electric motor drives, analog meters	Interpret ADAM-5510M output to analog signals (generally through transducers) for field devices.

Table 2-2 I/O Selection Guidelines

Advantech provides 19 types of ADAM-5000 I/O modules for various applications so far. The Figure 2-1 and table 2-3 will help you to select the ADAM-5000 I/O modules quickly and easily.

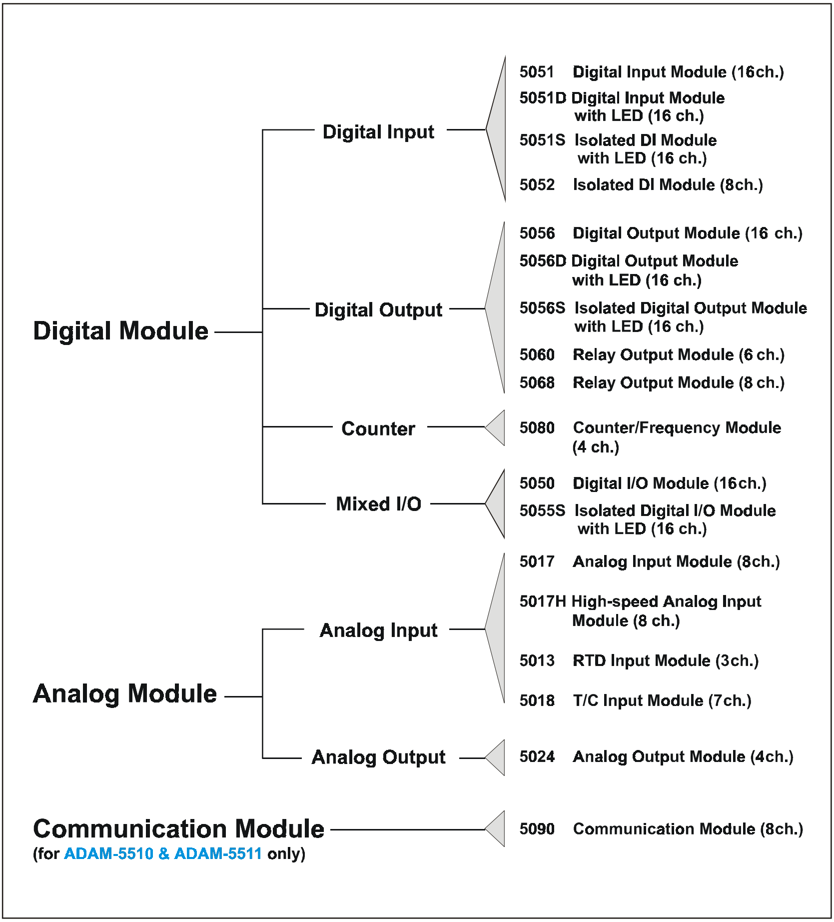


Figure 2-1 ADAM-5000 I/O Module Selection Chart

Module		ADAM-5013	ADAM-5017	ADAM-5017H	ADAM-5018	ADAM-5024	ADAM-5050	ADAM-5051	ADAM-5051D	ADAM-5051S
Analog Input	Resolution	16 bit	16 bit	12 bit	16 bit	-	-	-	-	-
	Input Channel	3	8	8	7	-	-	-	-	-
	Sampling Rate	10	10	8K	10	-	-	-	-	-
	Voltage Input	-	±150 mV ±500 mV ±1 V ±5 V ±10 V	±250 mV ±500 mV ±1 V ±5 V ±10 V	±15 mV ±50 mV ±100 mV ±500 mV ±1 V ±2.5 V	-	-	-	-	-
	Current Input	-	±20 mA*	±20 mA*	±20 mA*	-	-	-	-	-
	Direct Sensor Input	Pt or Ni RTD	-	-	J, K, T, E, R, S, B	-	-	-	-	-
Analog Output	Resolution	-	-	-	-	12 bit	-	-	-	-
	Voltage Output	-	-	-	-	0~10 V	-	-	-	-
	Current Output	-	-	-	-	0~20 mA 4~20 mA	-	-	-	-
Digital Input and Digital Output	Digital Input Channels	-	-	-	-	-	16 DIO (bit-wise selectabl-e)	16	16 W/LED	16 W/LED
	Digital Output Channels	-	-	-	-	-		-	-	-
Count-er (32-bit)	Channels	-	-	-	-	-	-	-	-	-
	Input Frequency	-	-	-	-	-	-	-	-	-
	Mode	-	-	-	-	-	-	-	-	-
COM-M	Channels	-	-	-	-	-	-	-	-	-
	Type	-	-	-	-	-	-	-	-	-
Isolation		3000 V_{DC}	3000 V_{DC}	3000 V_{DC}	3000 V_{DC}	3000 V_{DC}	-	-	-	2500 V_{DC}

Module		ADAM-5052	ADAM-5055S	ADAM-5056	ADAM-5056D	ADAM-5056S /5056SO	ADAM-5060	ADAM-5068	ADAM-5080	ADAM-5090
Analog Input	Resolution	-	-	-	-	-	-	-	-	-
	Input Channel	-	-	-	-	-	-	-	-	-
	Sampling Rate	-	-	-	-	-	-	-	-	-
	Voltage Input	-	-	-	-	-	-	-	-	-
	Current Input	-	-	-	-	-	-	-	-	-
	Direct Sensor Input	-	-	-	-	-	-	-	-	-
Analog Output	Resolution	-	-	-	-	-	-	-	-	-
	Voltage Output	-	-	-	-	-	-	-	-	-
	Current Output	-	-	-	-	-	-	-	-	-
Digital Input and Digital Output	Digital Input Channels	8	8 W/LED	-	-	-	-	-	-	-
	Digital Output Channels	-	8 W/LED	16	16 W/LED	16 W/LED	6 relay (2 form A/ 4 form C)	8 relay (8 form A)	-	-
Count-er (32-bit)	Channels	-	-	-	-	-	-	-	4	-
	Input Frequency	-	-	-	-	-	-	-	5000 Hz (max)	-
	Mode	-	-	-	-	-	-	-	Frequency, Up/Down Counter, Bi-direction Counter	-
COM-M	Channels	-	-	-	-	-	-	-	-	4
	Type	-	-	-	-	-	-	-	-	RS-232
Isolation		5000 V_{RMS}	2500 V_{DC}	-	-	2500 V_{DC}	-	-	1000 V_{RMS}	-

Table 2-3 I/O Selection Guidelines

2.2.2 Selecting Power Supply Module

ADAM-5510M controller works under unregulated power source between +10 and +30 VDC. When you arrange different I/O modules on ADAM-5510M's back plant, it may require comparable power supply. Use the following steps as guidelines for selecting a power supply for your ADAM-5510M control system.

Refer to table 2.4 to check the power consumption of ADAM-5510M Controller and each I/O module.

Main Units	Description	Power Consumption
ADAM-5000/485	Distributed Data Acquisition and Control System based on RS-485	1.0 W
ADAM-5000E	Distributed Data Acquisition and Control System based on RS-485	4.0 W
ADAM-5000/TCP	Distributed Data Acquisition and Control System based on Ethernet	5.0 W
ADAM-5510	PC-Based Programmable Controller (With Battery Backup)	1.0 W
ADAM-5510M	Enhanced PC-Based Programmable Controller (With Battery Backup)	1.2 W
ADAM-5511	PC-Based Programmable Controller with Modbus	1.0 W
I/O Modules	**Description**	**Power Consumption**
ADAM-5013	3-Channel RTD Input Module	1.1 W
ADAM-5017	8-Channel Analog Input Module (mV, mA or High Voltage)	1.25 W
ADAM-5017H	8-Channel High speed Analog Input Module (mV, mA or High Voltage)	2.2 W
ADAM-5018	7-Channel Thermocouple Input Module (mV, V, mA, Thermocopule)	0.63 W
ADAM-5024	4-Channel Analog Output Module (V, mA)	2.9 W
ADAM-5050	16-Channel Universal DIO	1.2 W
ADAM-5051	16-Channel Digital Input Module	0.53 W
ADAM-5051D	16-Channel Digital Input w/LED Module	0.84 W
ADAM-5056S	16-Channel Isolated Digital Input w/LED Module	0.8 W
ADAM-5056SO	16-Channel Digital Input w/LED Module	0.84 W
ADAM-5052	8-Channel Isolated DI	0.27W
ADAM-5055S	16-Channel Isolated DIO w/LED Module	0.68 W
ADAM-5056	16-Channel Digital Output Module	0.53 W
ADAM-5056D	16-Channel Digital Output w/LED Module	0.84 W
ADAM-5056S	16-Channel Isolated Digital Output w/LED Module	0.6 W
ADAM-5060	6-Channel Relay Output Module (2 of Form A, 4 of Form C)	1.8 W
ADAM-5068	8-Channel Relay Output Module (8 of Form A)	1.8 W
ADAM-5080	4-Channel Counter/ Frequency Input Module	1.5 W
ADAM-5090	4-Port RS232 Module	0.6 W

Table 2.4 Power Consumption of ADAM-5000 series

Calculate the Summary of the whole system's power consumption. For example, there are following items in your system.

ADAM-5510M * 3 & ADAM-5024 * 2 & ADAM-5017 * 4 & ADAM-5068 * 2 & ADAM-5080 * 2

⇓

The power consumption is:

1.2W * 3 + 2.9W * 2 + 1.25 * 4 + 1.8W * 2 + 1.5W * 2 = 21W

Select a suitable power supply from Table 2.5 or other comparable power resource for system operation.

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

Table 2.5 Power Supply Specification Table

2.2.3 Install Main Unit and Modules

When inserting modules into the system, align the PC board of the module with the grooves on the top and bottom of the system. Push the module straight into the system until it is firmly seated in the backplane connector. Once the module is inserted into the system, push in the retaining clips (located at the top and bottom of the module) to firmly secure the module to the system.

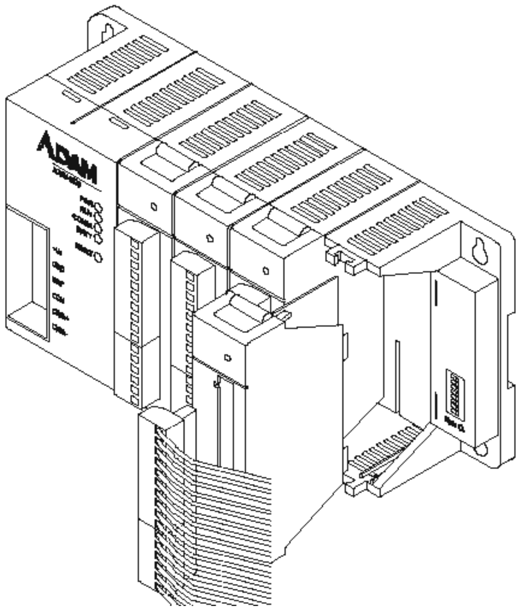


Figure 2-2 Module alignment and installation

2.2.4 I/O Slots and I/O Channel Numbering

The ADAM-5510M system provides 4 slots for use with I/O modules. The I/O slots are numbered 0 through 3, and the channel numbering of any I/O module in any slot starts from 0. For example, the ADAM-5017 is an 8-channel analog input module. Its input channel numbering is 0 through 7.

2.2.5 Mounting

The ADAM-5510M system can be installed on a panel or on a DIN rail.

Panel mounting

Mount the system on the panel horizontally to provide proper ventilation. You cannot mount the system vertically, upside down or on a flat horizontal surface. A standard #7 tatting screw (4 mm diameter) should be used.

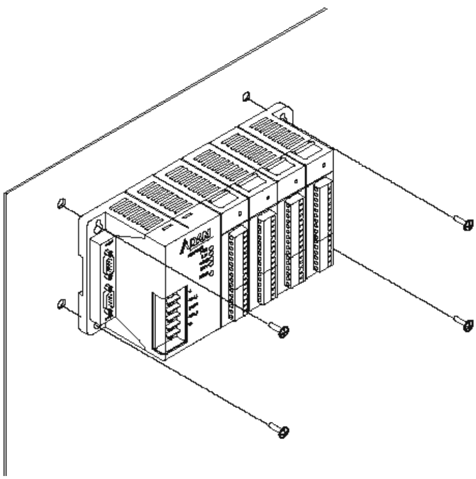


Figure 2-3: ADAM-5510M panel mounting screw placement

DIN rail mounting

The system can also be secured to the cabinet by using mounting rails. If you mount the system on a rail, you should also consider using end brackets at each end of the rail. The ended brackets help keep the system from sliding horizontally along the rail. This minimizes the possibility of accidentally pulling the wiring loose. If you examine the bottom of the system, you will notice two small retaining clips. To secure the system to a DIN rail, place the system on to the rail and gently push up on the retaining clips. The clips lock the system on the rail. To remove the system, pull down on the retaining clips, lift up on the base slightly, and pull it away from the rail.

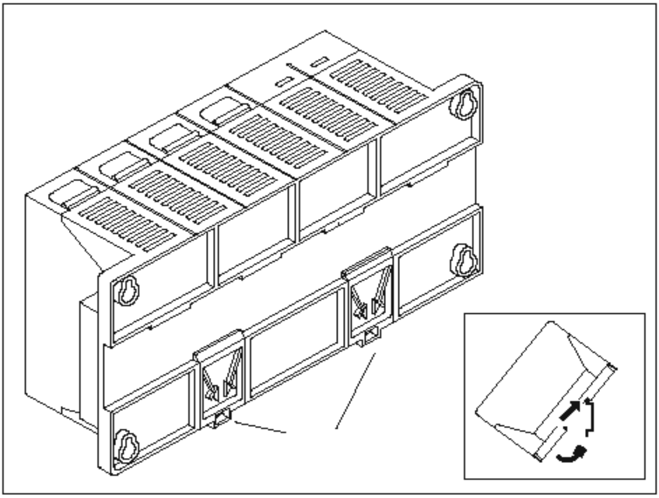


Figure 2-4: ADAM-5510M rail mounting

2.2.6 Jumper Settings and DIP Switch Settings

This section tells you how to set the jumpers and DIP switches to configure your ADAM-5510M system. It gives the system default configuration and your options for each jumper and dip switch. There are three jumpers (JP2~JP4) on the CPU card, and one 8-pin DIP switch on backplane board.

JP2 is for the watchdog timer setting

JP3 is for COM2 port RS-485 setting

JP4 is for battery power ON/OFF setting

The following figure shows the location of the jumpers:

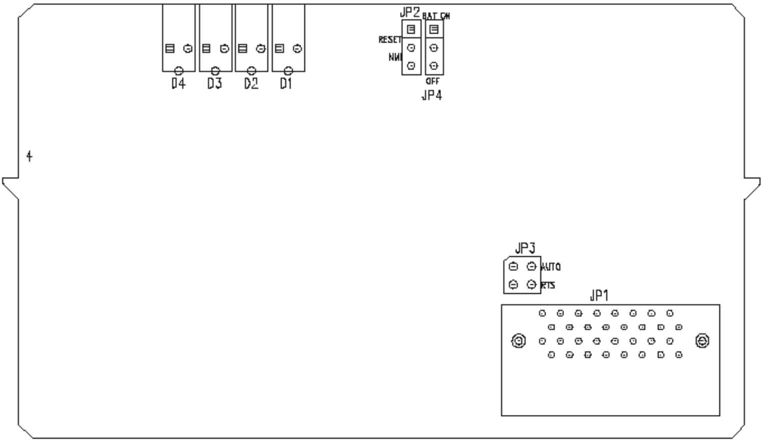


Figure 2-5: Jumper locations on the CPU card

COM2 port RS-485 control mode setting

The COM2 port is dedicated as an RS-485 interface. In an RS-485 network, handshaking signals such as RTS (Request to Send), normally control the direction of the data flow. A special I/O circuit in the ADAM-5510M senses the data flow direction and automatically switches the transmission direction, making handshaking signals unnecessary. Jumper JP3 gives users the option of configuring the COM2 port for automatic control or RTS control. Jumper settings are shown in Figure 2-5:

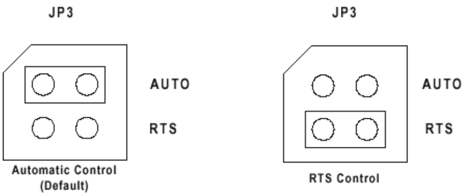


Figure 2-6: COM2 port RS-485 control mode setting (JP3)

Watchdog timer setting

Jumper JP2 on the CPU card lets you configure the watchdog timer to disable mode, reset mode or NMI (Non-maskable interrupt) mode.

Jumper settings are shown below:

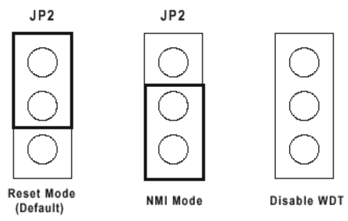


Figure 2-7: Watchdog timer setting

2.3 System Wiring and Connections

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

2.3.1 Power supply wiring

Although the ADAM-5510M systems are designed for a standard industrial unregulated 24 V_{DC} power supply, they accept any power unit that supplies within the range of +10 to +30 V_{DC} . The power supply ripple must be limited to 200 mV peak-to-peak, and the immediate ripple voltage should be maintained between +10 and +30 V_{DC}. Screw terminals +Vs and GND are for power supply wiring.

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

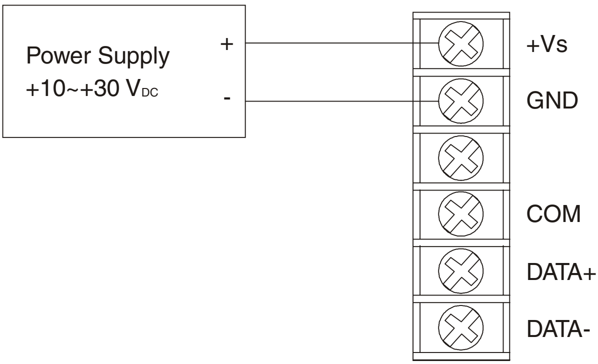


Figure 2-8: ADAM-5510M power wiring

2.3.2 I/O modules wiring

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

1. The terminal block accepts wires from 0.5 mm 2 to 2.5 mm.
2. Always use a continuous length of wire. Do not combine wires to make them longer.
3. Use the shortest possible wire length.
4. Use wire trays for routing where possible.
5. Avoid running wires near high energy wiring.
6. Avoid running input wiring in close proximity to output wiring where possible.
7. Avoid creating sharp bends in the wires.

2.3.3 System Network Connection

The ADAM-5510M has four communication ports. These ports allowed you to program, configure, monitor, and integrate the remote devices.

Network Connection for System Configuration and Download

The ADAM-5510M has a programming port with a DB-9 connection. This port (COM3) allows you to program, configure, and troubleshoot the ADAM-5510M from your host computer. The programming port has an RS-232 interface and only uses TX, RX, and GND signals. The cable connection and the pin assignment are as follows:

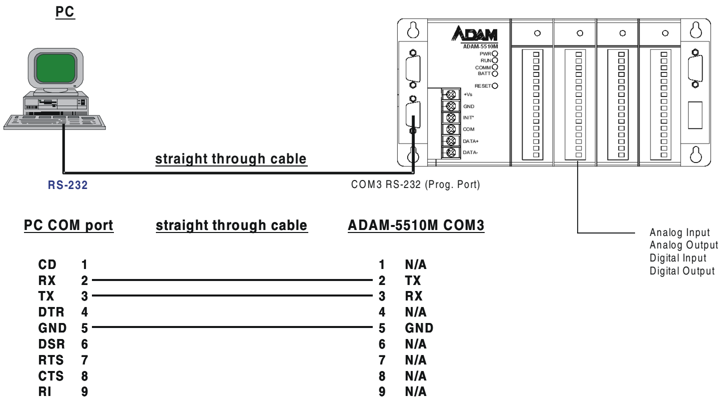


Figure 2-9: System Configuration Wiring

RS-232 Network Connection for System Monitoring and Integration

Since the connection for an RS-232 interface is not standardized, different devices implement the RS-232 connection in different ways. If you are having problems with a serial device, be sure to check the pin assignments for the connector. The following table shows the pin assignments for the ADAM-5510M RS-232 COM ports.

Pin No.	Description
Pin 1	DCD
Pin 2	Rx
Pin 3	Tx
Pin 4	DTR
Pin 5	GND
Pin 6	DSR
Pin 7	RTS
Pin 8	CTS
Pin 9	RI

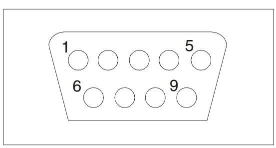


Table 2.6 RS-232 Port Pin Assignment

The COM1 is dedicated as an RS-232 interface and the COM4 is an RS-232/485 selectable DB-9 connector. See Figure 2-10.

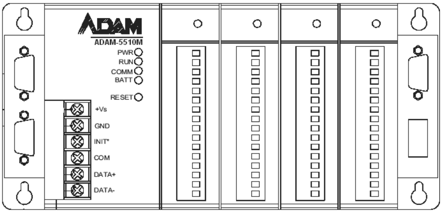


Figure 2-10: RS-232 System Configuration Wiring

RS-485 Network Connection for System Monitoring and Integration

The ADAM-5510M provides RS-485 interfaces for multi-drop network integration. The COM2 is a dedicate RS-485 interface (Screw terminals DATA- and DATA+ are used for making the COM2 RS-485 connections). The COM4 is an RS-232/485 selectable DB-9 connector. Usually, you will need to prepare an ADAM-4520 RS232/485 converter to link with host PC for data monitoring See Figure 2-11.

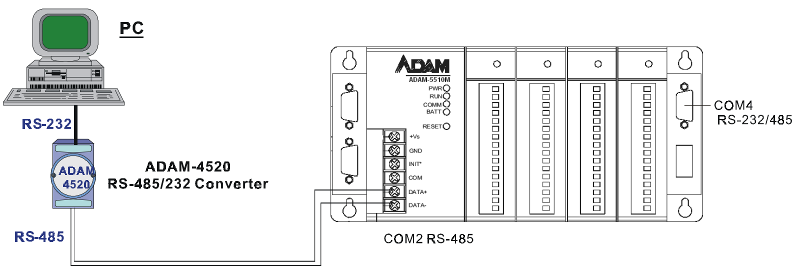
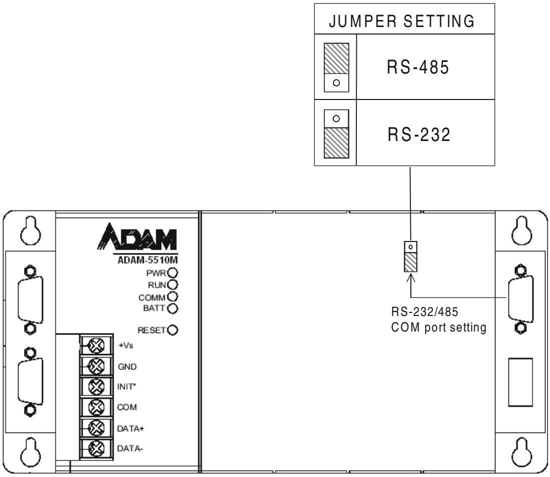


Figure 2-11: System Monitoring Wiring

Note: The Communication Mode of COM4 is setting by the Jumper 1 on the backplane. Please refer to Figure 2-12 to set the communication interface you prefer to.



2.4 Software Installation

When main unit installation is completed, you may begin to configure the I/O modules and download program to the ADAM-5510M. ADAM-5510M Systems come packaged with a Utility CD, containing ADAM Product series Utilities as system configuration tool. While you Insert the CD into the CD drive (e.g. D:) of the host PC, the Utility software setup menu will start up automatically. Click the ADAM-5510M icon to execute the setup program, and there will be a Utility executive program installed in your host PC.

See Chapter 4: I/O Configuration and Download Program for the detail operation.

Chapter 3
I/O Configuration and Program Download

This chapter explains how to use the ADAM-5510M Windows Utility to configure the I/O modules and download application programs into the ADAM-5510M system. There will also be a section to introduce the programming detail about the ADAM-5510M controller.

3.1 System Hardware Configuration

Before the system configuration, you will need to setup the environment as we mentioned in Chapter 2-1: System Requirements.

3.2 Install Utility Software on Host PC

ADAM-5510M systems packaged with a Utility CD, containing ADAM Product series Utilities as system configuration tools. While you Insert the CD into the CD drive (e.g. D:) of the host PC, the Utility software setup menu will start up automatically.

Click the ADAM-5510M icon to execute the setup program. There will be a shortcut of the Utility executive program on Windows' desktop after completing the installation.

3.3 ADAM-5510M Windows Utility Overview

The Windows Utility offers a graphical interface that helps you configure the ADAM-5510M controller and I/O modules. The following guidelines will give you some brief instructions on how to use this Utility.

- ☐ Main Menu
- ☐ I/O Module Configuration
- ☐ Program Download
- ☐ Terminal Emulation

3.3.1 Main Menu

Double Click the icon of ADAM-5510M Windows Utility shortcut , the Operation screen will pop up as Figure 3-1. This main menu is similar to general ftp software. The left screen is the file system in the host PC, and the right screen is the file system in the ADAM-5510M.

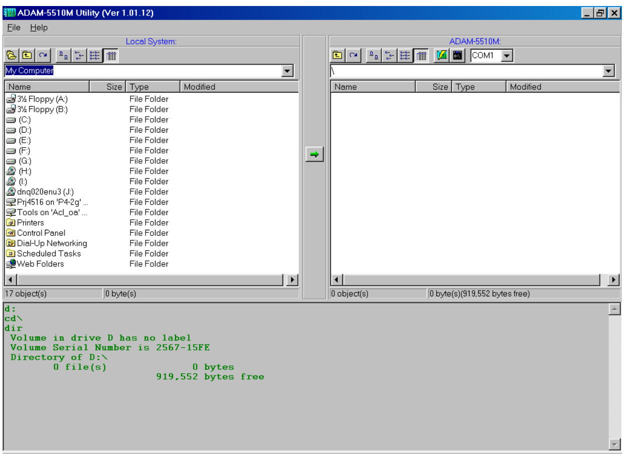


Figure 3-1 ADAM-5510M Windows Utility Startup Display

3.3.2 I/O Module Configuration

Click the  icon on the menu bar, the Windows Utility will detect the I/O modules inserted in the ADAM-5510M main unit automatically. Then you can configure the I/O module one by one.

Digital I/O: As Figure 3-2, you can see the Node ID information and the module description.

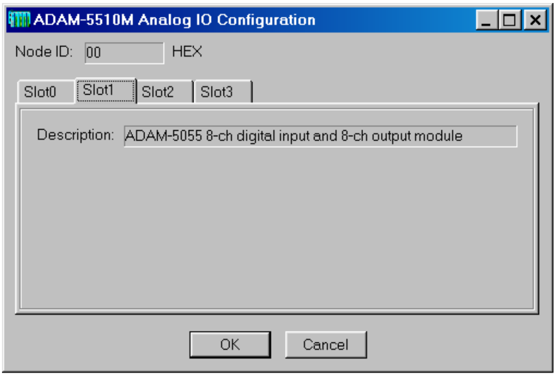


Figure 3-2 Digital I/O Configuration

Analog Input: As Figure 3-3, you can configure the range of the Analog input channels.

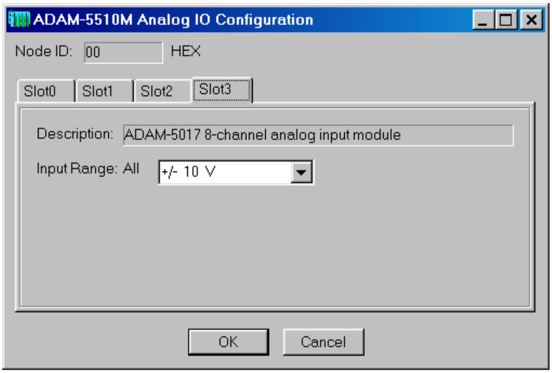


Figure 3-3 Analog Input Configuration

Analog Output: As Figure 3-4, you can configure the ranges of the Analog Output channels.

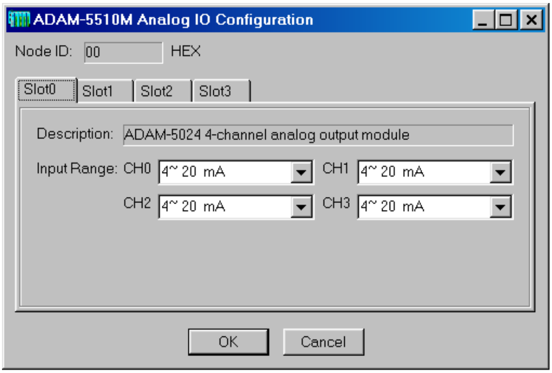


Figure 3-4 Analog Output Configuration

3.3.3 Program Download

Once you complete the application program, you can download it into the ADAM-5510M through the Windows Utility. Choose the specific execution file in the left screen (local system, your PC), then click the  icon. The specific file will begin to transfer to the ADAM-5510M system.

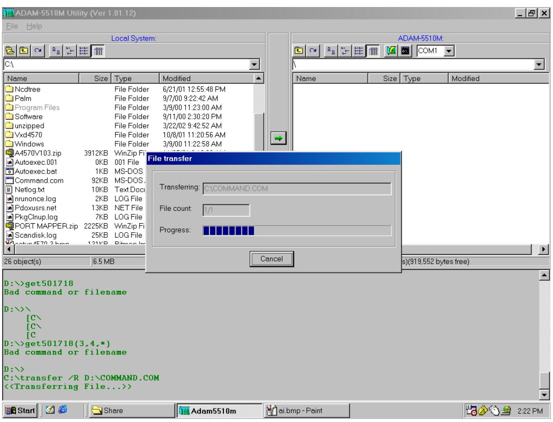


Figure 3-5 File transfer to ADAM-5510M

3.3.4 Terminal Emulation

In order to provide a convenience operation environment for users, the Windows Utility also provides the DOS mode operation interface in the button of the display. You can click the  icon to switch to the DOS mode for execute the DOS commands.

3.4 Programming detail about the ADAM-5510M Controller

The operating system of ADAM-5510M is ROM-DOS, an MS-DOS equivalent system. It allows users to run application programs written in assembly language as well as high level languages such as C or C++. Certainly, there will be some limitations when running application programs in the ADAM-5510M. In order to build successful applications, please keep the following limitations and concerns in mind.

3.4.1 Mini BIOS functions

The ADAM-5510M provides only three serial communication ports (not includes programming port) for connecting peripherals, so the mini BIOS of ADAM-5510M only provides 10 function calls. Since the user's program cannot use other BIOS function calls, the ADAM-5510M may not work as intended.

Additionally, certain language compilers such as QBASIC directly call BIOS functions that are not executable in ADAM-5510M. The ADAM-5510M mini BIOS function calls are listed in the following table.

Function	Sub-function	Task
07h		186 or greater cd-processor esc instruct
10h	0eh	TTY CLear output
11h		Get equipment
12h		Get memory size
15h	87h	Extended menmory read
	88h	Extended menmory size
	c0h	PS/2 or AT style A20 Gate table
16h	0	Read TTY char
	1	Get TTY status
	2	Get TTY flags
18h		Print "Failed to BOOT ROM-DOS" message
19h		Reboot system
1ah	0	Get tick count
	1	Set tick count
	2	Get real time clock
	3	Set real time clock
	4	Get data
	5	Set data
1ch		Timer tick

Table 3-1: ADAM-5510M mini BIOS function calls

3.4.2 Converting program codes

The ADAM-5510M has an 80188 CPU. Therefore, programs downloaded into its flash ROM must first be converted into 80186 or 80188 compatible code, and the floating point operation must be set to emulation mode. For example, if you were to develop your application program in Borland C, you would compile the program as indicated in the screen below.

Figure 3-6: Converting program codes

3.4.3 Other limitation

Certain critical files are always kept in flash ROM, such as operating system, BIOS, and monitoring files. The ADAM-5510M provides an additional 400KB free space of flash memory and up to 512 KB backup RAM for downloading and operation user applications.

3.4.4 Programming the watchdog timer

The ADAM-5510M is equipped with a watchdog timer function that resets the CPU or generates an interrupt if processing comes to a standstill for any reason. This feature increases system reliability in industrial standalone and unmanned environments.

If you decide to use the watchdog timer, you must write a function call to enable it. When the watchdog timer is enabled, it must be cleared by the application program at intervals of less than 1.6 seconds. If it is not cleared at the required time intervals, it will activate and reset the CPU, or generate a NMI (Non-Maskable Interrupt). You can use a function call in your application program to clear the watchdog timer. At the end of your program, you still need a function call to disable the watchdog timer.

Note: In order to provide one stop shop service, Advantech now also provide the Turbo C++ IDE (Integration Developing environment) as users' application programming tool. Please contact local sales for detail order information.

Chapter 4
I/O Module Introduction

Introduction

This chapter introduces the detail specifications functions and application wiring of each ADAM-5000 I/O modules. To organize an ADAM-5510M data acquisition & control system, you need to select I/O modules to interface the main unit with field devices or processes that you have previously determined. Advantech provides 19 types of ADAM-5000 I/O modules for various applications so far. Following table is the I/O modules support list we provided for user's choice.

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

Table 4-1: I/O Module Support List

4.1 Analog Input Modules

Analog input modules use an A/D converter to convert sensor voltage, current, thermocouple or RTD signals into digital data. The digital data is then translated into engineering units. The analog input modules protect your equipment from ground loops and power surges by providing opto-isolation of the A/D input and transformer based isolation up to 3,000 V_{DC}.

ADAM-5013 3-channel RTD input module

The ADAM-5013 is a 16-bit, 3-channel RTD input module that features programmable input ranges on all channels. This module is an extremely cost-effective solution for industrial measurement and monitoring applications. Its opto-isolated inputs provide 3,000 V_{DC} of isolation between the analog input and the module, protecting the module and peripherals from damage due to high input line voltage.

Note: Owing to the conversion time required by the A/D converter, the initialization time of each ADAM-5013 module is 5 seconds. Thus the total initialization time will be about 20 seconds if all 4 I/O slots in an ADAM-5000 main unit contain ADAM-5013 modules.

ADAM-5013

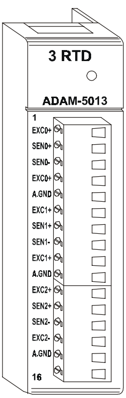
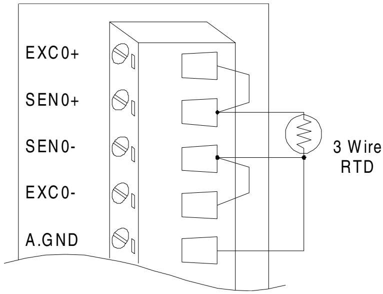


Figure 4-1: ADAM-5013 module frontal view

Application wiring



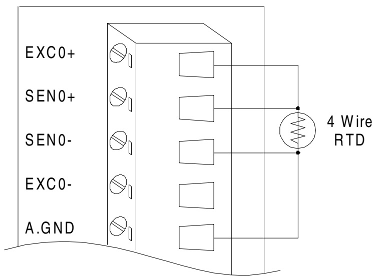


Figure 4-2: RTD inputs

Technical specifications of ADAM-5013

Analog input channels	three
Input type	Pt or Ni RTD
RTD type and temperature range	Pt -100 to 100° C a=0.00385 Pt 0 to 100° C a=0.00385 Pt 0 to 200° C a=0.00385 Pt 0 to 600° C a=0.00385 Pt -100 to 100° C a=0.00392 Pt 0 to 100° C a=0.00392 Pt 0 to 200° C a=0.00392 Pt 0 to 600° C a=0.00392 Ni -80 to 100° C Ni 0 to 100° C
Isolation voltage	3000 V_{DC}
Sampling rate	10 samples/sec (total)
Input impedance	2 MΩ
Bandwidth	13.1 Hz @ 50 Hz, 15.72 Hz @ 60 Hz
Input connections	2, 3 or 4 wire
Accuracy	± 0.1% or better
Zero drift	± 0.015 °C/°C
Span drift	± 0.01 °C/°C
CMR@50/60 Hz	150 dB
NMR@50/60 Hz	100 dB
Power consumption	1.2 W

Table 4-2: Technical specifications of ADAM-5013

4.2 ADAM-5013 RTD Input Resistance Calibration

1. Apply power to the module and let it warm up for about 30 minutes.
2. Make sure that the module is correctly installed and is properly configured for the input range you want to calibrate. You can use the ADAM utility software to help in this.
3. Connect the correct reference self resistance between the screw terminals of the ADAM-5013 as shown in the following wiring diagram. Table 4-2 below shows the correct values of the span and zero calibration resistances to be connected. Reference resistances used can be from a precision resistance decade box or from discrete resistors with the values 60, 140, 200 and 440 ohms.

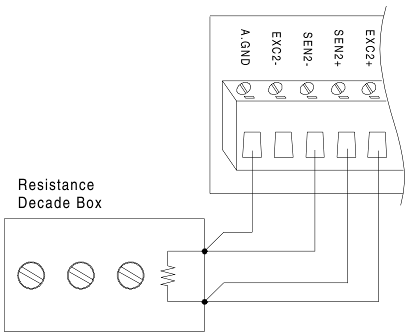


Figure 4-3: Applying calibration resistance

4. First, with the correct zero (offset) calibration resistance connected as shown above, issue a Zero Calibration command to the module using the Calibrate option in the ADAM utility software.
5. Second, with the correct span resistance connected as shown above, issue a Span Calibration command to the module using the Calibrate option in the ADAM utility software. Note that the module zero calibration must be completed prior to the span calibration.

Note: If the above procedure is ineffective, the user must first issue an RTD Self Calibration command $aaSi2 to the module and then complete steps 4 and 5 after self calibration is complete.

Calibration resistances (ADAM-5013)

Input Range Code (Hex)	Input Range	Span Calibration Resistance	Zero Calibration Resistance
20	Pt, -100 to 100° C A = 0.00385	140 Ohms	60 Ohms
21	Pt, 0 to 100° C A = 0.00385	140 Ohms	60 Ohms
22	Pt, 0 to 200° C A = 0.00385	200 Ohms	60 Ohms
23	Pt, 0 to 600° C A = 0.00385	440 Ohms	60 Ohms
24	Pt, -100 to 100° C A = 0.00392	140 Ohms	60 Ohms
25	Pt, 0 to 100° C A = 0.00392	140 Ohms	60 Ohms
26	Pt, 0 to 200° C A = 0.00392	200 Ohms	60 Ohms
27	Pt, 0 to 600° C A = 0.00392	440 Ohms	60 Ohms
28	Ni, -80 to 100° C	200 Ohms	60 Ohms
29	Ni, 0 to 100° C	200 Ohms	60 Ohms

Table 4-3: Calibration resistances of ADAM-5013

ADAM-5017 8-channel analog input module

The ADAM-5017 is a 16-bit, 8-channel analog differential input module that provides programmable input ranges on all channels. It accepts millivolt inputs (±150mV, ±500mV), voltage inputs (±1V, ±5V and ±10V) and current input (±20 mA, requires 125 ohms resistor). The module provides data to the host computer in engineering units (mV, V or mA). This module is an extremely cost-effective solution for industrial measurement and monitoring applications. Its opto-isolated inputs provide 3,000 V_{DC} of isolation between the analog input and the module, protecting the module and peripherals from damage due to high input line voltage. Additionally, the module uses analog multiplexers with active overvoltage protection. The active protection circuitry assures that signal fidelity is maintained even under fault conditions that would destroy other multiplexers. This module can withstand an input voltage surge of 70 Vp-p with ±15 V supplies.

ADAM-5017

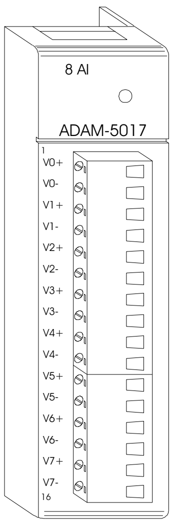


Figure 4-4: ADAM-5017 module frontal view

Application wiring

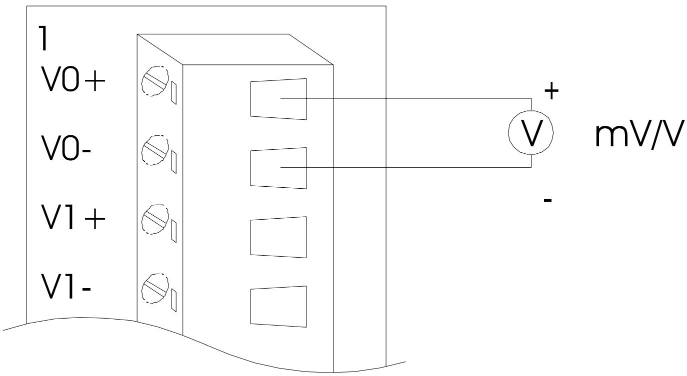


Figure 4-5: Millivolt and volt input

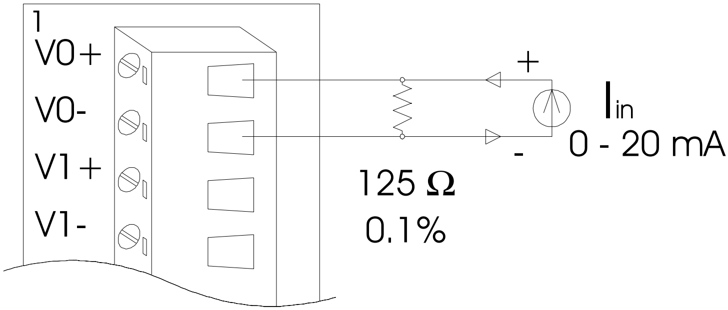


Figure 4-6: Process current input

Note: To keep measurement accuracy please short the channels that are not in use.

Technical specifications of ADAM-5017

Analog Input Channels	Eight differential
Input Type	mV, V, mA
Input Range	± 150 mV, ± 500 mV, ± 1 V, ± 5 V, ± 10 V and ± 20 mA
Isolation Voltage	3000 V_{DC}
Sampling Rate	10 samples/sec (total)
Analog Input Signal Limit	15 V max.
Max. allowable voltage difference between two connectors in a module	15 V max.
Input Impedance	2 Mohms
Bandwidth	13.1 Hz @ 50 Hz, 15.72 Hz @ 60 Hz
Accuracy	± 0.1% or better
Zero Drift	± 1.5 µV/°C
Span Drift	± 25 PPM/°C
CMR @ 50/60 Hz	92 dB min.
Power Requirements	+ 10 to + 30 V_{DC} (non-regulated)
Power Consumption	1.2 W

Table 4-4: Technical specifications of ADAM-5017

ADAM-5017H 8-channel high speed analog input module

The ADAM-5017H is a 12-bit plus sign bit, 8-channel analog differential input module that provides programmable input ranges on each channel. It accepts millivolt inputs (± 500 mV, 0-500 mV), voltage inputs (±1 V, 0-1 V, ±2.5 V, 0-2.5 V, ±5 V, 0-5 V, ±10 V and 0-10 V) and current inputs (0-20 mA and 4-20 mA; requires a 125 ohms resistor). The module provides data to the host microprocessor in engineering units (mV, V or mA) or two's complement format. Its sampling rate depends on the data format received: up to 100 Hz (total). Space is reserved for 125-ohm, 0.1%, 10 ppm resistors (See Figure 4-9). Each input channel has 3000 V_{DC} of optical isolation between the outside analog input line and the module, protecting the module and peripherals from high input line voltages. Additionally, the module uses analog multiplexers with active overvoltage protection. The active protection circuitry assures that signal fidelity is maintained even under fault conditions that would destroy other multiplexers. The analog inputs can withstand a constant 70 Vp-p input with ±15 V supplies.

ADAM-5017H

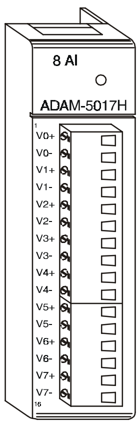


Figure 4-7: ADAM-5017H module frontal view

Application wiring

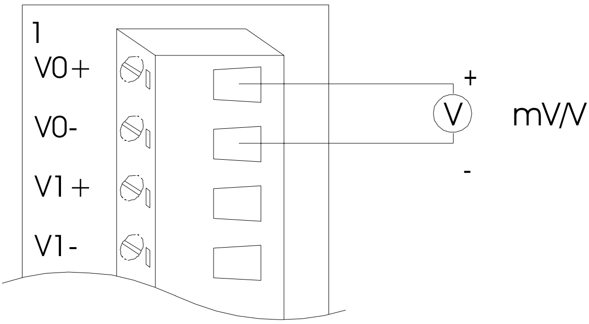


Figure 4-8: Millivolt and volt input

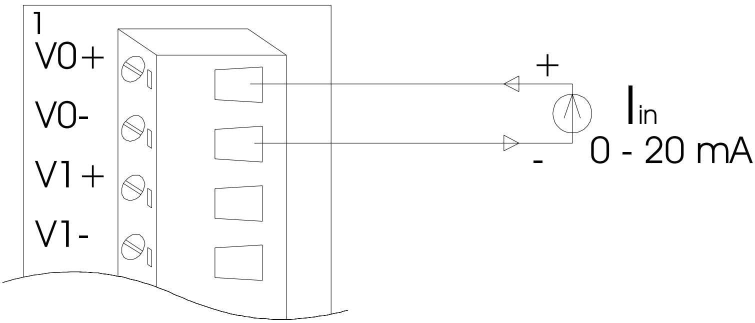


Figure 4-9: Process current input

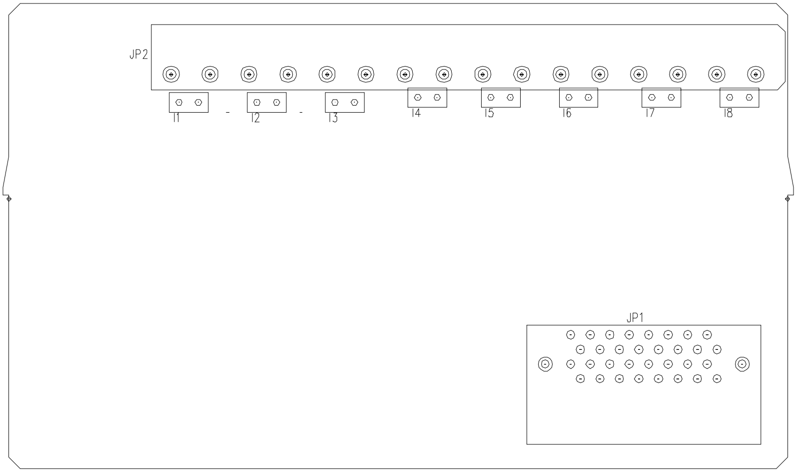


Figure 4-10: Locations of 125-ohm resistors

Note: To maintain measurement accuracy please short channels not in use.

Technical specifications of ADAM-5017H

Analog Input Channels	8 differential
ADC Resolution	12 bits, plus sign bit
Type of ADC	Successive approximation
Isolation Voltage	3000 V_{DC}
Sampling Rate	100 Hz
Input Impedance	20 Mohms (voltage inputs); 125 ohms (current inputs)
Signal Input Bandwidth	1000 Hz for both voltage inputs and current inputs
Analog Signal Range	±15 V max.
Analog Signal Range for any two measured Pins	±15 V max.
Power Requirements	+10 to +30 V_{DC} (non-regulated)
Power Consumption	1.8 W

Table 4-5: Technical specifications of ADAM-5017H

	Input Range	With Overranging	Offset Error @ 25° C	Offset Error @ -10 to +70° C	Gain Error @ 25° C	Gain Error @ -10 to +70° C	Offset Drift	Gain Drift	Display Resolution
Voltage Inputs	0 ~ 10 V	0 ~ 11 V	±1 LSB	±2 LSB	±1 LSB	±2 LSB	17 µV/°C	50 ppm/°C	2.7 mV
	0 ~ 5 V	0 ~ 5.5 V	±1 LSB	±2 LSB	±1.5 LSB	±2 LSB	16 µV/°C	50 ppm/°C	1.3 mV
	0 ~ 2.5 V	0 ~ 2.75 V	±1 LSB	±2 LSB	±1.5 LSB	±2 LSB	20 µV/°C	55 ppm/°C	0.67 mV
	0 ~ 1 V	0 ~ 1.375 V	±1 LSB	±2.5 LSB	±2 LSB	±2.5 LSB	20 µV/°C	60 ppm/°C	0.34 mV
	0 ~ 500 mV	0 ~ 687.5 mV	-	±5 LSB	±3 LSB	±3.5 LSB	20 µV/°C	67 ppm/°C	0.16 mV
	± 10 V	±11 V	±1 LSB	±2 LSB	±1 LSB	±2 LSB	17 µV/°C	50 ppm/°C	2.7 mV
	± 5 V	±0 ~ 5.5 V	±1 LSB	±2 LSB	±1.5 LSB	±2 LSB	17 µV/°C	50 ppm/°C	1.3 mV
	± 2.5 V	±0 ~ 2.75 V	±1 LSB	±2 LSB	±1.5 LSB	±2 LSB	20 µV/°C	55 ppm/°C	0.67 mV
	± 1 V	±0 ~ 1.375 V	±1 LSB	±2.5 LSB	±2 LSB	±2.5 LSB	20 µV/°C	60 ppm/°C	0.34 mV
	± 500 mV	±0 ~ 687.5 mV	-	±5 LSB	±3 LSB	±3.5 LSB	20 µV/°C	67 ppm/°C	0.16 mV
Current Inputs	0 ~ 20 mA	22 mA	±1 LSB	±1 LSB	±1.5 LSB	±2 LSB	nA/°C	ppm/°C	5.3 µA
	4 ~ 20 mA	22 mA	±1 LSB	±1 LSB	±1.5 LSB	±2 LSB	nA/°C	ppm/°C	5.3 µA

Table 4-6: ADAM-5017H input signal ranges

ADAM-5018 7-channel thermocouple input module

The ADAM-5018 is a 16-bit, 7-channel thermocouple input module that features programmable input ranges on all channels. It accepts millivolt inputs (±15 mV, ±50 mV, ±100 mV, ±500 mV), voltage inputs (±1 V, ±2.5 V), current input (±20 mA, requires 125 ohms resistor) and thermocouple input (J, K, T, R, S, E, B).

The module forwards the data to the host computer in engineering units (mV, V, mA or temperature °C). An external CJC on the plug-in terminal is designed for accurate temperature measurement.

ADAM-5018

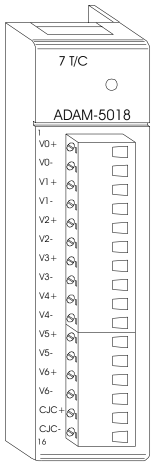


Figure 4-11: ADAM-5018 module frontal view

Application wiring

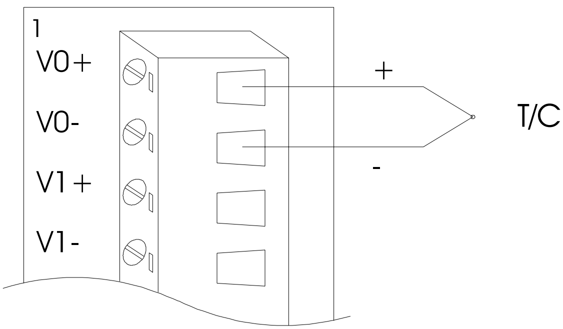


Figure 4-12: *Thermocouple input*

Technical specifications of ADAM-5018

Analog Input Channels	Seven differential
Input Type	mV, V, mA, Thermocouple
Input Range	± 15 mV, ± 50 mV, ± 100 mV, ± 500 mV, ± 1 V, ± 2.5 V and ± 20 mA
T/C Type and Temperature Range	J 0 to 760 °C K 0 to 1370 °C T -100 to 400 °C E 0 to 1400 °C R 500 to 1750 °C S 500 to 1750 °C B 500 to 1800 °C
Isolation Voltage	3000 V_{DC}
Sampling Rate	10 samples/sec (total)
Input Impedance	2 Mohms
Bandwidth	13.1 Hz @ 50 Hz, 15.72 Hz @ 60 Hz
Accuracy	± 0.1% or better
Zero Drift	± 0.3 µV/°C
Span Drift	± 25 PPM/°C
CMR @ 50/60 Hz	92 dB min.
Power Consumption	1.2 W

Table 4-7: Technical specifications of ADAM-5018

4.3 Analog Output Modules

ADAM-5024 4-channel analog output module

The ADAM-5024 is a 4-channel analog output module. It receives its digital input through the RS-485 interface of the ADAM-5510 system module from the host computer. The format of the data is engineering units. It then uses the D/A converter controlled by the system module to convert the digital data into output signals.

You can specify slew rates and start up currents through the configuration software. The analog output can also be configured as current or voltage through the software utility. The module protects your equipment from ground loops and power surges by providing opto-isolation of the D/A output and transformer based isolation up to 500 V_{DC}.

Slew rate

The slew rate is defined as the slope indicated the ascending or descending rate per second of the analog output from the present to the required.

ADAM-5024

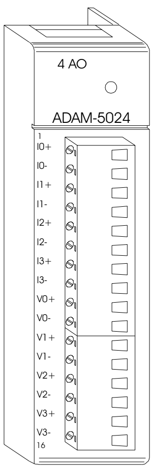


Figure 4-13: ADAM-5024 module frontal view

Application wiring

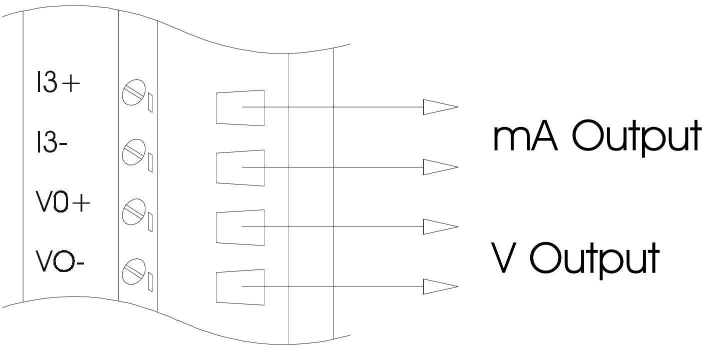


Figure 4-14: Analog output

Technical specifications of ADAM-5024

Analog Output Channels	Four
Output Type	V, mA
Output Range	0-20mA, 4-20mA, 0-10V
Isolation Voltage	3000 Vdc
Output Impedance	0.5 Ohms
Accuracy	±0.1% of FSR for current output ±0.2% of FSR for voltage output
Zero Drift	Voltage output: ±30 μV/ºC Current output: ±0.2 μA/ºC
Resolution	±0.015% of FSR
Span Temperature Coefficient	±25 PPM/ºC
Programmable Output Slope	0.125-128.0 mA/sec 0.0625-64.0 V/sec
Current Load Resistor	0-500 Ohms (source)
Power Consumption	2.5W (Max.)

Table 4-8: Technical specifications of ADAM-5024

4.4 Analog I/O Modules Calibration

Analog input/output modules are calibrated when you receive them. However, calibration is sometimes required. No screwdriver is necessary because calibration is done in software with calibration parameters stored in the ADAM-5000 analog I/O module‘s onboard EEPROM.

The ADAM-5000 system comes with the ADAM utility software that supports calibration of analog input and analog output. Besides the calibration that is carried out through software, the modules incorporate automatic Zero Calibration and automatic Span Calibration at bootup or reset.

Analog input module calibration

Modules: ADAM-5017, 5017H, 5018

1. Apply power to the ADAM-5000 system that the analog input module is plugged into and let it warm up for about 30 minutes
2. Assure that the module is correctly installed and is properly configured for the input range you want to calibrate. You can do this by using the ADAM utility software.
3. Use a precision voltage source to apply a span calibration voltage to the module‘s V0+ and V0- terminals. (See Tables 5-2 and 5-3 for reference voltages for each range.)

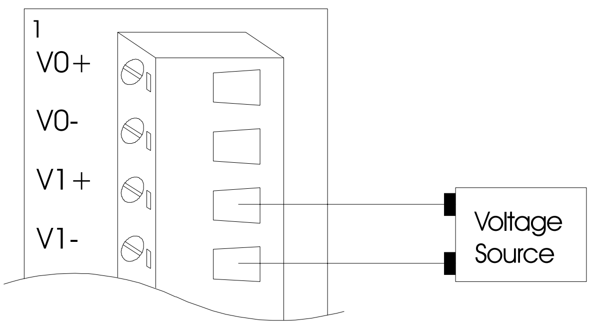


Figure 4-15: Applying calibration voltage

4. Execute the Zero Calibration command (also called the Offset Calibration command).

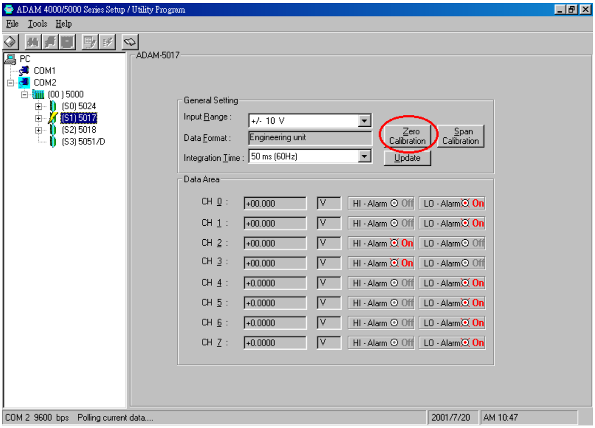


Figure 4-16: Zero calibration

5. Execute the Span Calibration command. This can be done with the ADAM utility software.

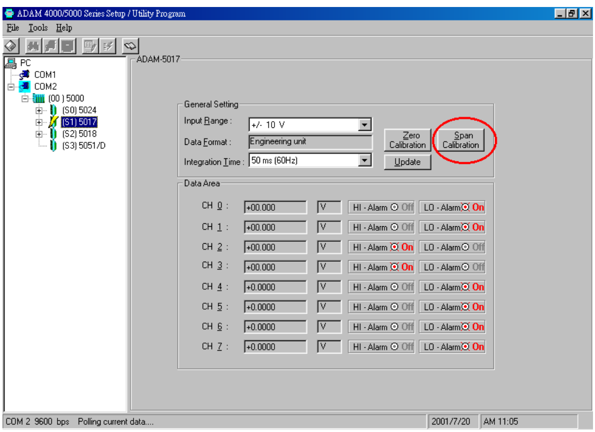


Figure 4-17: Span calibration

6. CJC Calibration (only for T/C input module)

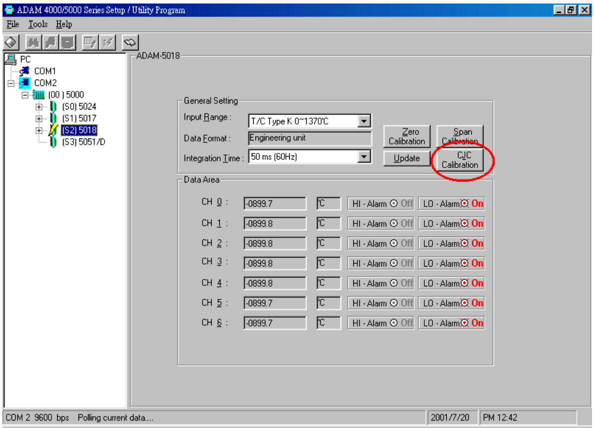


Figure 4-18: Cold junction calibration

*** Note:** Zero calibration and span calibration must be completed before CJC calibration. To calibrate CJC, the thermocouple attached to ADAM-5018 and a standard thermometer should be used to measure a standard known temperature, such as the freezing point of pure water. The amount of offset between the ADAM-5018 and the standard thermometer is then used in the ADAM utility to complete CJC calibration.

Calibration voltage (ADAM-5017/5018)

Module	Input Range Code (Hex)	Input Range	Span Calibration Voltage
5018	00h	±15 mV	+15 mV
	01h	±50 mV	+50 mV
	02h	±100 mV	+100 mV
	03h	±500 mV	+500 mV
	04h	±1 mV	+1 V
	05h	±2.5 V	+2.5 V
	06h	±20 mV	+20 mA (1)
	0Eh	J thermocouple 0 to 1370° C	+50 mV
	0Fh	K thermocouple 0 to 1370° C	+50 mV
	10h	T thermocouple -100 to 400° C	+22 mV
	11h	E thermocouple 0 to 1000° C	+80 mV
	12h	R thermocouple 500 to 1750° C	+22 mV
	13h	S thermocouple 500 to 1800° C	+22 mV
	14h	B thermocouple 500 to 1800° C	+152 mV
5017	07h	Not used	
	08h	±10 V	+10 V
	09h	±5 V	+5 V
	0Ah	±1 V	+1 V
	0Bh	±500 mV	+500 mV
	0Ch	±150 mV	+150 mV
	0Dh	±20 mA	+20 mV (1)

Table 4-9: Calibration voltage of ADAM-5017/5018

Calibration voltage (ADAM-5017H)

Module	Input Range Code (Hex)	Input Range	Span Calibration Voltage
5017H	00h	±10 V	+10 V
	01h	0 ~ 10 V	+10 V
	02h	±5 V	+5 V
	03h	0 ~ 5 V	+5 V
	04h	±2.5 V	+2.5 V
	05h	0 ~ 2.5 V	+2.5 V
	06h	±1 V	+1 V
	07h	0 ~ 1 V	+1 V
	08h	±500 mV	+500 mV
	09h	0 ~ 500 mV	+500 mV
	0ah	4 ~ 20 mA	*(1)
	0bh	0 ~ 20 mA	*(1)

Table 4-10: Calibration voltage of ADAM-5017H

Note: You can substitute 2.5 V for 20 mA if you remove the current conversion resistor for that channel. However, the calibration accuracy will be limited to 0.1% due to the resistor's tolerance.

Analog output module calibration

The output current of analog output modules can be calibrated by using a low calibration value and a high calibration value. The analog output modules can be configured for one of two ranges: 0-20 mA and 4-20 mA. Since the low limit of the 0-20 mA range (0 mA) is internally an absolute reference (no power or immeasurably small power), just two levels are needed for calibration: 4 mA and 20 mA.

1. Apply power to the ADAM-5000 system including the analog output module for about 30 minutes.
2. Assure that the module is correctly installed and that its configuration is according to your specifications and that it matches the output range you want to calibrate. You can do this by using the ADAM utility software.
3. Connect either a 5-digit mA meter or voltmeter with a shunt resistor (250 ohms, .01 % and 10 ppm) to the screw terminals of the module.

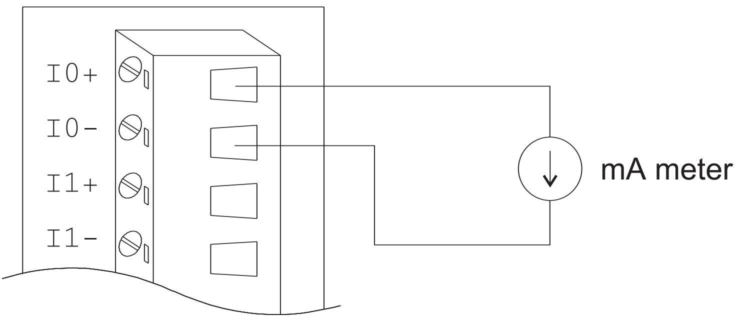


Figure 4-19: Output module calibration

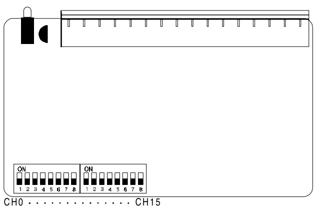
4. Issue the Analog Data Out command to the module with an output value of 4 mA.
5. Check the actual output value at the modules terminals. If this does not equal 4 mA, use the "Trim" option in the "Calibrate" submenu to change the actual output. Trim the module until the mA meter indicates exactly **4 mA**, or in case of a voltage meter with shunt resistor, the meter indicates exactly **1 V**. (When calibrating for **20 mA** using a voltage meter and shunt resistor, the correct voltage should be **5 V**.)
6. Issue the 4 mA Calibration command to indicate that the output is calibrated and to store the calibration parameters in the module's EEPROM.
7. Execute an Analog Data Out command with an output value of 20 mA. The module's output will be approximately 20 mA.
8. Execute the Trim Calibration command as often as necessary until the output current is equal to exactly 20 mA.
9. Execute the 20 mA Calibration command to indicate that the present output is exactly 20 mA. The analog output module will store its calibration parameters in the unit's EEPROM.

4.5 Digital Input/Output Modules

ADAM-5050 16-channel universal digital I/O module

The ADAM-5050 features sixteen digital input/output channels. Each channel can be independently configured to be an input or an output channel by the setting of its DIP switch. The digital outputs are open-collector transistor switches that can be controlled from the ADAM-5000. The switches can also be used to control solid-state relays, which in turn can control heaters, pumps and power equipment. The ADAM-5000 can use the module's digital inputs to determine the state of limit or safety switches, or to receive remote digital signals.

Warning! ***A channel may be destroyed if it is subjected to an input signal while it is configured to be an output channel.***



Dip Switch Key
ON = Digital Output
OFF = Digital Input

Figure 4-20: Dip switch setting for digital I/O channel

ADAM-5050

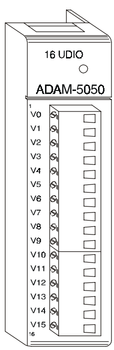


Figure 4-21: ADAM-5050 module frontal view

Application wiring

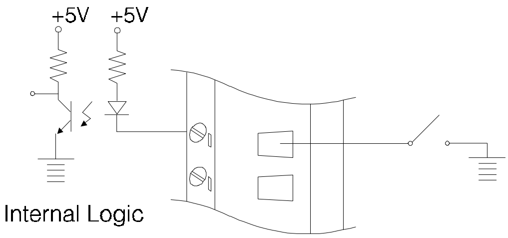


Figure 4-22: Dry contact signal input (ADAM-5050)

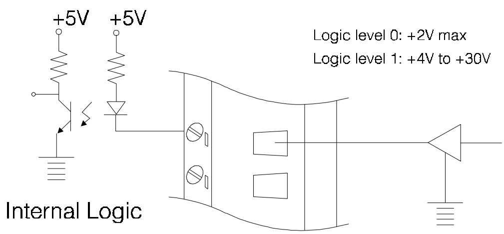


Figure 4-23: Wet contact signal input (ADAM-5050)

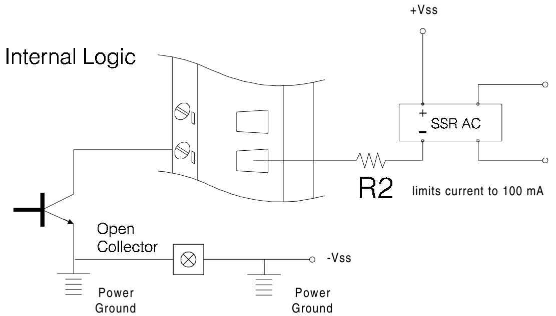


Figure 4-24: Digital output used with SSR (ADAM-5050/5056)

Technical specifications of ADAM-5050

Points	16
Channel Setting	Bitwise selectable by DIP switch
Digital Input	Dry Contact Logic Level 0: close to GND Logic Level 1: open Wet Contact Logic Level 0: +2 V max Logic Level 1: +4 V to 30 V
Digital Output	Open collector to 30 V, 100mA max load
Power Dissipation	450 mW
Power Consumption	0.4 W

Table 4-11: Technical specifications of ADAM-5050

ADAM-5051(D) 16-channel digital input module

The ADAM-5051 provides sixteen digital input channels. The ADAM-5510 can use the module's digital inputs to determine the state of limit or safety switches or to receive remote digital signals.

ADAM-5051/5051 D

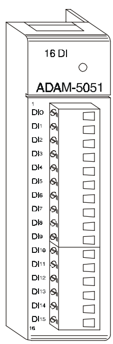


Figure 4-25: ADAM-5051 module frontal view

Application wiring

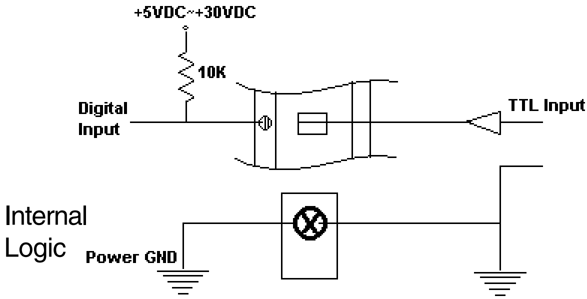


Figure 4-26: TTL input (ADAM-5051/5051D)

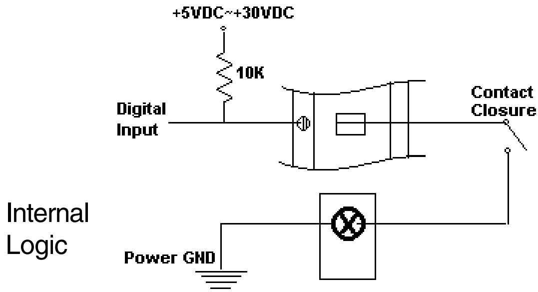


Figure 4-27: Contact closure input (ADAM-5051/5051D)

Technical specifications of ADAM-5051/5051D

Points	16
Digital input	Logic level 0: + 1 V max Logic level 1: + 3.5 to 30 V Pull up current: 0.5 mA 10 kΩ resistor to + 5 V
Power consumption	0.3 W
indicator	ADAM-5051 D only

Table 4-12: Technical specifications of ADAM-5051

ADAM-5051S 16-channel Isolated Digital Input Module with LED

The ADAM-5051S provides 16 isolated digital input channels for critical environments need individual channel isolating protection. Different from other ADAM-5000 I/O modules, ADAM-5051S designed with 21 pins plug terminal.

ADAM-5051S

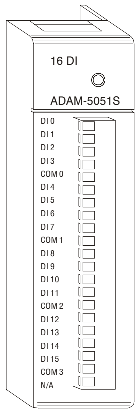


Figure 4-28: ADAM-5051S module front view

Application Wiring

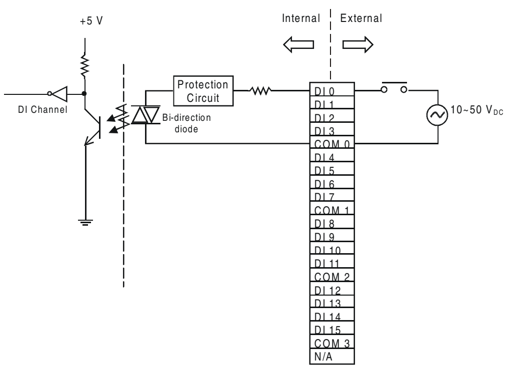


Figure 4-29: ADAM-5051S module wiring diagram

Technical specification of ADAM-5051S

Point	16(4-channel/group)
Digital Input	Logic Level 0: + 3 V max Logic Level 1: + 10 to 50 V
Optical Isolation	2500 V_{DC}
Opto-isolator response time	25 μs
Over-voltage Protection	70 V_{DC}
Power Consumption	0.8 W
LED Indicator	On when active
I/O Connector Type	21-pin plug-terminal

Table 4-13: Technical specification of ADAM-5051S

ADAM-5052 8-channel isolated digital input module

The ADAM-5052 provides eight fully independent isolated channels. All have 5000 V_{RMS} isolation to prevent ground loop effects and to prevent damage from power surges on the input lines.

ADAM-5052

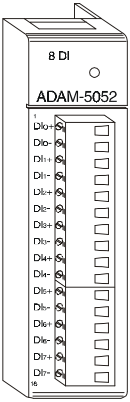


Figure 4-30: ADAM-5052 module frontal view

Application wiring

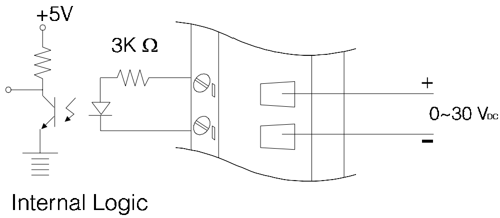


Figure 4-31: Isolation digital input (ADAM-5052)

Technical specifications of ADAM-5052

Points	8 Differential
Digital input	Logic level 0: + 1 V max Logic level 1: + 3.5 to 30 V Isolation voltage: 5000 V_{RMS} Resistance: 3 kΩ / 0.5 W
Power consumption	0.4 W

Table 4-14: Technical specifications of ADAM-5052

ADAM-5055S 16-channel Isolated Digital I/O Module with LED

The ADAM-5056S provides 8 isolated digital input and 8 isolated output channels for critical environments need individual channel isolating protection. Different from other ADAM-5000 I/O modules, ADAM-5051S designed with 21 pins plug terminal.

ADAM-5055S

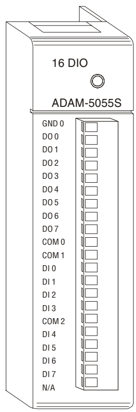


Figure 4-32: ADAM-5055S module front view

Application Wiring

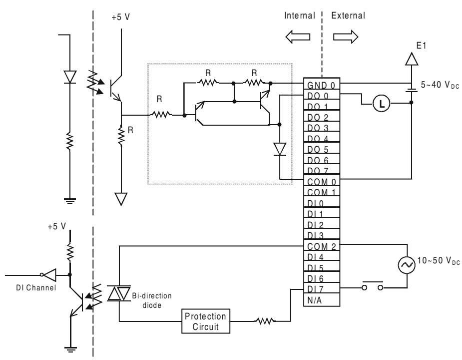


Figure 4-33: ADAM-5055S module wiring diagram

Technical specification of ADAM-5055S

Points	16
Digital Output	8 (8-channel/group)
Open collector to 40 V	200 mA max load per channel
Optical Isolation	2500 V_{DC}
Opto-isolator response time	25 µs
Supply Voltage	5 ~ 40 V_{DC}
Digital Input	8(4-channel/group) **Dry Contact** Logic Level 0: close to GNDLogic Level 1: open **Wet Contact** Logic Level 0: + 3 V maxLogic Level 1: + 10 to 50 V
Dry Contact & Wet contact	Selectable
Optical Isolation	2500 V_{DC}
Opto-isolator response time	25 µs
Over-voltage Protect	70 V_{DC}
Power Consumption	0.68 W
LED Indicator	On when active
I/O Connector Type	21-pin plug-terminal

Table 4-15: Technical specification of ADAM-5055S

ADAM-5056(D) 16-channel digital output module w/LED

The ADAM-5056 features sixteen digital output channels. The digital outputs are open-collector transistor switches that you can control from the ADAM-5000 main unit. You also can use the switches to control solid-state relays.

ADAM-5056

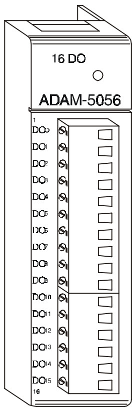


Figure 4-34: ADAM-5056 module frontal view

Application wiring

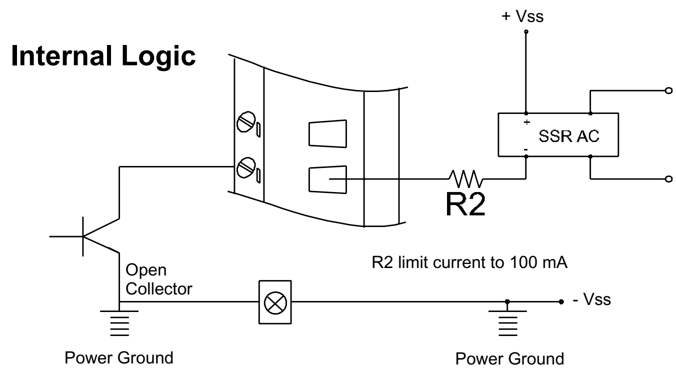


Figure 4-35: Digital output used with SSR (ADAM-5050/5056)

Technical specifications of ADAM-5056

There are 16-point digital input and 16-point digital output modules in the ADAM-5000 series. The addition of these solid state digital I/O devices allows these modules to control or monitor the interfaces between high power DC or AC lines and TTL logic signals. A command from the host converts these signals into logic levels suitable for the solid-state I/O devices.

Points	16
Digital output	Open collector to 30 V 100 mA max load
Power dissipation	450 mW
Power consumption	0.25 W

Table 4-16: Technical specifications of ADAM-5056

ADAM-5056S 16-channel Isolated Digital Output Module with LED

The ADAM-5056S provides 16 isolated digital output channels for critical environments need individual channel isolating protection. Different from other ADAM-5000 I/O modules, ADAM-5056S designed with 21 pins plug terminal.

ADAM-5056S

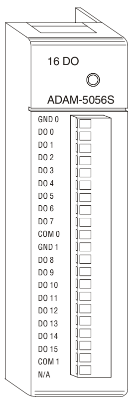


Figure 4-36: ADAM-5056S module front view

Application wiring

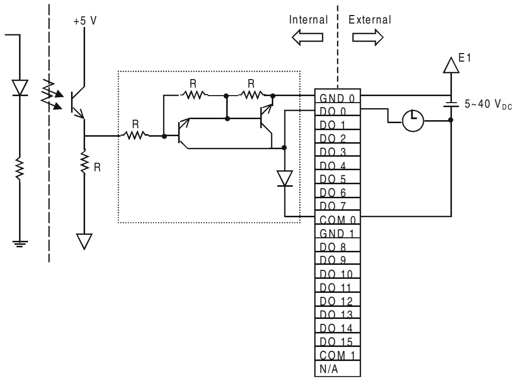


Figure 4-37: ADAM-5056S module wiring diagram

Technical Specification of ADAM-5056S

Points	16(8-channel/group)
Digital Output	Open collector to 40 V 200 mA max load per channel
Optical Isolation	2500 V_{DC}
Opto-isolator response time	25 µs
Supply Voltage	5 ~ 40 V_{DC}
Power consumption	0.6 W
LED Indicator	On when active
I/O Connector Type	21-pin plug-terminal

Table 4-17: Technical specification of ADAM-5055S

ADAM-5056SO 16-channel Isolated Digital Output Module with LED

The ADAM-5056SO provides 16 channels source type isolated digital output for critical environments need individual channel isolating protection. Addition to the source output wiring, all of the specification and command sets are the same with ADAM-5056S.

ADAM-5056SO

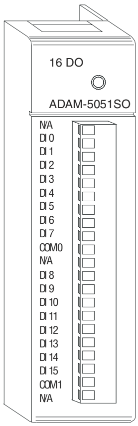


Figure 4-38: ADAM-5056SO module front view

Application wiring

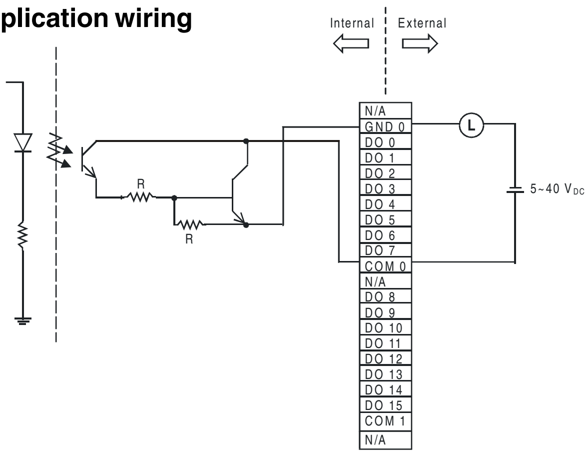


Figure 4-39: ADAM-5056SO module wiring diagram

Technical Specification of ADAM-5056SO

Points	16(8-channel/group)
Digital Output	Open collector to 40 V 200 mA max load per channel
Optical Isolation	2500 VDC
Opto-isolator response time	25 us
Supply Voltage	5 ~ 40 VDC
Power consumption	0.6 W
LED Indicator	On when active
I/O Connector Type	21-pin plug-terminal

Table 4-18: Technical specification of ADAM-5056SO

4.6 Relay Output Modules

ADAM-5060 relay output module

The ADAM-5060 relay output module is a low-cost alternative to SSR modules. It provides 6 relay channels, two of Form A and four of Form C.

ADAM-5060

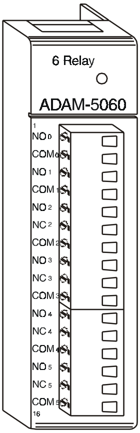


Figure 4-40: ADAM-5060 module frontal view

Application wiring

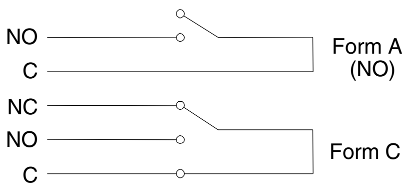


Figure 4-41: Relay output

Technical specifications of ADAM-5060

Points	6, two Form A and four Form C
Contact rating	AC: 125 V @ 0.6A; 250 V @ 0.3 A DC: 30 V @ 2 A; 110 V @ 0.6 A
Breakdown voltage	500 V_{AC} (50/60 Hz)
Relay on time (typical)	3 ms
Relay off time (typical)	1 ms
Total switching time	10 ms
Insulation resistance	1000 MΩ min. @ 500 V_{DC}
Power consumption	0.7 W

Table 4-19: Technical specifications of ADAM-5060

ADAM-5068 relay output module

The ADAM-5068 relay output module provides 8 relay channels of Form A. Switches can be used to control the solid-state relays.

ADAM-5068

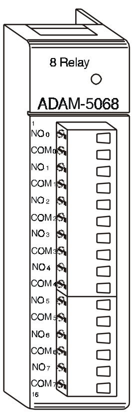


Figure 4-42: ADAM-5068 module frontal view

Application wiring

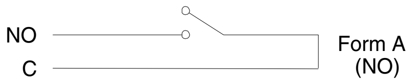


Figure 4-43: Relay output

Technical specifications of ADAM-5068

Points	8 Form A
Contact Rating	AC: 120 V @ 0.5 A DC: 30 V @ 1 A
Breakdown Voltage	500 V_{AC} (50/60 Hz)
Relay On Time (typical)	7 msec.
Relay Off Time (typical)	3 msec.
Total Switching Time	10 msec.
Power Consumption	2.0 W

Table 4-20: Technical specifications of ADAM-5068

4.7 Counter/Frequency Module

Overview

Compatible ADAM-5000 Series Main Units

ADAM-5080 is a 4-channel counter/frequency module designed to be implemented within the following Advantech ADAM-5000 series main units:
ADAM-5000/485
ADAM-5510
ADAM-5511

Please make sure that the ADAM-5080 counter/frequency module is properly inserted into the compatible main units.

ADAM-5080 4-channel Counter/Frequency Module

With ADAM-5080 4-Channel Counter/Frequency Module, users can select either counter or frequency mode for data output. ADAM-5080 offers users a variety of very flexible and versatile applications such as below:

Counter Mode or Frenquency Mode

If you want to measure the number of input signals for totalizer function, you may use counter mode to measure quantities such as movement and flow quantity. Alternatively, you can also select frequency mode to calculate the instantaneous differential of quantities such as rotating speed, frequency or flow rate, and present them in specific engineering formats.

Up/Down or Bi-direction Function

When operating in counter mode, you can choose either the Up/Down function or the Bi-direction function for different application purposes. The counter will count up or down according to your applications. This counting function helps users obtain the most accurate data.

Alarm Setting Function

While in counter mode, you can set alarm status--Disable and Latch. If you want to disable it, you can select Disable.If Latch status is selected, it means the Alarm status will be "latched" whenever the alarm being triggered. Once the alarm status being "latched," it will thereafter stay in that triggered state. Users will have to issue a "Clear Alarm Status" command to return the "latched" alarm status back to normal. Users can designate the high-limit value and low-limit value to regulate your alarm behavior through the utility program.

Digital Output Mapping

Users can either run the utility program or issue a "Set Alarm Connetion" command to designate a specific digital output module for the alarm signal to be sent through.

ADAM-5080 Module Diagram

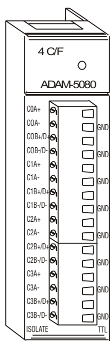


Figure 4-44: ADAM-5080 Module

ADAM-5080 Application Wiring

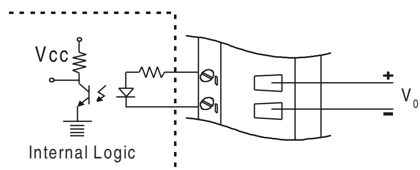


Figure 4-45: Isolated Input Level

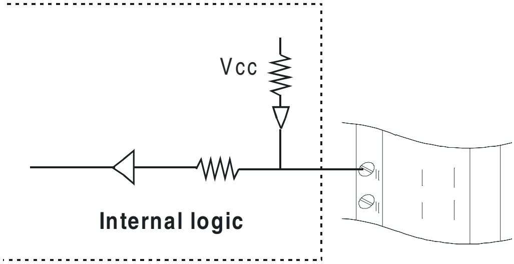


Figure 4-46: TTL Input Level

ADAM-5080 Counter/Frequency Mode Selection

Users can select Bi-direction, Up/Down Counter or Frequency option as shown in Figure 4-44.

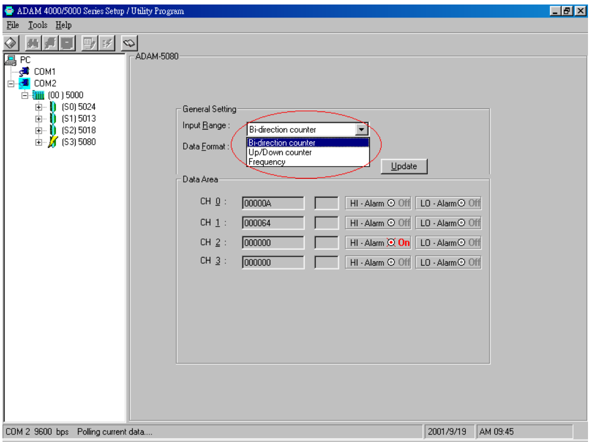


Figure 4-47: Counter / Frequency Mode

Note: All four channels of ADAM-5080 will operate simultaneously in the mode you have selected. i.e. If you switch the ADAM-5080 to Counter Mode, all four channels will operate in Counter Mode.

Features -- Counter Mode

Up/Down Counting

The Up/Down Counter Function offers two types of counting:
Up Couting (increasingly) and Down Counting (decreasingly).

Up Counting : when C0A+ and C0A- sense any input signals, the counter counts up.

Down Counting : when C0B+ and C0B- sense any input signals, the counter counts down.

On receiving Up and Down signal simultaneously, the counter will not perform each specific counting accordingly, but will remain at the previous counting value, since these simultaneous signals won't have any effect on counting values.

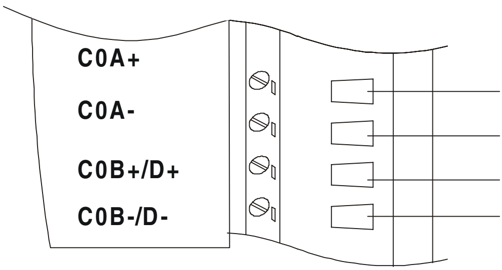


Figure 4-48: Wiring for Up/Down Counting

Note: If you need only one type of counting, connect C0A+ and C0A- for Up Counting only; or connect C0B+ and C0B- for Down Counting only.

Bi-direction Counting

For implementing Bi-derection Counting, you need to connect C0B+/D+ and C0B-/D- to implement the control function for Up/Down Counting.

Up Counting : when the input signal is within logic level "1", the counter value increases.

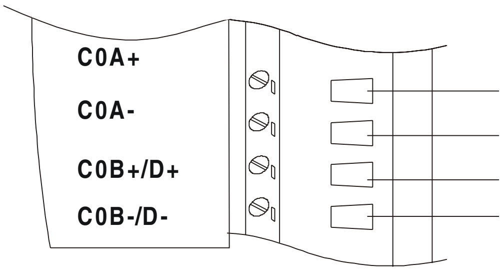


Figure 4-49: Wiring for Bi-direction Counting

Down Counting : when the input signal is within logic level "0", the counter value decreases.

Note: If users select TTL mode and don't connect C0B+ C0B-, the counter value will increase. If users select Isolated mode and don't connect C0B+ C0B-, the counter value will decrease.

Features -- Frequency Mode

If users want to select frequency mode, they can only utilize Up Counting type, and can only connect to C0A+ and C0A-.

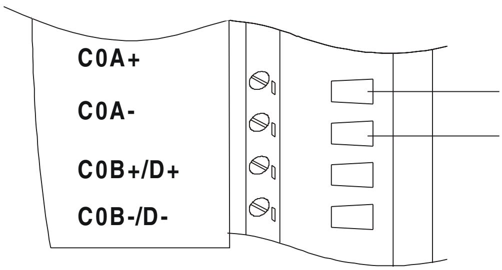


Figure 4-50: Wiring for Frequency Mode

Features -- Alarm Setting

According to your application purposes, you can run the utility program to set different limit values for High/Low Alarm.

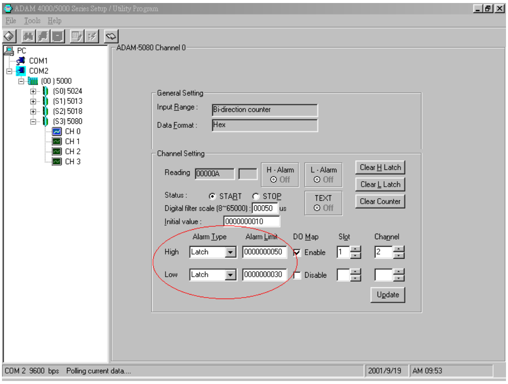


Figure 4-51: Setting Alarm Limit

Setting Initial Counter Value

In oder to utilize the alarm function, users have to set a high-alarm limit value and/or a low alarm limit value, and a initial value to fulfill the requirements for a basic alarm setting.

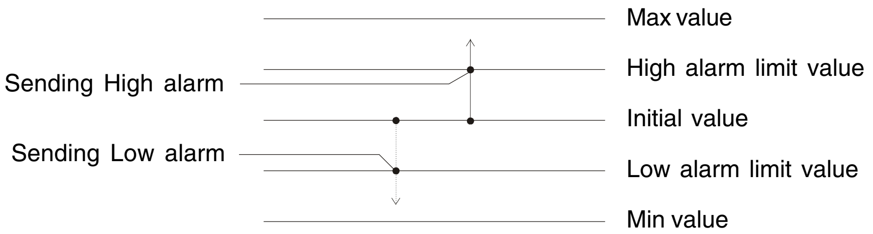


Figure 4-52: Sending Alarm Signal (recommended settings)

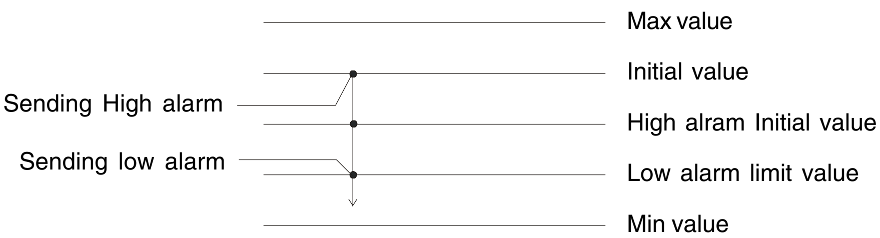


Figure 4-53: Sending Alarm Signal (settings not recommended)

Overflow Value

Overflow value is the number of times the counter value exceeds the Max/Min values you specified. When the counter value exceeds Maximum value, the overflow value increases; When the counter value goes under Minimum value, the overflow value decreases. Besides, when the counter value runs beyond the range of Max/Min value, it will continue counting from the initial value. Furthermore, if users want to check the counter value to see if it is higher or lower than the Max/Min value, they can use the "ReadOverflowFlag" library to gain a readout of the overflow value.

Getting the Totalizer Value

If users want to get the actual counter value, a formula such as follows can facilitate an easy calculation from the initial counter value, overflow value and current counter value:

$$V_{tol} = \{ |V_{ini} - V_{min} (\text{or } V_{max})| + 1 \} \times |V_{vf}| + |V_{ini} - V_{cur}|$$

V_{tol} : totalizer value

V_{ini} : initial counter value

V_{min} : min. couner value = 0 (fixed value)

V_{max} : max. counter value = 2^{32} = 4,294,967,295 (fixed value)

V_{vf} : overflow value

V_{cur} : current counter value

Example:

If the initial value =10, overflow value =4, min. value = 0, current counter value = 3, the totalizer value could be calculated as

totalizer value = {|10 - 0| + 1} x| 4 |+ |10 -3| = 51

Features--Digital Output Mapping

If users want to use Digital Output function, ADAM utility is available for setting specifically which module, channel or slot to receive the alarm signals.

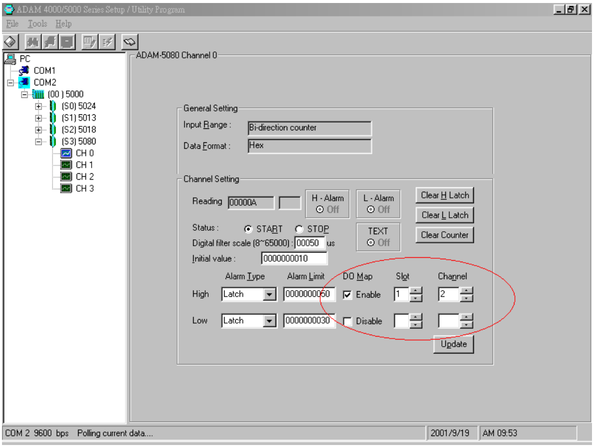


Figure 4-54: Digital Output Mapping

1: **High Alarm State**--Set Alarm state to "Latch" or "Disable".

2: **High Alarm Limt**--Set Alarm limit from 0 to 4,294,967,295.

3: **High Alarm Output Mode**--Enable or Disable D.O. Mapping.

4: **High Alarm Output Slot**--Users can select D.O Modules such as ADAM-5050, ADAM-5055, ADAM-5056, ADAM-5060, ADAM-5068 for the alarm signal to be sent through.

5: **High Alarm Output Channel**--Select Alarm Output Channel

6: **Clear Latch Alarm**--Users can Select "Enable" or "Disable" option. When selecting "Enable", the latch will be relieved and the alarm state will return to normal. Once the alarm state returns to normal, the **Clear Latch Alarm** will return to "Disable".

TTL/Isolated Input Level

According to your need, you can select either TTL or Isolated Input Level by setting the configuration for the jumpers. Select the proper jumper settings for either TTL or Isolated Input according to Figure 4-53. Please note that you must configure all six jumpers to the correct configuration for proper function.

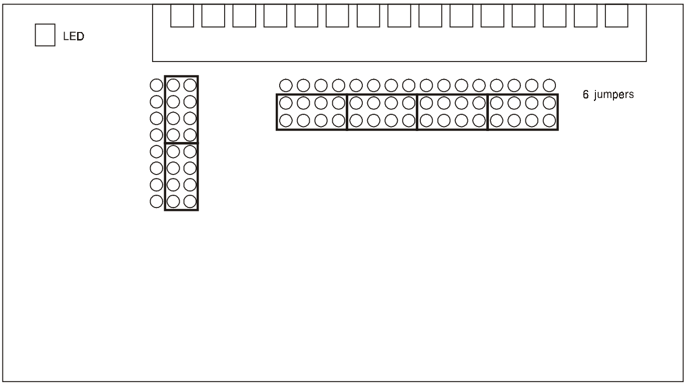
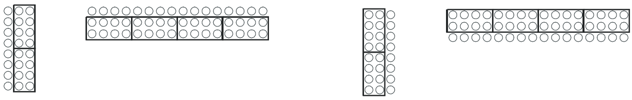


Figure 4-55: Jumper Location on the ADAM-5080 Module



TTL Input Level Isolated Input Level

Figure 4-56: TTL/Isolated Input Level Selectting

ADAM-5080 Technical Specifications

Channel	4
Input Frequency	0.3 ~ 1000 Hz max. (Frequency mode) 5000 Hz max. (Counter mode)
Input Level	Isolated or TTL level
Minimum Pulse Width	500 μ sec. (Frequency mode) 100 μ sec. (Counter mode)
Minimum Input Current	2mA (Isolated)
Isolated Input Level	Logic Level 0 : +1 V_{MAX} Logic Level 1 : + 3.5 V to 30 V
TTL Input Level	Logic Level 0 : 0 V to 0.8 V Logic Level 1 : 2.3 to 5 V
Isolated Voltage	1000 V_{RMS}
Mode	Counter (Up/Down, Bi-direction) Frequency
Programmable Digital Noise Filter	8 ~ 65000 μ sec

Table 4-21: ADAM-5080 technical specifications

4.8 Serial Module

Overview

Compatible ADAM-5000 Series Main Units

The ADAM-5090 is a 4-port RS-232 communication module to be implemented with the following Advantech ADAM-5000 series main units:

ADAM-5510 (with library Version V1.10 or above)

ADAM-5511 (with library Version V1.10 or above)

ADAM-5090 4-port RS-232 Communication Module

Bi-direction Communication

The ADAM-5090 is equipped with four RS-232 ports, which makes it especially suitable for bi-direction communication. It can simultaneously read data from other third-party devices such as Bar Code and PLC as long as these devices are equipped with a RS-232 interface. Furthermore, the ADAM-5090 can issue commands to control other devices. It is fully integrated with the ADAM-5000, ADAM-5500 and ADAM-4000 series, and transmits data to each other through the RS-232 port. The whole integrated system is an intelligent stand-alone system and can connect and issue commands to control devices such as printers and PLCs in remote factory location.

The ADAM-5090 transmits and receives data by polling communication, and each port can receive up to 128 bytes in the FIFO. For continuous data longer than 128 bytes, please refer to Table 4.20 for Baud Rate setting to avoid data loss.

Baud Rate (bps)	115200	57600	38400	19200	9600	4800	2400
Polling interval (ms)	11.11	22.22	33.33	66.66	133.33	266.66	533.33

Table 4-22: Baud Rate setting reference table

Communication Backup Function

With the ADAM-5090 you can implement dual communication channels between your PC and the ADAM system. Even when one of the two communication channels is down, your system can still function through the alternative communication channel. This dual communication channels can be implemented by application software.

ADAM-5090 Module Diagram

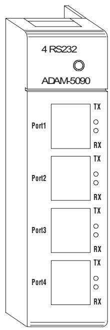


Figure 4-57: ADAM-5090 Module

ADAM-5090 Application Wiring

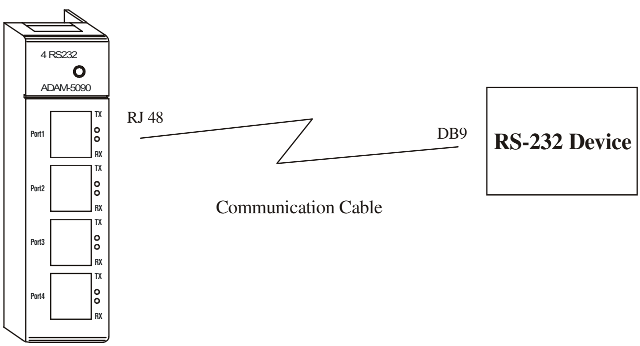


Figure 4-58: ADAM-5090 Application Wiring

PIN Mapping

PIN Name	RJ-48	DB9
/DCD	1	1
RX	2	2
TX	3	3
/DTR	4	4
GND	5	5
/DSR	6	6
/RTS	7	7
/CTS	8	8
RI or +5V	9	9
GND	10	X

Table 4-23: Pin Mapping

ADAM-5090 Technical Specification

Function	Provides communication ports for the ADAM-5510 to integrate other devices with communication function into your system
Electrical Interface	4 ports (RS-232)
Communication Rates	4800, 9600, 19200, 38400, 115200bps
FIFO	128 bytes/per UART (Tx/Rx)
Indicator	Tx (Orange), Rx (Green)
Power Required	100mA @ $5V_{DC}$ Default in RI mode (*)

Table 4-24: ADAM-5090 technical specifications

* User can define the communication ports with 5VDC output by switching the jumper, and the maximum current output is 400mA.

I/O Slots and I/O Ports Numbering

The ADAM-5090 module provides four RS-232 ports for communication with target devices. The ports are numbered 1 through 4. For programming, the definition of port number depends on the slot number and port number. For example, the second port on the ADAM-5090 in slot 1 is defined to port 12 (refer to table 6.1).

Jumper Settings

This section tells you how to set the jumpers to configure your ADAM-5090 module. There are four jumpers on the PC Board. User can choose RI signal or 5V output for each port by setting these jumpers (system default is RI signal).

The following figure shows the location of the jumpers:

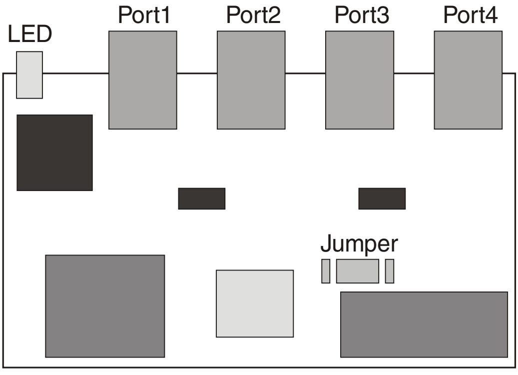


Figure 4-59: Jumper locations on the CPU card

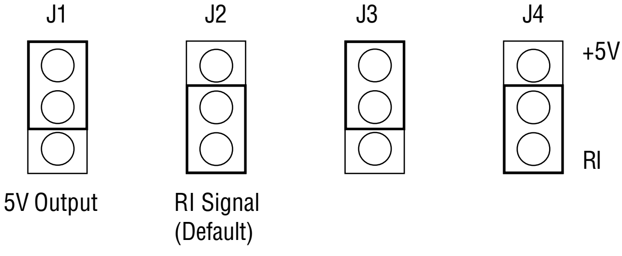


Figure 4-60: Jumper Settings

LED Status of the ADAM-5090 Module

There are two LEDs for each port on the front panel of the ADAM-5090 to display specific communication status:

a. Green LED (RX): Data Receiving Status; the LED indicator is on when the port is receiving data.

b. Orange LED (TX): Data Transmitting Status; the LED indicator is on when the port is transmitting data.

Configure Your ADAM-5090 Module

This section explains how to configure an ADAM-5090 module before implementing it into your application.

Quick Start

Step 1: Get your host PC ready, and run the ADAM-5510 Utility Software.

Step 2: Install the ADAM-5090 Module and power on your ADAM-5510 main unit.

Step 3: Download the executable program to the main unit

Step 4: Monitor the ADAM-5090 Module's current status from the PC through the utility software.

A basic example program for the ADAM-5090

```
main ()
{
  //Install the port you would like to use. Here we install slot 0, port 1.
  port_install(1);
  // Here we install slot 2, port 2.
  port_install(22);

  //Select working port. Here we select slot 0, port 1.
  port_select(1);
  //Set port data format.
  //Here we set the data format of port 1 as lengh:8; parity:0;stop_bit:1. (N81)
```

```
    port_set_format(1,8,0,1);

    //Set port speed. Here we set communication speed of port 1 as 115200 bps.
    //(L is necessary)
    port_set_speed(1,115200L);

    //Enable Port FIFO. Here we enable 128 byte FIFO for port1.
    port_enable_fifo(1);

    //After these above settings are enabled, you can apply any other
    function library to implement your program.
}
```

—A receive-and-transmit example program for the ADAM-5090

```
main()
{
    int err_value, char character
    port_installed(1)
    :
    :
    port_enable_fifo(1);

    //check whether error has been received or not
    err_value=port_rx_error(1);

    //if error detected, print out the message
    if(err_value)
    {
     printf("\n Rx Error, The LSR Value=%02X", Err_value)";
    }
    //check whether FIFO receives data or not; if data received, read a character
    if(port_rx_ready(1))
```

```
    {
    character=port_rx(1);
    }

    //check whether FIFO is empty or not, if empty, send a character
    if(port_tx_empty(1));
    {
    port_tx(1, character)
    }
}
```

Chapter 5
Function Library

5.1 Introduction

User-designed ADAM-5510M application programs make use of ADAM-5510M library functions. To make the most efficient use of ADAM-5510M's memory space, the ADAM-5510M function library has been separated into five smaller libraries. Therefore, a user can link only those libraries needed to run his application, and only those libraries will be included in the compiled executable. The smaller the linked libraries, the smaller the compiled executable will be.

Note 1: These function libraries support Borland Turbo C 3.0 for DOS only.

Note 2: Please includes all necessary ADAM-5510M function libraries in your project file.

5.1.1 Library Classification

ADAM-5510 has five function libraries, categorized according to usage:

Category A. System Functions: (UTILITY*.LIB)

Category B. Communication Functions: (COMM*.LIB)

Category C. Low Speed I/O Module Access Functions: (LIO*.LIB)

Category D. High Speed I/O Module Access Functions: (HIO*.LIB)

Category E. Counter/Frequency Module Access Functions: (LAI*.LIB)

5.2 Libraries Sized for Different Memory Modes

The ADAM-5510M function libraries support four memory modes:

SMALL, MEDIUM, COMPACT and LARGE. You can use library files sized according to your memory mode. For example, if you use ***small*** mode you can link UTILITYS.LIB and LIOS.LIB to implement system and low speed I/O module access functions. On the other hand, if you use ***large*** mode, you can link UTILITYL.LIB and LIOL.LIB.

5.3 Library Index

5.3.1 Category A. System Functions: (UTILITY.LIB)

5.3.2 Category B. Communication Functions: (COMM*.LIB)

checksum()

RS-485 Port (COM2) Functions

com_485_install()

com_485_deinstall()

com_485_set_format()

com_485_set_speed()

com_485_flush_rx()

com_485_flush_tx()

com_485_rx()

com_485_rx_empty()

com_485_tx()

com_485_tx_string()

com_485_tx_empty()

Program Port (COM3) Functions

com_pgm_install()

com_pgm_deinstall()

com_pgm_flush_rx()

com_pgm_flush_tx()

com_pgm_rx()

com_pgm_rx_empty()

com_pgm_set_format()

com_pgm_set_speed()

com_pgm_tx()

com_pgm_tx_empty()

com_pgm_tx_string()

RS-232 Port (COM1) Functions

com_install()

com_deinstall()

com_set_format()

com_set_parity()

com_set_speed()

com_rx()

com_tx()

com_rx_empty(), com_tx_empty()

com_tx_ready()

com_tx_string()

com_flush_rx(), com_flush_tx()

com_carrier()

com_clear_break(), com_set_break()

com_clear_local_loopback(),com_set_local_loopback()

com_disable_fifo(), com_enable_fifo()

com_get_line_status(), com_set_line_params(),

com_lower_dtr(), com_raise_dtr()

com_lower_rts(), com_raise_rts()

com_read_scratch_register(), com_write_scratch_register()

CRC16()

com_get_modem_status()

modem_autoanswer()

modem_command_state()

modem_command()

modem_dial()

modem_handup()

modem_initial()

RS-232/485 Port (COM4) Functions

com_232_485_install()

com_232_485_deinstall()

com_232_485_set_format()

com_232_485_set_speed()

com_232_485_flush_rx()

com_232_485_flush_tx()

com_232_485_rx()

com_232_485_rx_empty()

com_232_485_tx()

com_232_485_tx_string()

com_232_485_tx_empty()

5.3.3 Category C. Low Speed I/O Module Access Functions (LIO*.LIB)

AiUpdate()

Init5013()

Get5013()

GetRange5013()

Init501718()

Get501718()

GetRange501718()

5.3.4 Category D. High Speed I/O Module Access Functions (HIO*.LIB)

Init5017H()

GetRange5017H()

Get5017H()

Init5024()

Set5024()

Get5050()

Get5051()

Get5052()

Set5050()

Set5056()

Set5060()

Set5068()

5.3.5 Category E. Counter/Frequency Module Access Functions (LAI*.LIB)

Init5080()

Get5080()

Clear_Counter()

Start_Stop_Counter()

ReadOverflowFlag()

SetInitCounterVal()

5.4 Function Library Description

5.4.1 System Utility Library (UTILITY*.LIB)

ADAMdelay

Syntax:

void ADAMdelay (unsigned short msec)

Description:

Delays program operation by a specified number of milliseconds.

Parameter	Description
msec	From 0 to 65535.

Return value:

None.

Example:

```
#include "5510drv.h"
void main (void)
{
/* codes placed here by user */
ADAMdelay(1000); /* delay 1 sec. */
/* codes placed here by user */
}
```

Remarks:

None

Get_BoardID

Syntax:

unsigned char Get_BoardID (int Board)

Description:

Gets the type identification of the I/O module in a controller slot.

Parameter	**Description**
Int Board	The slot number of an ADAM-5510M, from 0 to 3.

Return value:

The return values are:

I/O Module name	Return Value
ADAM-5017	ADAM5017_ID
ADAM-5018	ADAM5018_ID
ADAM-5017H	ADAM5017H_ID
ADAM-5013	ADAM5013_ID
ADAM-5080	ADAM5080_ID
ADAM-5052	ADAM5052_ID
ADAM-5050	ADAM5050_ID
ADAM-5051	ADAM5051_ID
ADAM-5056	ADAM5056_ID
ADAM-5060	ADAM5060_ID
ADAM-5068	ADAM5068_ID
ADAM-5024	ADAM5024_ID

Remarks:

None

Get_NodeID

Syntax:

unsigned char Get_NodeID (void)

Description:

Gets the DIP switches number of the ADAM-5510M controller.

Parameter **Description**

None.

Return value:

The DIP switches number of the ADAM-5510M controller.

Example:

```
#include "5510drv.h"
unsigned char SystemNodeNumber;
unsigned char IOModuleName, SlotNumber;
void main(void)
{
  SystemNodeNumber = Get_NodeID();
  if( SystemNodeNumber == 0x15)
  {
          /* Read IO module name in Slot 0*/
          SlotNumber = 0;
          IOModuleName = Get_BoardID(SlotNumber);
          if( IOModuleName == ADAM5051_ID)
          {
          /* IO Board is current, put your code in Here*/
          }
          else
```

```
            {
                        printf("\nThe IO Board is NOT ADAM5051");
                        printf("\nPlease Check your system setup");
            }
  }
  else
            printf("\nNode number Error!");
}
```

Remarks :

None

GetRTCtime

SetRTCtime

Syntax:

unsigned char GetRTCtime(unsigned char Time)

void SetRTCtime(unsigned char Time,unsigned char data)

Description:

GetRTCtime: Reads Real-Time Clock chip timer. A user can activate a program on the date desired.

SetRTCtime: Sets date and time of the real-time clock.

Parameter	Description	
Time	RTC_sec	the second
	RTC_min	the minute
	RTC_hour	the hour
	RTC_day	the day
	RTC_week	day of the week
	RTC_month	the month
	RTC_year	the year
data	New contents.	

Return value:

The value requested by the user.

Example:

```
#include "5510drv.h"
void main(void)
{  unsigned char sec=0,min=0,hour=12;
   printf("Time  %02d:%02d:%02d \n",GetRTCtime(RTC_hour),
      GetRTCtime(RTC_min), GetRTCtime(RTC_sec));
```

```
    printf("Set current time 12:00:00\n");
    SetRTCtime(RTC_sec,sec);
    SetRTCtime(RTC_min,min);
    SetRTCtime(RTC_hour,hour);
    printf("Time  %02d:%02d:%02d \n",GetRTCtime(RTC_hour),
        GetRTCtime(RTC_min), GetRTCtime(RTC_sec));
}
```

Remarks:

None.

LED_init

LED_OFF

LED_ON

Syntax:

void LED_init(void)

void LED_OFF(int which_led)

void LED_ON(int which_led)

Description:

Turns LED lights on and off. The LED I/O port must be initialized first. It will take a little time for the light to stabilize following the signal for the turning on and turning off of the light.

Parameter	Description
which_led	PWR
	RUN
	COMM

Return value:

None.

Example:

```
#include "5510drv.h"
void main(void)
{
LED_init();
/* flash COMM led */
while(1)
{
```

```
LED_ON(COMM);
ADAMdelay(500);
LED_OFF(COMM);
}
}
```

Remarks:

None.

ProgramByte

ProgramSector

EraseSector

Syntax:

BOOL EraseSector(unsigned long ulBase)

BOOL ProgramByte(unsigned long ulAddress, BYTE byte)

BOOL ProgramSector(unsigned long ulAddress_s, unsigned char far * SECTOR_DATA)

Description:

EraseSector :	Erases a 64 KB sector of data in the 256 KB Flash memory
ProgramByte :	Programs a byte of information into the 256 KB Flash memory. This feature supports data-logging or mass information storage.
ProgramSector :	Programs an entire 32 KB sector of data of the global variable, SECTOR_DATA, into 256 KB Flash memory.

Parameter	**Description**
ulBase	User-determined address range to be erased, taken from addresses in the range 0x80000L to 0xB0000L.
ulAddress	User-determined destination address for byte transfer, taken from the range 0x80000L to 0xBFFFFL.
ulAddress_s	User-determined destination address in the Flash memory, taken from addresses in the range 0x80000L to 0xB8000L.
SECTOR_DATA	Pointer at the starting address in the origin memory of the user's data array.

Return value:

TRUE	Successful transfer to Flash memory.
FALSE	Error (destination already occupied, excess address range, or program error).

read_mem

Syntax:

unsigned char read_mem (int memory_segment , unsigned int i)

Description:

Reads far memory data, 256 KB Flash memory, from 0x80000L to 0xBFFFFL, where (the Absolute Address) = (SEG*16 + OFFSET). For example, (0x800FFL) = (0x8000*16 + 0x00FF).

Parameter	Description
memory_segment	User-determined address taken from the range 0x8000 to 0xBF00.
i	Offset for use in location of memory taken from the range 0x0000 to 0x0FFF.

Return value:

The value in memory storage at the indicated address.

Example:

```
#include “5510drv.h”
void main(void)
{
  unsigned char sector[32768];
  unsigned char data;
  unsigned long addr,sector_num;
  unsigned int i;

  printf(“erase sector 0x80000L\n”);
  if(EraseSector(0x80000L))
            printf(“erase succeed \n”);
  printf(“Write data(55) to 0x80000~0x80001\n”);
```

```
    data=55;
    ProgramByte(0x80000L,data);
    ProgramByte(0x80000L+1,data);
    ProgramByte(0x80000L+2,data);
    for(i=0;i<3;i++)
    {
            printf("read%d data=%d\n",i,read_mem(0x8000,0x0000+i));
    }
    printf("erase sector 0x80000L\n");
    if(EraseSector(0x80000L))
            printf("erase succeed \n");
    data = 1;
    for(i=0;i<32768;i++)
                    *(sector+i)=data;
    printf("Write data(0x01) to 0x80000~0x87FFF\n");
    ProgramSector(0x80000,&sector);
    for(i=0;i<100;i++)
    {
            printf("read%d data=%d\n",i,read_mem(0x8000,0x0000+i));
    }
}
```

Remarks:

None.

Get_SysMem

Set_SysMem

Syntax:

unsigned char Get_SysMem(unsigned char which_byte)

void Set_SysMem(unsigned char which_byte, unsigned char data)

Description:

Get_SysMem: Reads a byte from security SRAM.

Set_SysMem: Writes a byte to security SRAM. Security SRAM supports 113 bytes for user storage of important information.

Parameter	**Description**
which_byte	From 0 to 112, user-determined.
data	Value to be saved.

Return value:

The value in a byte of security SRAM.

Example:

```
#include "5510drv.h"
void main(void)
{
  unsigned char data[4] = {1,2,3,4};
  int i;
  /* save current value */
  for(i=10;i < 14;i++)
  {
        Set_SysMem(i, data[i-10]);
        printf("data=%d\n",Get_SysMem(i));
  }
}
```

Remarks:

None

Get_NVRAM_Size

Set_NVRAM_Size

Syntax:

unsigned char Get_NVRAM_Size(void)
void Set_NVRAM_Size(unsigned char sector)

Description:

Gets the size of battery backup RAM.
Sets the size of battery backup RAM.
(The unit is sectors, each sector is 4KB in size. Maximum size is 384 KB theoretically.)

Parameter	**Description**
sector	NVRAM size in 4 KB sectors, from 1 to 96 sectors.

Return value:

Get_NVRAM_Size: sector Number of sectors NVRAM size is set to, from 1 to 96.

Example:

```
#include "5510drv.h"
void main()
{
unsigned char sector;
sector = Get_NVRAM_Size();
printf("Backup ram=%dKbyte\n",sector*4);
/*Set Bacup ram 40Kbyte*/
Set_NVRAM_Size(10);
}
```

Remarks:

None.

write_backup_ram

read_backup_ram

Syntax:

void write_backup_ram(unsigned long index, BYTE data)

unsigned char read_backup_ram(unsigned long index)

Description:

Writes a byte to battery backup memory.

Reads the value in backup RAM at index address, maximum 384 KB total backup RAM, index = 0 - 393214;

Parameter	Description
index	An index for data in the battery backup RAM, from 0 to 393214; maximum 384 KB battery backup SRAM in total.
data	A byte of data that the programmer wants to write to battery-protected SRAM.

Return value:

The single-byte value in backup RAM at address index.

Example:

```
#include "5510drv.h"
void main()
{
   unsigned long addr;
   unsigned char data;
   /*write the data 0x55 into battery backup memory, index=10*/
   data=0x55;
   write_backup_ram(10,data);
   printf("data=%x\n",read_backup_ram(10));
}
```

Remarks:

None

Timer_Init()

Syntax:

int Timer_Init()

Description:

Initializes the timer built into the 80188 microprocessor. The return value "0" means the initialization of the time was successful. The return value "1" means the timer had already been initialized.

Parameter	**Description**
None.	

Return value:

0: Initialization was successful.

1: The timer had already been initialized.

Remarks:

None.

Timer_Reset

Syntax:

void Timer_Reset(int idx)

Description:

Resets the timer identified by the integer idx to its initial state.

Parameter	Description
idx	Timer index.

Return value:

None.

Remarks:

None.

Timer_Set

Syntax:

int Timer_Set(unsigned int msec)

Description:

Requests a timer function from the microprocessor and then sets the time interval of the function. Timer intervals are set in 5 millisecond increments. The function return value is an integer representing the ID of the timer function when it is successful.

A return value "-1" means the request failed. Programmers should consider whether an assigned timer has timed-out when programming for timer functions. The value of the variable tmArriveCnt[idx] can be checked to verify timer status.

A value of 0 indicates that the timer is still counting. Values other than 0 mean the timer has timed-out.

Parameter	Description
msec	Time interval set, max. value is 65536.

Return value:

Integer Function success, value represents function timer ID. Max. value of 100.

-1 Function failure.

Remarks:

Timer function calls in the ADAM-5510M are emulated as timer functions in a PLC. Applications using timer functions will run less efficiently the more timer functions are running simultaneously in a program. Please refer to Example 9 on the utility diskettes for details.

Release_All

Syntax:

void Release_All()

Description:

Releases all timer resources of the ADAM-5510M system.

Parameter Description

None.

Return value:

None.

Remarks:

None.

Example:

```
#include "5510drv.h"
void main()
{
  int idx;
  /*  Initializes the timer built into the 80188 microprocessor */
  Timer_Init();
  /*  Sets time interval of the timer to 1 second.          */
  idx=Timer_Set(1000);
  /*  Checks whether the timer has timed out            */
  while(tmArriveCnt[idx]==0)
  {
  /*  user can attend to other tasks...                  */
            printf("test");
```

```
    }

    /*  Resets the current timer to its initial state.          */
    Timer_Reset(idx);
    /*  Releases all timer resources                    */
    Release_All();
}
```

WDT_clear,WDT_disable,WDT_enable

Syntax:

```
void WDT_clear(void)
void WDT_disable(void)
void WDT_enable(void)
```

Description:

Clear watchdog timer.

Disable watchdog timer.

Enable watchdog timer.

When the watchdog timer is enabled, it will have to be cleared at least once every 1.5 seconds. The watchdog timer default value is “disable”.

Parameter	Description
None.	

Return value:

None.

Example:

```
#include "5510drv.h"
void main(void)
{
  int i;
  WDT_enable();
  for(i=0;i<100;i++)
  {
          /*put your code in Here*/
          WDT_clear();
          /*put your code in Here*/
```

```
    }
    WDT_disable();
}
```

Remarks:

None

5.4.2 Low speed I/O module access functions (LIO*.LIB)

AiUpdate

Syntax:

int AiUpdate(int Board, int *channel)

Description:

Checks whether the data of a low-speed analog input module, such as ADAM-5017, ADAM-5018 and ADAM-5013, is ready to be accessed.

Parameter	Description
int Board	The slot number of an ADAM-5510M, from 0 to 3.
int *channel	The return value indicates the channel for which data is ready.

Valid value 0 to 7 for ADAM-5017.

Valid value 0 to 6 for ADAM-5018.

Valid value 0 to 2 for ADAM-5013.

Return value:

int status; 0 : Ready

-1 : Not ready

-2 : The hardware of the module failed

Example:

Please refer to the ADAM-5017/5018 Example

Remarks:

None.

Get5013

Syntax:

void Get5013(int Board, int Channel, void *pValue)

Description:

Reads the data value in an ADAM-5013 module.

Parameter	**Description**
Board	0 - 3 for Slot0 ...Slot3.
Channel	0 - 2 for ADAM-5013.
*pValue	The value returned.

Note: The *pValue for ADAM-5013 must be interpreted in reference to the input range that was set during module configuration.

Return Value:

None.

Example:

Please refer to the ADAM-5013 Example

Remarks:

None.

Get501718

Syntax:

void Get501718(int Board, int Channel, void *pValue)

Description:

Reads the data value in an I/O module.

Parameter	**Description**
Board	0 - 3 for Slot0 ...Slot3.
Channel	0 - 6 for ADAM-5018; 0 - 7 for ADAM-5017
*pValue	The value returned.

Note: The *pValue for ADAM-5017 and ADAM-5018 must be interpreted in reference to the range input that was set during module configuration.

Return value:

None.

Example:

Please refer to the ADAM-5017/5018 Example

Remarks:

None.

GetRange5013

Syntax:

void GetRange5013 (int Board, int Channel, void *pRange)

Description:

Reads the input range in an ADAM-5013 module.

Parameter	Description
Board	0 - 3 for Slot0 ...Slot3.
Channel	0 - 2 for ADAM-5013.
*pRange	The input range code returned. (See Appendix C.)

Return Value:

None.

Example:

Please refer to the ADAM-5013 Example

Remarks:

None.

GetRange501718

Syntax:

void GetRange501718 (int Board, int Channel, void *pRange)

Description:

Reads the input range in an ADAM-501718 module.

Parameter	Description
Board	0 - 3 for Slot0 ...Slot3.
Channel	0 - 7 for ADAM-5017, 0-6 for ADAM-5018.
*pRange	The input range code returned (See Appendix C.)

Return Value:

*pRange The input range code returned.

Example:

Please refer to the ADAM-5017/5018 Example

Remarks:

None.

Init5013

Syntax:

void Init5013 (int Slot)

Description:

Initializes ADAM-5013. Note that ADAM-5013 must be initialized before other commands are issued to it.

Parameter	**Description**
Slot	From 0 to 3.

Return Value:

None.

Example:

Please refer to the ADAM-5013 Example

Remarks:

None.

Init501718

Syntax:

void Init501718(int Slot)

Description:

Initializes ADAM-5017 or ADAM-5018. Note that ADAM-5017 or ADAM-5018 must be initialized prior to other commands being issued to them.

Parameter	**Description**
Slot	From 0 to 3.

Return value:

None.

Example:

Please refer to the ADAM-5017/5018 Example

Remarks:

None.

ADAM-5013 Example

```
#include "5510drv.h"
void main()
{
   char ch;
   unsigned char Range;
   int *pRange,*pVaule;
   int i,j;
   int channel,slot;

   /*Initial ADAM-5013)*/
   /*One ADAM-5013 module on slot 2*/
   slot=2;
   Init5013(slot);
   GetRange5013(slot,0,pRange);

   Range=*pRange & 0xff;
   printf("range is 0x%x \n",Range);

   for(i=0;i<100;)
   {
     while(AiUpdate(slot, &channel)==0)
     {
       Get5013(slot,channel,pVaule);
       printf("\n channel= %d ADAM-5013=%04d \n",channel,*pVaule);
        i++;
     }
   }
   Release_All();
}
```

ADAM-5017/5018 Example

```
#include "5510drv.h"
void main()
{
  unsigned char Range,Format;
  int *pRange,*pVaule;
  int i;
  int channel,slot;

  char *RangeArray[6]={"+/-10V","+/-5V","+/-1V","+/-500mv","+/-
150mV","+/-20mv"};
  /*Initial ADAM-5017(ADAM-5018)*/
  /*One ADAM-5017 module on slot 0*/
  slot=0;
  Init501718(slot);
  GetRange501718(slot,0,pRange);

  Range=*pRange & 0xff;
  Format=(*pRange & 0xff00)>>8;
  printf("with range is %s format is 0x%x\n",RangeArray[Range-8],For-
mat);

  for(i=0;i<100;)
  {
    while(AiUpdate(slot, &channel)==0)
    {
     Get501718(slot,channel,pVaule);
                    printf("\n channel= %d ADAM-5017=%04d
mV\n",channel,*pVaule);
     i++;
    }
  }
}
```

5.4.3 High speed I/O module access functions (HIO*.LIB)

Get5017H

Syntax:

void Get5017H(int Board, int Channel, void *pValue)

Description:

Reads the data value in an ADAM-5017H module.

Parameter	Description
Board	0 - 3 for Slot0 ...Slot3.
Channel	0 - 7 for ADAM-5017H.
*pValue	The value returned.

Note: The pValue for ADAM-5017H must be interpreted in reference to the input range that be setup in the module configuration

Return Value:

None.

Example:

Please refer to the ADAM-5017H Example

Remarks:

None.

GetRange5017H

Syntax:

void GetRange5017H (int Board, int Channel, void *pRange)

Description:

Reads the input range in an ADAM-5017H module.

Parameter	**Description**
Board	0 - 3 for Slot0 ...Slot3.
Chanel	0 - 7 for ADAM-5017H.
*pRange	The input range code returned. (See Appendix C.)

Return Value:

None.

Example:

Please refer to the ADAM-5017H Example

Remarks:

None.

Init5017H

Syntax:

void Init5017 (int Slot)

Description:

Initializes ADAM-5017H. Note that ADAM-5017H must be initialized before other commands are issued to it.

Parameter	Description
Slot	From 0 to 3.

Return Value:

None.

Example:

Please refer to the ADAM-5017H Example

Remarks:

None.

ADAM-5017H Example

```
#include "5510drv.h"
void main()
{
  int channel,*pRange;
  int Format,Range;
  int slot;
  int *pValue[8];
  char *RangeArray[12]={"+/-10V","0~10V","+/-5V","0~5V",
              "+/-2.5v","0-2.5V","+/-1V","0-1V",
                  "+/-500mV","0~500mV","4~20mA","0~20mA"};
  slot=1;
  Init5017H(slot);
  channel=0;
  GetRange5017H(slot,channel,pRange);
  Format=(*pRange & 0xff00)>>8;
  Range=*pRange & 0xff;
  printf("\n(with range is %s format is 0x%x)",RangeArray[Range],Format);
 Init5017H(slot);
  for(channel=0;channel<8;channel++)
  {
        Get5017H(slot,channel,pValue+channel);
        printf("\n adam5017h channel:%d = %d",channel,*(pValue+channel));
  }
}
```

Init5024

Syntax:

void Init5024(int Slot, int ch0_val, int ch1_val, int ch2_val, int ch3_val)

Description:

Initializes ADAM-5024 module in the slot indicated, loading user-specified analog output values into each of the modules' four channels.

Parameter	**Description**
ch0_val	The initial value output by channel 0.
ch1_val	The initial value output by channel 1.
ch2_val	The initial value output by channel 2.
ch3_val	The initial value output by channel 3.

Return Value:

None.

Example:

Please refer to the ADAM-5024 Example

Remarks:

None.

Set5024

Syntax:

void Set5024(void *pValue, int Board, int Channel)

Description:

Specifies the output of a channel of a selected ADAM-5024.

Parameter	**Description**
*pValue	The value set for analog output.
Board	Slot number = 0 - 3.
Channel	AO channel = 0 - 3.

Return Value:

None.

Example:

Please refer to the ADAM-5024 Example

Remarks:

None.

ADAM-5024 Example

```
#include "5510drv.h"
void main()
{
   unsigned long *pValue;
   int channel,slot;
   slot=3;
   /*initializes outputs of all channels
   of the ADAM-5024 in slot 3 to output a
   value of 0 */
   Init5024(slot,0,0,0,0);
   /*Value set 2000mV*/
   *pValue=2000;
   for(channel=0;channel<4;channel++)
   {
    Set5024(pValue,slot,channel);
    printf("\n channel %d = %d  mV",channel,*pValue);
   }
}
```

Get5050, Get5051, Get5052

Syntax:

void Get5050(int Board, int Bit, int Size, void *pValue)
void Get5051(int Board, int Bit, int Size, void *pValue)
void Get5052(int Board, int Bit, int Size, void *pValue)

Description:

Reads the data value in an I/O module.

Parameter	**Description**
Board	ADAM-5510 slot number, from 0 to 3.
Bit	See "Size" parameter below.
Size	ABit, AByte, AWord
	If Size= ABit, Bit=0..15 (pin0..pin15)
	If Size=AByte, Bit=0 for Low Byte data; Bit=8 for High Byte data
	If Size=AWord, Bit does not care. Always word data.
pValue	The value returned.

Return value:

None.

Example:

```
void main(void)
{
unsigned char Bdata;
unsigned int Wdata;
/* Slot0, pin13, data=0 or 1 */
Get5051(0, 13, ABit, &Bdata);
/* Slot2, pin0~pin7, Bdata=Low Byte data */
```

```
Get5051(2, 0, AByte, &Bdata);
/* Slot3, pin0~pin15, Wdata=Word data */
Get5051(3, 0, AWord, &Wdata);
```

Remarks:

None.

Set5050, Set5056, Set5060, Set5068

Syntax:

```
void Set5050 (void *pValue, int Board, int Bit, int Size)
void Set5056 (void *pValue, int Board, int Bit, int Size)
void Set5060 (void *pValue, int Board, int Bit, int Size)
void Set5068 (void *pValue, int Board, int Bit, int Size)
```

Description:

Sets the digital output for ADAM-5050, ADAM-5056, ADAM-5060 and ADAM-5068 modules to the specified values.

Parameter	Description
pValue	The digital value specified by the user to be output.
Board	0 to 3 (Slot0 .. Slot3).
Bit	See "Size" parameter below.
Size	ABit, AByte, AWord If Size = ABit, Bit = 0 ...15 (pin0 ... pin15) If Size = AByte, Bit = 0 is Low Byte data Bit = 8 is High Byte data If Size = AWord, Bit does not care, always word data.

Return Value:

None.

Example:

```
void main(void)
{
unsigned char Bitdata = 1;
Set5056( &Bitdata, 0, 13, ABit);
/ * Output 1 to slot 0, pin 13 */
}
```

Remarks:

None

5.4.4 Counter/Frequency Module Access Functions (LIA*.LIB)

Init5080

Description:

Initiate ADAM-5080 Module

Syntax:

void Init5080 (int slotno)

Parameter	Description
slotno	The specific slot inserted with ADAM-5080 0-3 or slot0-slot3

Return Value:

None

Example:

Please refer to the ADAM-5080 Example

Get5080

Description:

Get Value from specific channel in ADAM-5080

Syntax:

void Get5080 (int slotno, int channel, long *pValue)

Parameter	Description
slotno	The specific slot inserted with ADAM-5080 0-3 or slot0-slot3
channel	The specific channel in ADAM-5080, 0-3
*pValue	The Value returned

Return Value:

The Value from the specific channel

Example:

Please refer to the ADAM-5080 Example

Clear_Counter

Description:

Reset the current counter value to its initial value

Syntax:

int Clear_Counter(int slotno, int channel)

Parameter	Description
slotno	The specific slot inserted with ADAM-5080 0-3 or slot0-slot3
channel	The specific channel in ADAM-5080, 0-3

Return Value:

None

Example:

Please refer to the ADAM-5080 Example

Start_Stop_Counter

Description:

Start or stop the specific counter

Syntax:

int Stop_Start_Counter(int slotno, int channel, StartOrStop)

Parameter	Description
slotno	The specific slot inserted with ADAM-5080, 0-3 or slot0-slot3
channel	The specific channel in ADAM-5080, 0-3
Start	1
Stop	0

Return Value:

None

Example:

Please refer to the ADAM-5080 Example

ReadOverflowFlag

Description:

Check if counter value reach max. count limit

Syntax:

void ReadOverflowFlag(int slotno, char *pValue)

Parameter	Description
slotno	The specific slot inserted with ADAM-5080, 0-3 or slot0-slot3
*pValue	The value returned

Return Value:

The overflow value returned

Example:

Please refer to the ADAM-5080 Example

SetInitCounterVal

Description:

Set initial counter value (between 0 to 4,294,967,295)

Syntax:

int SetInitCounterVal(int slotno, int channel, unsigned long Value)

Parameter	**Description**
slotno	The specific slot inserted with ADAM-5080, 0-3 or slot0-slot3
channel	The specific channel in ADAM-5080, 0-3

Return Value:

None

Example:

Please refer to the ADAM-5080 Example

ADAM-5080 Example

```
#include "5510drv.h"

void main()
{

   int slot=0;
      unsigned long int *data;
   int start=1;
   int channel;
   unsigned char *pOverFlag,i;
   slot=0;
   *pOverFlag=0;
   Init5080(slot);

  for(channel=0;channel<4;channel++)
   {
   if (Start_Stop_Counter(slot,channel,start)==0)
   {
               printf("Start channel %d  failed!!!\n",channel);
               exit(0);
   }
   Clear_Counter(slot,channel);
   SetInitCounterVal(slot,channel,10);
   }
   for(i=0;i<100;i++)
   {
               for(channel=0;channel<4;channel++)
               {
```

```
            Get5080(0,0,data);
            printf("channel=%d data=%d\n",channel,*data);
            ReadOverflowFlag(channel,pOverFlag);
            printf("Channel %d over_flag_value=%d
\n",channel,*pOverFlag);
        }
  }
}
```

5.4.5 Communication functions (COMM*.LIB)

checksum()

RS-485 Port (COM2) Functions

com_485_install()

com_485_deinstall()

com_485_set_format()

com_485_set_speed()

com_485_flush_rx()

com_485_flush_tx()

com_485_rx()

com_485_rx_empty()

com_485_tx()

com_485_tx_string()

com_485_tx_empty()

Programming Port (COM3) Functions

com_pgm_install()

com_pgm_deinstall()

com_pgm_flush_rx()

com_pgm_flush_tx()

com_pgm_rx()

com_pgm_rx_empty()

com_pgm_set_format()

com_pgm_set_speed()

com_pgm_tx()

com_pgm_tx_empty()

com_pgm_tx_string()

RS-232 Port (COM1) Functions

com_install()

com_deinstall()

com_set_format()

com_set_parity()

com_set_speed()

com_rx()

com_tx()

com_rx_empty(), com_tx_empty()

com_tx_ready()

com_tx_string()

com_flush_rx(), com_flush_tx()

com_carrier()

com_clear_break(), com_set_break()

com_clear_local_loopback(),com_set_local_loopback()

com_disable_fifo(), com_enable_fifo()

com_get_line_status(), com_set_line_params(),

com_lower_dtr(), com_raise_dtr()

com_lower_rts(), com_raise_rts()

com_read_scratch_register(), com_write_scratch_register()

CRC16()

com_get_modem_status()

modem_autoanswer()

modem_command_state()

modem_command()

modem_dial()

modem_handup()

modem_initial()

RS-232/485 Port (COM4) Functions

com_232_485_install()

com_232_485_deinstall()

com_232_485_set_format()

com_232_485_set_speed()

com_232_485_flush_rx()

com_232_485_flush_tx()

com_232_485_rx()

com_232_485_rx_empty()

com_232_485_tx()

com_232_485_tx_string()

com_232_485_tx_empty()

Example:

```
#include "5510drv.h"
void main()
{
  unsigned long speed=9600L;
   unsigned char data,ch;
   int status,com,sp;
  unsigned int i,j;
 while(1)
 {
  printf("\n com1: com232");
  printf("\n com2: com485");
  printf("\n com3: com232485");
    printf("\n Input the coummunction  at com :  ");
    scanf("%d",&com);
   printf("\n Select baud rate 9600L ");
   printf("\n [1] 9600L ");
   printf("\n [2] 19200L ");
   printf("\n [3] 38400L ");
   printf("\n [4] 57600L ");
   printf("\n [5]115200L ");
    printf("\n baudrate=");
    scanf("%d",&sp);

  //com2 port (RS-485) install
    switch(sp)
    {
      case 1:
        speed=9600L;break;
      case 2:
        speed=19200L;break;
```

```
    case 3:
     speed=38400L;break;
    case 4:
     speed=57600L;break;
    case 5:
     speed=115200L;break;
    default:
     speed=115200L;break;
  }
  switch(com)
  {
    case 1:
      status=com_install(1);
      if(status==0)
        printf("\n The allocation of com%d port is ok\n ",com);
                          else if (status==1)
                          {
                                 printf("\ncom%d port is already
installed\n ",com);
                          }
      else
      {
        printf("\n The allocation of com%d port is not ok\n ",com);
         printf("status=%d\n",status);
       exit(0);
      }

      //Format and Speed Setting
     com_set_format(8,0,1);
      com_set_speed(speed);
      //Transfer data
```

```
 printf("Stop trasnfer data  <ESC>");
for(i=0;i<1000;i++)
 {
  for(j=0;j<1000;j++)
   {
     com_tx_string("  com232 test  ");
    if(kbhit())
     {
       ch=getch();
     if(ch==0x1b){ i=1000;j=1000 ;}
     }
   }
 }
 //Receive data
printf("\n\n Please transfer data from server or <ESC> to exit\n");
while(1)
 {
  if(com_rx_empty()==0)
   {
     data=com_rx();
      printf("\n %c",data);
   }
   if(kbhit())
   {
      ch=getch();
    if(ch==0x1b){ break; }
   }
 }
 break;
case 2:
 status=com_485_install();
```

```
if(status==0)
  printf("\n The allocation of com%d port is ok\n ",com);
                                else if (status==1)
                                {
                                      printf("\ncom%d port is already installed\n ",com);
                                }
else
{
  printf("\n The allocation of com%d port is not ok\n ",com);
   printf("status=%d\n",status);
 exit(0);
}

//Format and Speed Setting
com_485_set_format(8,0,1);
com_485_set_speed(speed);
//Transfer data
printf("Stop trasnfer data  <ESC>");
for(i=0;i<1000;i++)
{
for(j=0;j<1000;j++)
  {
    com_485_tx_string("  com485  test  ");
    if(kbhit())
     {
        ch=getch();
     if(ch==0x1b){ i=1000;j=1000 ;}
     }
  }
}
```

```
//Receive data
printf("\nPlease transfer data from server or input<ESC> to exit\n");
 while(1)
 {
 if(com_485_rx_empty()==0)
   {
    data=com_485_rx();
     printf("\n %c",data);
   }
  if(kbhit())
   {
     ch=getch();
    if(ch==0x1b){ break; }
   }
 }
 break;
case 3:
 status=com_232_485_install();
 if(status==0)
   printf("\n The allocation of com%d port is ok\n ",com);
                         else if (status==1)
                         {
                                   printf("\ncom%d port is already
installed\n ",com);
                         }
 else
 {
   printf("\n The allocation of com%d port is not ok\n ",com);
    printf("status=%d\n",status);
  exit(0);
 }
```

```
 //Format and Speed Setting
com_232_485_set_format(8,0,1);
 com_232_485_set_speed(speed);
 //Transfer data
 printf("Stop trasnfer data  <ESC>");
for(i=0;i<1000;i++)
 {
 for(j=0;j<1000;j++)
   {
   com_232_485_tx_string("  com232485 test  ");
    if(kbhit())
     {
       ch=getch();
     if(ch==0x1b){ i=1000;j=1000 ;}
     }
   }
 }
 //Receive data
printf("\n\n Please transfer data from server or <ESC> to exit\n");
while(1)
 {
 if(com_232_485_rx_empty()==0)
   {
   data=com_232_485_rx();
     printf("\n %c",data);
   }
  if(kbhit())
   {
     ch=getch();
   if(ch==0x1b){ break; }
   }
```

```
        }
        break;
      default:
        break;
    }
    printf("\n <ESC> to exit or anykey to continue\n ");
            ch=getch();
            if(ch==0x1b){ break;}
  }
}
```

Appendix A
COM Port Register Structure

This appendix gives a short description of each module's registers. For more information, please refer to the STARTECH 16C550 UART chip data book.

All registers are one byte. Bit 0 is the least significant bit, and bit 7 is the most significant bit. The address of each register is specified as an offset from the port base address (BASE), COM1 is 3F8h and COM2 is 2F8h.

DLAB is the "Divisor Latch Access Bit", bit 7 of BASE+3.

BASE+0 Receiver buffer register when DLAB=0 and the operation is a read.

BASE+0 Transmitter holding register when DLAB=0 and the operation is write.

BASE+0 Divisor latch bits 0 - 7 when DLAB=1

BASE+1 Divisor latch bits 8-15 when DLAB=1.

Bytes BASE+0 and BASE+1 together form a 16-bit number, the divisor, which determines the baud rate. Set the divisor as follows:

Baud rate	Divisor	Baud rate	Divisor
50	2304	2400	48
75	1536	3600	32
110	1047	4800	24
133.5	857	7200	16
150	768	9600	12
300	384	19200	6
600	192	38400	3
1200	96	56000	2
1800	64	115200	1
2000	58	x	x

BASE+1 Interrupt Status Register (ISR) when DLAB=0
bit 0: Enable received-data-available interrupt
bit 1: Enable transmitter-holding-register-empty interrupt
bit 2: Enable receiver-line-status interrupt
bit 3: Enable modem-status interrupt

BASE+2 FIFO Control Register (FCR)
bit 0: Enable transmit and receive FIFOs
bit 1: Clear contents of receive FIFO
bit 2: Clear contents of transmit FIFO
bits 6-7: Set trigger level for receiver FIFO interrupt

Bit 7	Bit 6	FIFO trigger level
0	0	01
0	1	04
1	0	08
1	1	14

BASE+3 Line Control Register (LCR)
bit 0: Word length select bit 0
bit 1: Word length select bit 1

Bit 1	Bit 0	Word length (bits)
0	0	5
0	1	6
1	0	7
1	1	8

bit 2: Number of stop bits
bit 3: Parity enable
bit 4: Even parity select
bit 5: Stick parity
bit 6: Set break
bit 7: Divisor Latch Access Bit (DLAB)

BASE+4	Modem Control Register (MCR)
	bit 0: DTR
	bit 1: RTS
BASE+5	Line Status Register (LSR)
	bit 0: Receiver data ready
	bit 1: Overrun error
	bit 2: Parity error
	bit 3: Framing error
	bit 4: Break interrupt
	bit 5: Transmitter holding register empty
	bit 6: Transmitter shift register empty
	bit 7: At least one parity error, framing error or break indication in the FIFO
BASE+6	Modem Status Register (MSR)
	bit 0: Delta CTS
	bit 1: Delta DSR
	bit 2: Trailing edge ring indicator
	bit 3: Delta received line signal detect
	bit 4: CTS
	bit 5: DSR
	bit 6: RI
	bit 7: Received line signal detect
BASE+7	Temporary data register

Appendix B
Data Formats and I/O Ranges

B.1 Analog Input Formats

The ADAM analog input modules can be configured to transmit data to the host in Engineering Units.

Engineering Units

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

This format presents data in natural units, such as degrees, volts, millivolts, and milliamps. The Engineering Units format is readily parsed by the majority of computer languages because the total data string length, including sign, digits and decimal point, does not exceed seven characters.

The data format is a plus (+) or minus (-) sign, followed by five decimal digits and a decimal point. The input range which is employed determines the resolution, or the number of decimal places used, as illustrated in the following table:

Input Range	Resolution
±15 mV, ±50 mV	1 µV (three decimal places)
±100 mV, ±150 mV, ±500 mV	10 µV (two decimal places)
±1 V, ±2.5 V, ±5 V	100 µV (four decimal places)
±10 V	1 mV (three decimal places)
±20 mA	1 µA (three decimal places)
Type J and T thermocouple	0.01ºC (two decimal places)
Type K, E, R, S, and B thermocouple	0.1ºC (one decimal place)

Example 1

The input value is -2.65 V and the corresponding analog input module is configured for a range of ±5 V. The response to the Analog Data In command is:

-2.6500(cr)

Example 2

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

+305.50(cr)

Example 3

The input value is +5.653 V. The analog input module is configured for a range of ±5 V range. When the engineering units format is used, the ADAM Series analog input modules are configured so that they automatically provide an over range capability. The response to the Analog Data In command in this case is:

+5.6530(cr)

B.2 Analog Input Ranges - ADAM-5017

Module	Range Code	Input Range Description	Data Formats	+F.S.	Zero	-F.S.	Displayed Resolution	Actual Value
ADAM-5017	08h	±10 V	Engineering Units	+10.000	±00.000	-10.000	1 mV	Reading/1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	09h	±5 V	Engineering Units	+5.0000	±0.0000	-5.0000	100.00 µV	Reading/1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	0Ah	±1 V	Engineering Units	+1.0000	±0.0000	-1.0000	100.00 µV	Reading/10000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	0Bh	±500 mV	Engineering Units	+500.00	±000.00	-500.00	10 µV	Reading/10
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	0Ch	±150 mV	Engineering Units	+150.00	±000.00	-150.00	10 µV	Reading/100
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	0Dh	±20 mA	Engineering Units	+20.000	±00.000	-20.000	1 µV	Reading/1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	

B.3 Analog Input Ranges - ADAM-5018

Module	Range Code	Input Range Description	Data Formats	+F.S.	Zero	-F.S.	Displayed Resolution	Actual Value
ADAM-5018	00h	±15 mV	Engineering Units	+15.000	±00.000	-15.000	1 µV	Reading/ 1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	01h	±50 mV	Engineering Units	+50.000	±00.000	-50.000	1 µV	Reading/ 100
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	02h	±100 mV	Engineering Units	+100.00	±000.00	-100.00	10 µV	Reading/ 100
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	03h	±500 mV	Engineering Units	+500.00	±000.00	-500.00	10 µV	Reading/ 10
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	04h	±1 V	Engineering Units	+1.0000	±0.0000	-1.0000	100 µV	Reading/ 10000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	05h	±2.5 V	Engineering Units	+2.5000	±0.0000	-2.5000	100 µV	Reading/ 10000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	06h	±20 mA	Engineering Units	+20.000	±00.000	-20.000	1 µA	Reading/ 1000
			% of FSR	+100.00	±000.00	-100.00	0.01%	
			Two's Complement	7FFF	0000	8000	1 LSB	
	07h	Not Used						

Module	Range Code	Input Range Description	Data Formats	Maximum Specified Signal	Minimum Specified Signal	Displayed Resolution	Actual Value
ADAM-5018	0Eh	Type J Thermocouple 0° C to 760° C	Engineering Units	+760.00	+000.00	0.1° C	Reading/ 10
			% of FSR	+100.00	+000.00	0.01%	
			Two's Complement	7FFF	0000	1 LSB	
	0Fh	Type K Thermocouple 0° C to 1370° C	Engineering Units	+1370.0	+0000.0	0.1° C	Reading/ 10
			% of FSR	+100.00	+000.00	0.01%	
			Two's Complement	7FFF	0000	1 LSB	
	10h	Type T Thermocouple -100° C to 400° C	Engineering Units	+400.00	-100.00	0.1° C	Reading/ 10
			% of FSR	+100.00	-025.00	0.01%	
			Two's Complement	7FFF	E000	1 LSB	
	11h	Type E Thermocouple 0° C to 1000° C	Engineering Units	+1000.00	+0000.0	0.1° C	Reading/ 10
			% of FSR	+100.00	±000.00	0.01%	
			Two's Complement	7FFF	0000	1 LSB	
	12h	Type R Thermocouple 500° C to 1750° C	Engineering Units	+1750.0	+0500.0	0.1° C	Reading/ 10
			% of FSR	+100.00	+028.57	0.01%	
			Two's Complement	7FFF	2492	1 LSB	
	13h	Type S Thermocouple 500° C to 1750° C	Engineering Units	+1750.0	+0500.00	0.1° C	Reading/ 10
			% of FSR	+100.00	+028.57	0.01%	
			Two's Complement	7FFF	2492	1 LSB	
	14h	Type B Thermocouple 500° C to 1800° C	Engineering Units	+1800.0	+0500.0	0.1° C	Reading/ 10
			% of FSR	+100.00	+027.77	0.01%	
			Two's Complement	7FFF	2381	1 LSB	

B.4 Analog Input Ranges - ADAM-5017H

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

Note: The full scale values in this table are theoretical values for your reference; actual values will vary.

B.5 Analog Output Formats

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

Engineering Units

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

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

Example:

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

B.6 Analog Output Ranges

Range Code	Output Range Description	Data Formats	Maximum Specified Signal	Minimum Specified Signal	Displayed Resolution
30	0 to 20 mA	Engineering Units	20.000	00.000	5 µA
		% of Span	+100.00	+000.00	5 µA
		Hexadecimal Binary	FFF	000	5 µA
31	4 to 20 mA	Engineering Units	20.000	04.000	5 µA
		% of Span	+100.00	+000.00	5 µA
		Hexadecimal Binary	FFF	000	5 µA
32	0 to 10 V	Engineering Units	10.000	00.000	2.442 mV
		% of Span	+100.00	+000.00	2.442 mV
		Hexadecimal Binary	FFF	000	2.442 mV

B.7 ADAM-5013 RTD Input Format and Ranges

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

Note: See next page for table continuation.

Note: This table continued from previous page.

27	100 Ohms Platinum RTD 0 to 600° C a=0.00392	Engineering Units	+600.00	+000.00	±0.6° C
28	120 Ohms Nickel RTD -80 to 100° C	Engineering Units	+100.00	-80.00	±0.1° C
29	120 Ohms Nickel RTD 0 to 100° C	Engineering Units	+100.00	+000.00	±0.1° C

ADAM 5000 AI/AO Scaling

Module	Type	Range Low	Range High	Scale Low	Scale High	Data Format
5013RTD	385(IEC)	-100	100	0	65535	U16B
		0	100	0	65535	U16B
		0	200	0	65535	U16B
		0	600	0	65535	U16B
	395(JIS)	-100	100	0	65535	U16B
		0	100	0	65535	U16B
		0	200	0	65535	U16B
		0	600	0	65535	U16B
	Ni	-80	100	0	65535	U16B
		0	100	0	65535	U16B
5017AI	mV	-150	150	0	65535	U16B
	mV	-500	500	0	65535	U16B
	V	-1	1	0	65535	U16B
	V	-5	5	0	65535	U16B
	V	-10	10	0	65535	U16B
	mA	-20	20	0	65535	U16B
5017H AI	mV	-500	500	0	4095	U12B
	mV	0	500	0	4095	U12B
	V	-10	10	0	4095	U12B
	V	0	10	0	4095	U12B
	V	-5	5	0	4095	U12B
	V	0	5	0	4095	U12B
	V	-2.5	2.5	0	4095	U12B
	V	0	2.5	0	4095	U12B
	V	-1	1	0	4095	U12B
	V	0	1	0	4095	U12B
	mA	4	20	0	4095	U12B
	mA	0	20	0	4095	U12B
5018 AI	mV	-15	15	0	65535	U16B
	mV	-50	50	0	65535	U16B
	mV	-100	100	0	65535	U16B
	mV	-500	500	0	65535	U16B
	V	-1	1	0	65535	U16B
	V	-2.5	2.5	0	65535	U16B
	mA	-20	20	0	65535	U16B
	T/C(J)	0	760	0	65535	U16B
	T/C(K)	0	1370	0	65535	U16B
	T/C(T)	-100	400	0	65535	U16B
	T/C(E)	0	1000	0	65535	U16B
	T/C(R)	500	1750	0	65535	U16B
	T/C(S)	500	1750	0	65535	U16B
	T/C(B)	500	1800	0	65535	U16B
5024 AO	V	0	10	0	4095	U12B
	mA	4	20	0	4095	U12B
	mA	0	20	0	4095	U12B

This page is blank

Appendix C
Examples on CD

Three examples are included on the ADAM-5511 CD. After you install the utility CD on your host PC, these examples will be located in the directory C:\ADAM5511\Example. The following list describes these examples.

Example 1 (Ex1.prj)

This example scans all slots in an ADAM-5511 and then shows the status of any I/O modules(include AI/O, DI/O, Counter, and Series Communication Module) located in the slots.

Example 2 (Ex2.prj)

This is a modem test example which includes dial, hang-up, auto-answer and set break.

Example 3 (Ex3.prj)

Using ADAM-5511 COM port and ADAM-4520 (RS-232 to RS-422/485 converter) to scan ADAM-4000 series module as remote I/O function.

Appendix D
RS-485 Network

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

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

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

ADAM-5510/P31 systems are fully isolated and use just a single set of twisted pair wires to send and receive! Since the nodes are connected in parallel they can be freely disconnected from the host without affecting the functioning of the remaining nodes. An industry standard, shielded twisted pair is preferable due to the high noise ratio of the environment.

When nodes communicate through the network, no sending conflicts can occur since a simple command/response sequence is used. There is always one initiator (with no address) and many slaves (with addresses). In this case, the master is a personal computer that is connected with its serial, RS-232, port to an ADAM RS-232/RS-485 converter. The slaves are the ADAM-5510/P31 systems. When systems are not transmitting data, they are in listen mode. The host computer initiates a command/response sequence with one of the systems. Commands normally contain the address of the module the host wants to communicate with. The system with the matching address carries out the command and sends its response to the host.

D.1 Basic Network Layout

Multi-drop RS-485 implies that there are two main wires in a segment. The connected systems tap from these two lines with so called drop cables. Thus all connections are parallel and connecting or disconnecting of a node doesn't affect the network as a whole. Since ADAM-5510/P31 systems use the RS-485 standard, they can connect and communicate with the host PC. The basic layouts that can be used for an RS-485 network are:

Daisychain

The last module of a segment is a repeater. It is directly connected to the main-wires thereby ending the first segment and starting the next segment. Up to 32 addressable systems can be daisychained . This limitation is a physical one. When using more systems per segment the IC driver current rapidly decreases, causing communication errors. In total, the network can hold up to 64 addressable systems. The limitation on this number is the two-character hexadecimal address code that can address 64 combinations. The ADAM converter, ADAM repeaters and the host computer are non addressable units and therefore are not included in these numbers.

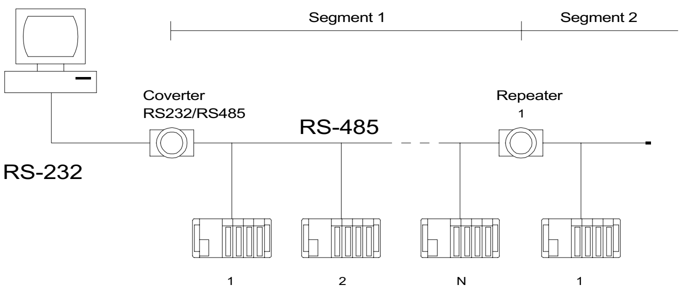


Figure D-1: Daisychaining

Star Layout

In this scheme the repeaters are connected to drop-down cables from the main wires of the first segment. A tree structure is the result. This scheme is not recommended when using long lines since it will cause a serious amount of signal distortion due to signal reflections in several line-endings.

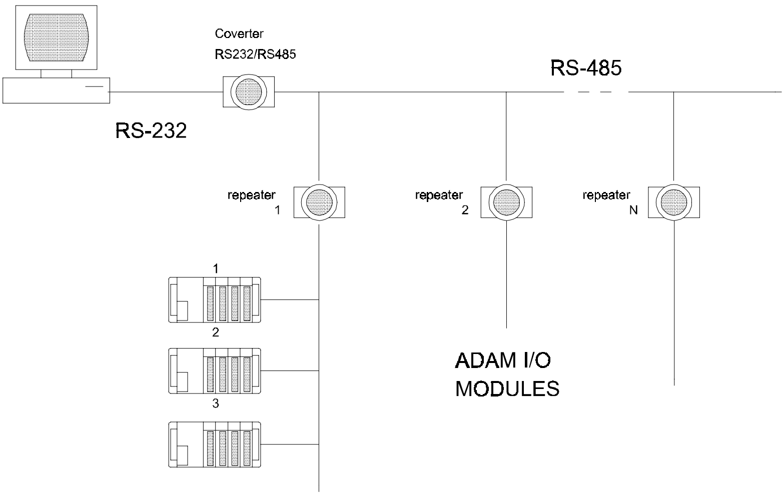


Figure D-2: Star structure

Random

This is a combination of daisychain and hierarchical structure.

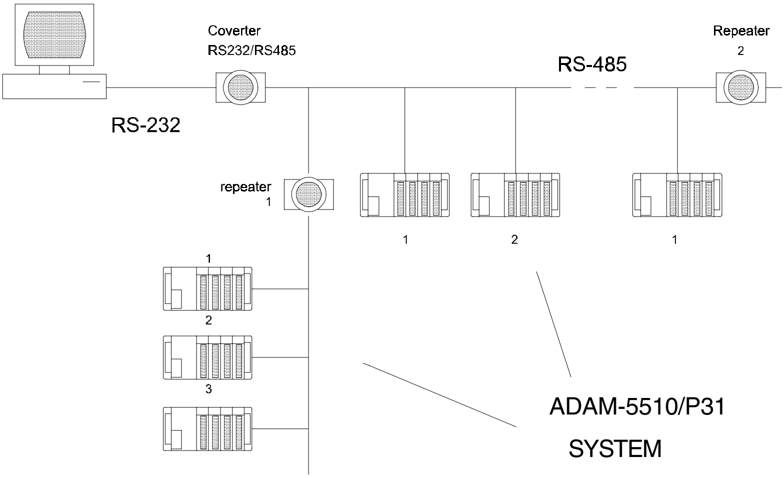


Figure D-3: Random structure

D.2 Line Termination

Each discontinuity in impedance causes reflections and distortion. When a impedance discontinuity occurs in the transmission line the immediate effect is signal reflection. This will lead to signal distortion. Specially at line ends this mismatch causes problems. To eliminate this discontinuity, terminate the line with a resistor.

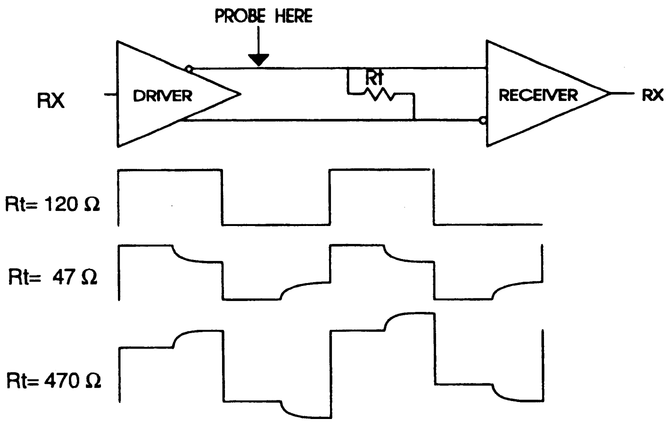


Figure D-4: Signal distortion

The value of the resistor should be a close as possible to the characteristic impedance of the line. Although receiver devices add some resistance to the whole of the transmission line, normally it is sufficient to the resistor impedance should equal the characteristic impedance of the line.

Example:

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

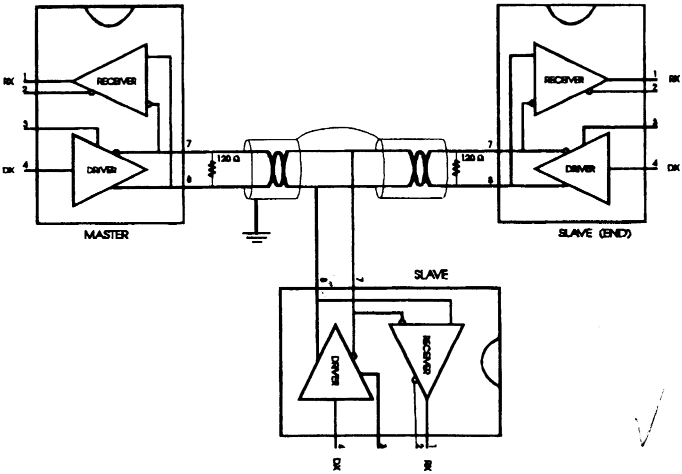


Figure D-5: Termination resistor locations

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

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

$R_P = 36 \times 10^3/30 = 1200\,\Omega$

While the termination receptor R_T will equal:

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

Thus for a line with a characteristic impedance of 100 W resistor

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

Since this value lies within 10% of the line characteristic impedance.

Thus as already stated above the line termination resistor R_T will normally equal the characteristic impedance Z_o.

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

Note: The recommend method wiring method, that causes a minimum amount of reflection, is daisy chaining where all receivers tapped from one transmission line needs only to be terminated twice.

D.3 RS-485 Data Flow Control

The RS-485 standard uses a single pair of wires to send and receive data. This line sharing requires some method to control the direction of the data flow. RTS (Request To Send) and CTS (Clear To Send) are the most commonly used methods.

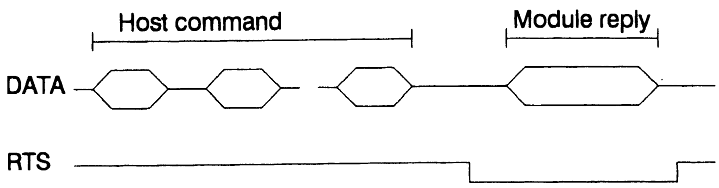


Figure D-6: RS-485 data flow control with RTS

Intelligent RS-485 Control

ADAM-4510 and ADAM-4520 are both equipped with an I/O circuit which can automatically sense the direction of the data flow. No handshaking with the host (like RTS, Request to Send) is necessary to receive data and forward it in the correct direction. You can use any software written for half-duplex RS-232 with an ADAM network without modification. The RS-485 control is completely transparent to the user.

Appendix E
Grounding Reference

Field Grounding and Shielding Application

Overview

Unfortunately, it's impossible to finish a system integration task at one time. We always meet some trouble in the field. A communication network or system isn't stable, induced noise or equipment is damaged or there are storms. However, the most usual issue is just simply improper wiring, ie, grounding and shielding. You know the 80/20 rule in our life: we spend 20% time for 80% work, but 80% time for the last 20% of the work. So is it with system integration: we pay 20% for Wire / Cable and 0% for Equipment. However, 80% of reliability depends on Grounding and Shielding. In other words, we need to invest more in that 20% and work on these two issues to make a highly reliable system.

This application note brings you some concepts about field grounding and shielding. These topics will be illustrated in the following pages.

1. Grounding
   - 1.1 The 'Earth' for reference
   - 1.2 The 'Frame Ground' and 'Grounding Bar'
   - 1.3 Normal Mode and Common Mode
   - 1.4 Wire impedance
   - 1.5 Single Point Grounding
2. Shielding
   - 2.1 Cable Shield
   - 2.2 System Shielding
3. Noise Reduction Techniques
4. Check Point List

E.1 Grounding

E.1-1 The 'Earth' for reference

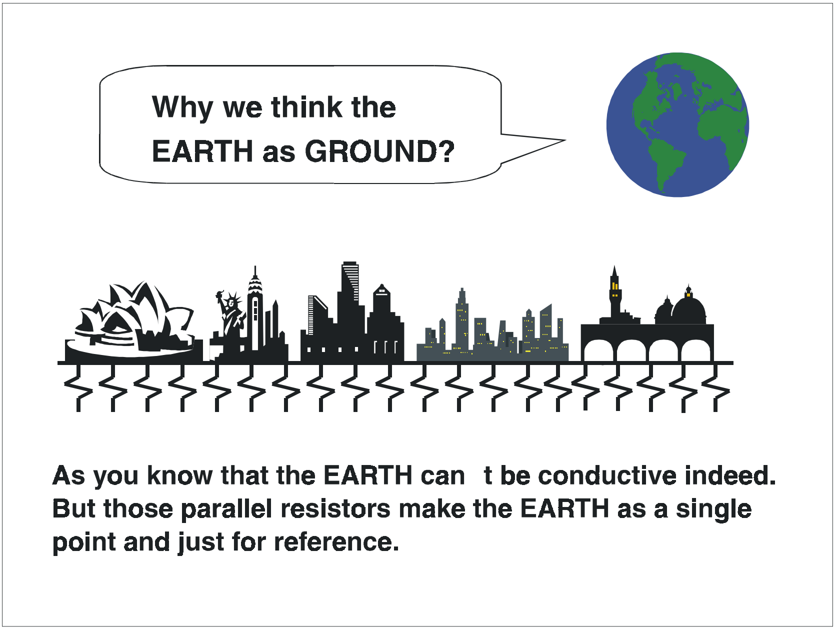
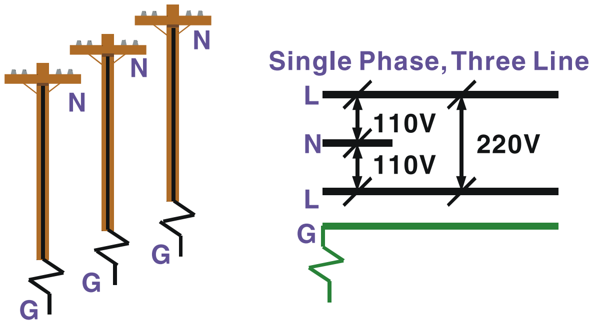


Figure E-1: Think the EARTH as GROUND.

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

E.1-2 The ‘Frame Ground’ and ‘Grounding Bar’



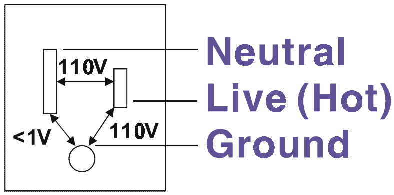
Neutral is the physical cable from Generator.
Ground is the local physical cable that connected to Ground Bar.

Figure E-2: Grounding Bar.

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

E.1-3 Normal Mode and Common Mode

Normal Mode & Common Mode



Normal Mode: refers to defects occurring between the live and neutral conductors. Normal mode is sometimes abbreviated as NM, or L-N for live-to-neutral.
Common Mode: refers to defects occurring between either conductor and ground. It is sometimes abbreviated as CM, or N-G for neutral-to-ground.

Figure E-3: Normal mode and Common mode.

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

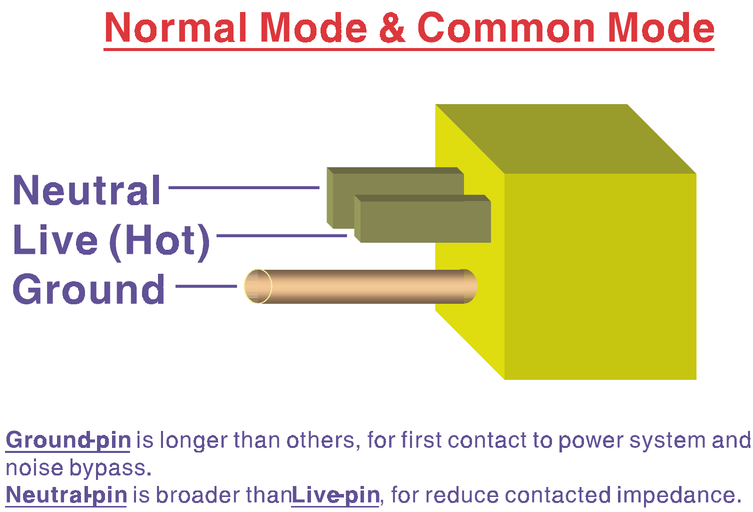


Figure E-4: Normal mode and Common mode.

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

E.1-4 Wire impedance

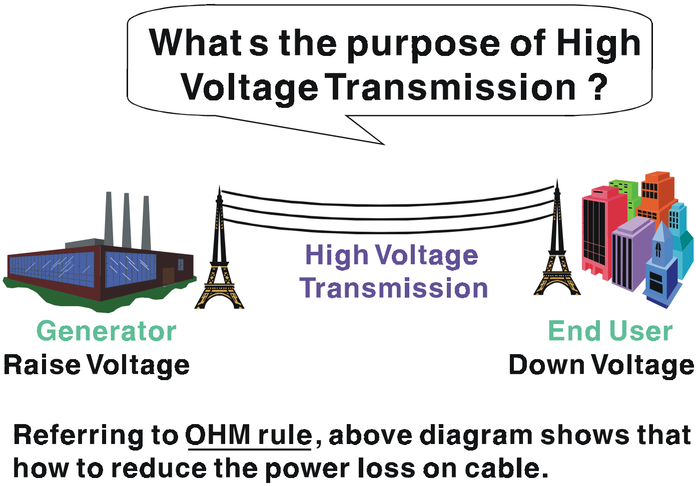


Figure E-5: The purpose of high voltage transmission

- What’s the purpose of high voltage transmission?

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

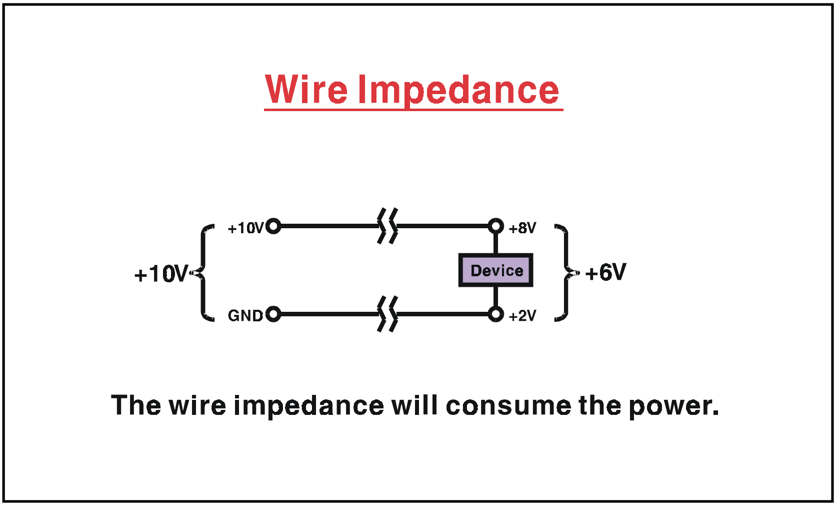


Figure E-6: wire impedance.

E.1-5 Single Point Grounding

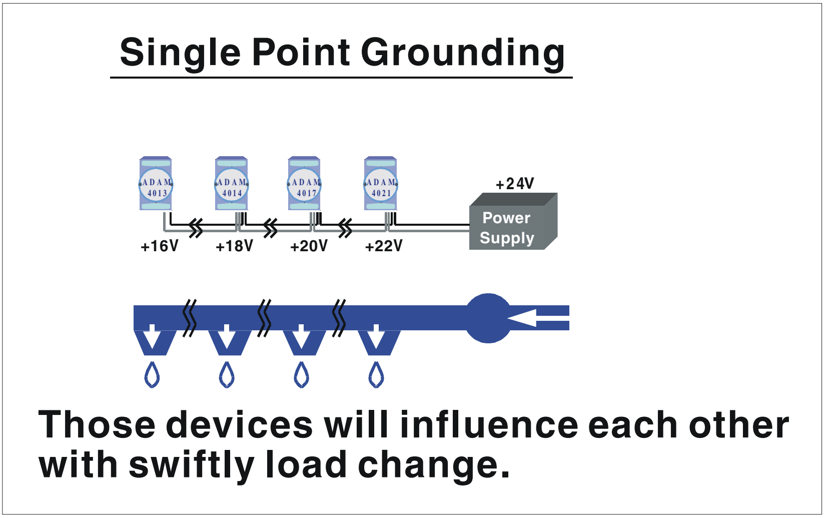


Figure E-7: Single point grounding. (1)

- What's Single Point Grounding?

Maybe you have had an unpleasant experience while taking a hot shower in Winter. Someone turns on a hot water faucet somewhere else. You will be impressed with the cold water!

The bottom diagram above shows an example of how devices will influence each other with swift load change. For example, normally we turn on all the four hydrants for testing. When you close the hydrant 3 and hydrant 4, the other two hydrants will get more flow. In other words, the hydrant cannot keep a constant flow rate.

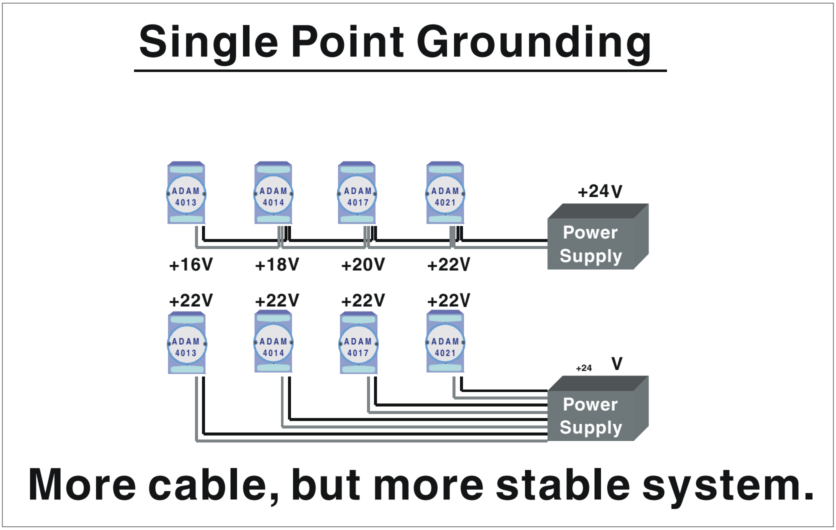


Figure E-8: Single point grounding. (2)

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

E.2 Shielding

E.2-1 Cable Shield

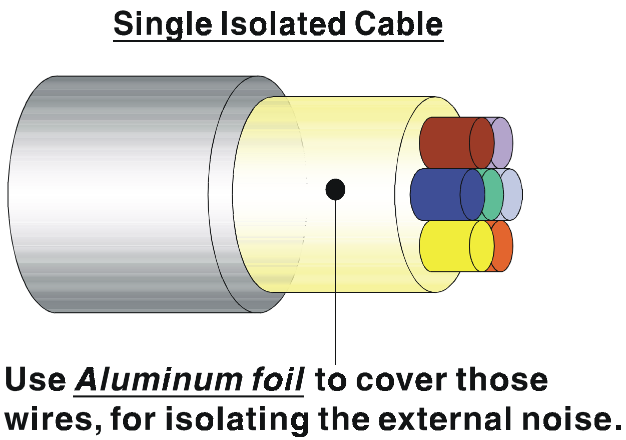


Figure E-9: Single isolated cable

- Single isolated cable

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

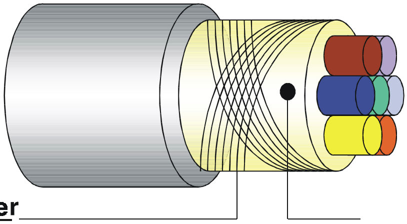


Figure E-10: Double isolated cable

- Double isolated cable

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

Additionally, follow these tips just for your reference.

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

E.2-2 System Shielding

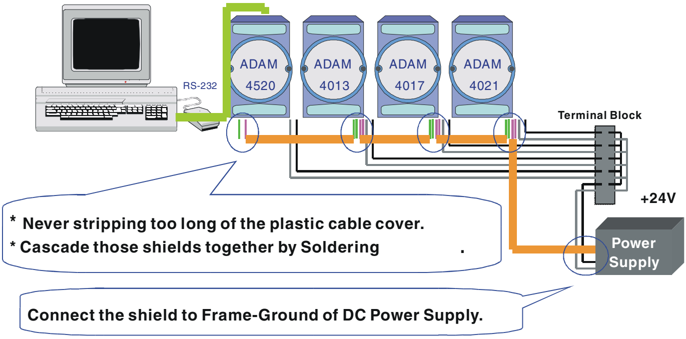


Figure E-11: System Shielding

- Never stripping too much of the plastic cable cover. This is improper and can destroy the characteristics of the Shielded-Twisted-Pair cable. Besides, the bare wire shield easily conducts the noise.
- Cascade these shields together by soldering. Please refer to following page for further detailed explanation.
- Connect the shield to Frame Ground of DC power supply to force the conducted noise to flow to the frame ground of the DC power supply. (The 'frame ground' of the DC power supply should be connected to the system ground)

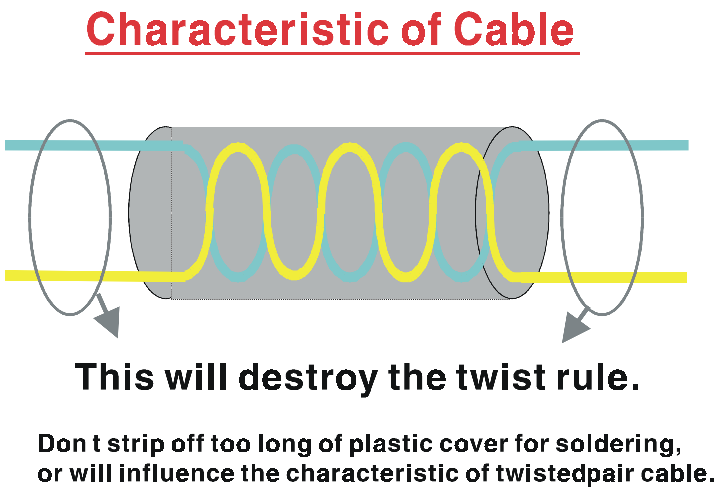


Figure E-12: The characteristic of the cable

- The characteristic of the cable

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

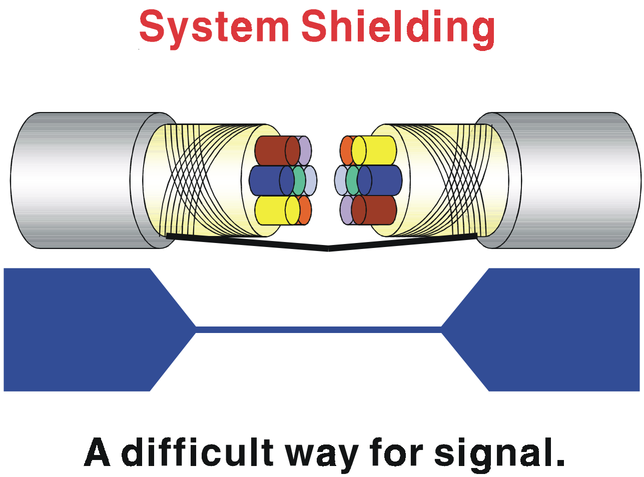


Figure E-13: System Shielding (1)

- Shield connection (1)

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

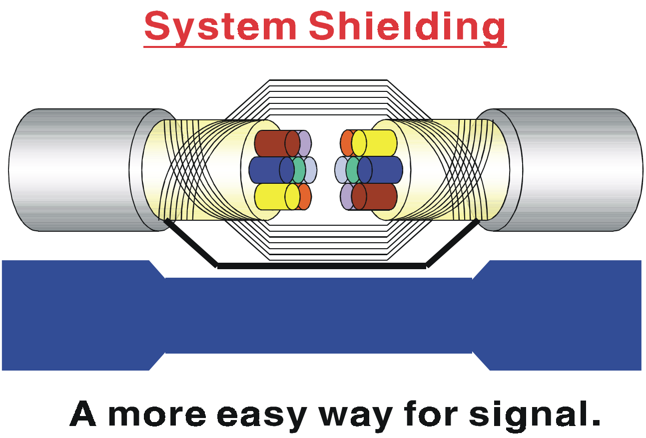


Figure E-14: System Shielding (2)

- Shield connection (2)

The previous diagram shows you that the fill soldering just makes an easier way for the signal.

E.3 Noise Reduction Techniques

- Isolate noise sources in shielded enclosures.
- Place sensitive equipment in shielded enclosure and away from computer equipment.
- Use separate grounds between noise sources and signals.
- Keep ground/signal leads as short as possible.
- Use Twisted and Shielded signal leads.
- Ground shields on one end ONLY while the reference grounds are not the same.
- Check for stability in communication lines.
- Add another Grounding Bar if necessary.
- The diameter of power cable must be over 2.0 mm^2.
- Independent grounding is needed for A/I, A/O, and communication network while using a jumper box.
- Use noise reduction filters if necessary. (TVS, etc)
- You can also refer to FIPS 94 Standard. FIPS 94 recommends that the computer system should be placed closer to its power source to eliminate load-induced common mode noise.

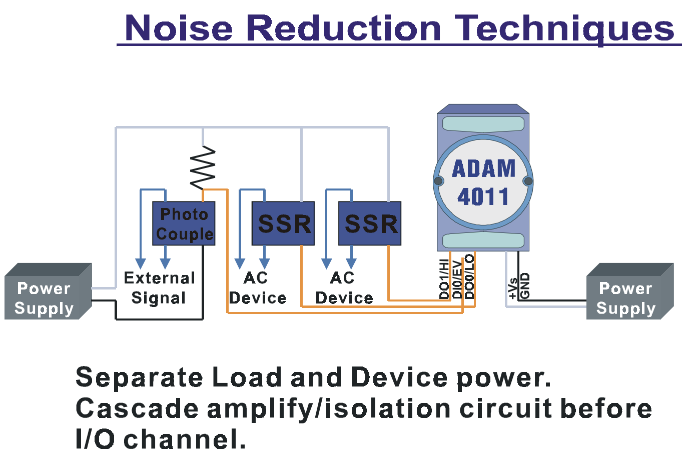


Figure E-15: Noise Reduction Techniques

E.4 Check Point List

- Follow the single point grounding rule?
- Normal mode and common mode voltage?
- Separate the DC and AC ground?
- Reject the noise factor?
- The shield is connected correctly?
- Wire size is correct?
- Soldered connections are good?
- The terminal screw are tight?