User Manual



ADAM-6200 Series

Intelligent Ethernet I/O Module

ADVANTECH

Enabling an Intelligent Planet

Copyright

The documentation and the software included with this product are copyrighted 2013 by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements in the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, nor for any infringements of the rights of third parties, which may result from its use.

Acknowledgements

Intel and Pentium are trademarks of Intel Corporation.

Microsoft Windows and MS-DOS are registered trademarks of Microsoft Corp.

All other product names or trademarks are properties of their respective owners.

Product Warranty

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

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

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

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

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

Part No. 2003620000 Edition 1

Printed in Taiwan June 2013

Declaration of Conformity

CE

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

FCC Class A

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

Technical Support and Assistance

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

Safety Instructions

1. Read these safety instructions carefully.
2. Keep this User Manual for later reference.
3. Disconnect this equipment from any AC outlet before cleaning. Use a damp cloth. Do not use liquid or spray detergents for cleaning.
4. For plug-in equipment, the power outlet socket must be located near the equipment and must be easily accessible.
5. Keep this equipment away from humidity.
6. Put this equipment on a reliable surface during installation. Dropping it or letting it fall may cause damage.
7. The openings on the enclosure are for air convection. Protect the equipment from overheating. DO NOT COVER THE OPENINGS.
8. Make sure the voltage of the power source is correct before connecting the equipment to the power outlet.
9. Position the power cord so that people cannot step on it. Do not place anything over the power cord.
10. All cautions and warnings on the equipment should be noted.
11. If the equipment is not used for a long time, disconnect it from the power source to avoid damage by transient overvoltage.
12. Never pour any liquid into an opening. This may cause fire or electrical shock.
13. Never open the equipment. For safety reasons, the equipment should be opened only by qualified service personnel.
14. If one of the following situations arises, get the equipment checked by service personnel:

- The power cord or plug is damaged.
- Liquid has penetrated into the equipment.
- The equipment has been exposed to moisture.
- The equipment does not work well, or you cannot get it to work according to the user's manual.
- The equipment has been dropped and damaged.
- The equipment has obvious signs of breakage.

15. DO NOT LEAVE THIS EQUIPMENT IN AN ENVIRONMENT WHERE THE STORAGE TEMPERATURE MAY GO BELOW -20° C (-4° F) OR ABOVE 60° C (140° F). THIS COULD DAMAGE THE EQUIPMENT. THE EQUIPMENT SHOULD BE IN A CONTROLLED ENVIRONMENT.
16. CAUTION: DANGER OF EXPLOSION IF BATTERY IS INCORRECTLY REPLACED. REPLACE ONLY WITH THE SAME OR EQUIVALENT TYPE RECOMMENDED BY THE MANUFACTURER, DISCARD USED BATTERIES ACCORDING TO THE MANUFACTURER'S INSTRUCTIONS.
17. The sound pressure level at the operator's position according to IEC 704-1:1982 is no more than 70 dB (A).

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

Contents

Chapter 1 Product Overview 1

1.1 Introduction 2
1.2 Feature Highlights 3
1.2.1 Easier Installation and Deployment 3
1.2.2 Integrated automation application with IT technologies 5
1.2.3 Excellent I/O application 5
1.2.4 Effective Management 5
1.3 ADAM-6200 Series Family 6
1.4 Mechanical Design and Dimensions 7
1.5 Switch 8
1.6 LED Definition 8
1.7 Certification and Safety Standard 9
1.8 Product Label 10
1.9 Front Name Plate 11
1.10 Package Information 11

Chapter 2 Product Specifications 13

2.1 General Specification 14
2.2 Analog Input Module (ADAM-6217) 15
2.2.1 Application Wiring 15
2.2.2 Pin Assignment: 16
2.2.3 Switch Setting 16
2.3 Analog Output Module (ADAM-6224) 17
2.3.1 Application Wiring 17
2.3.2 Pin Assignment 18
2.4 Digital Input/Output Modules (ADAM-6250/6251/6256) 18
2.4.1 Application Wiring 19
2.4.2 Pin Assignment 21
2.4.3 Switch Setting 23
2.5 Relay Output Modules (ADAM-6260/6266) 24
2.5.1 Application Wiring 25
2.5.2 Pin Assignment 26
2.5.3 Switch Setting 27

Chapter 3 Hardware Installation 29

3.1 Interface Introduction 30
3.2 Mounting 31
3.2.1 Panel Mounting 31
Figure 3.1 Bracket Dimension 31
Figure 3.2 Assembly with Bracket 31
3.2.2 DIN-Rail Mounting 32
Figure 3.3 Assembly With Din Rail Adapter 32
Figure 3.4 ADAM Module Mounted On A Rail 32
3.3 Wiring & Connections 33
3.3.1 Power Supply Wiring 33
3.3.2 I/O Units 33
3.3.3 Daisy Chain Connection 34

Chapter 4 System Configuration 37

4.1 ADAM.NET Utility (software) 38
4.2 Connection 38
4.3 Configure ADAM-6200 with ADAM.NET Utility 39
4.3.1 Operation Framework 40
4.3.2 General Configuration 50
4.3.3 I/O Configuration 57
4.3.4 Peer to Peer 80
4.3.5 GCL 85
4.3.6 Remotely Monitoring and Control via Web Server 119

Appendix A I/O Modbus Mapping Table 123

A.1 I/O Modbus Mapping Table 124
A.1.1 Modbus Function Code Introduction 124
A.1.2 ADAM-6217 8-ch Isolated Analog Input Module 124
A.1.3 ADAM-6224 4-ch Isolated Analog Output Module 127
A.1.4 ADAM-6250 15-ch Digital Input/Output Module 129
A.1.5 ADAM-6251 16-ch Digital Input Module 132
A.1.6 ADAM-6256 16-ch Digital Output Module 135
A.1.7 ADAM-6260 6-ch Relay Output Module 138
A.1.8 ADAM-6266 4-ch Relay Output Module with 4-ch DI 139

Appendix B ADAM-6200 ASCII Commands 141

B.1 ASCII Commands for ADAM-6200 modules 142
B.1.1 System Command Set 142
B.1.2 Analog Input Command Set 147
B.1.3 Analog Input Range code 156
B.1.4 Analog Input Alarm Command Set 156
B.1.5 Analog Output Command Set 165
B.1.6 Analog Output Range code 170
B.1.7 Digital Input/Output Command Set 170

Appendix C REST for ADAM-6200 175

C.1 Introduction 176
C.2 REST Resources for ADAM 176
C.2.1 Analog Input 176
C.2.2 Analog Output 178
C.2.3 Digital Input 181
C.2.4 Digital Output 182

Chapter 1

Product Overview

1.1 Introduction

The ADAM-6200 series is a new ADAM Ethernet I/O family and includes analog I/O, digital I/O and relay modules. Not limited to equipment with robust I/O functions, Advantech gets Ethernet switch involved in hardware design to make user easily deploy ADAM module with daisy chain connection and the connection is protected by auto-bypass when power outage occurs.Users are also able to monitor and control ADAM-6200 modules remotely by smart phone or pad in field maintenance. Advantech believes that, with new design and strong capabilities, ADAM-6200 can be a well-integrated I/O solution in Ethernet control system.



1.2 Feature Highlights

1.2.1 Easier Installation and Deployment

Daisy Chain connection with auto-bypass protection

ADAM-6200 module has built in Ethernet switches to allow daisy chain connections in an Ethernet network, making it easier to deploy, and helping improve scalability. The two Ethernet ports are fully compliant with IEEE 802.3u 10/100Mbpst through standard RJ-45 connectors

Although daisy chain topology brings attractive benefits for users, it still comes with the risk that once any device in the daisy-chain network suffers power outage, it will cause the disconnection of all devices downstream..



Note! ***Auto-bypass protection***

To prevent this critical issue from happening, Advantech especially refined the hardware design of ADAM-6200 so that it can rapidly recover the network connection in about 2.5 seconds, and endure for 4 days after module power outage. Therefore, the damage will be greatly minimized.

Group Configuration

In certain application scenarios, it's necessary to set multiple modules with the same settings because these modules are doing the same tasks on different sites. Users have to set configurations of module one after another before on-site deployment. After the modules are installed and the system is running, it will still require repetitive effort to carry out firmware updates.

The ADAM-6200 series modules are equipped with group configuration capability to reduce the repetitive effort and quickly finish the multiple module setups, including firmware upgrades, configuration and HTML 5 files, at one time. Users can start the implementation of modules faster as the time for configuration is tremendously reduced.

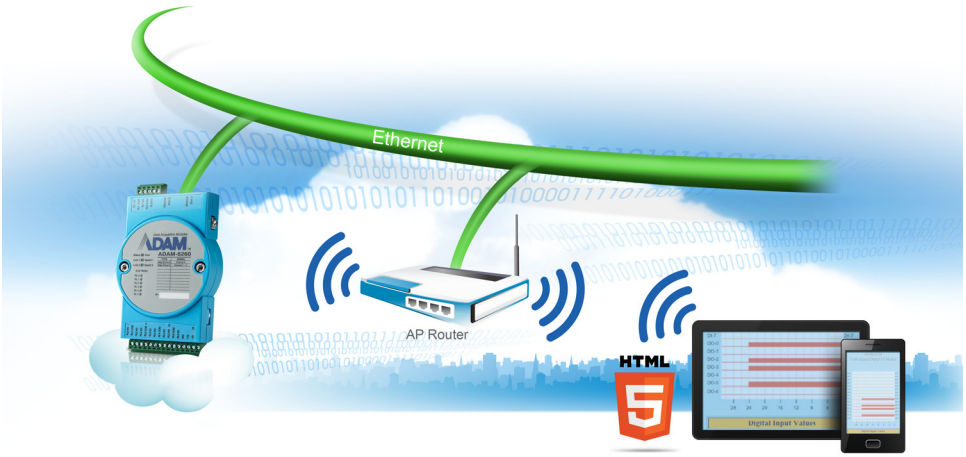


1.2.2 Integrated automation application with IT technologies

- **Remote Monitoring and Control with Smart Phone/Pad**

Previously, due to different communication modes and data formats, it wasn't easy to implement automation control and monitor in an IT-based infrastructure. Users need to build up a data converter to transform I/O data stream from SCADA system to IT database/management system.

The ADAM-6200 series module integrates the latest Web language (HTML 5) and Web-based software style (REST) with basic authentication for users to remotely acquire I/O data in any Web service of smart device without routing from SCADA system, for example, user can use Web browser of smart phone to remotely access I/O module via HTTP. Furthermore, it also can be implemented in Web API by REST access instruction..



HTML 5

Hyper Text Markup Language (HTML) is popularly used to program the content of Web page over the Internet. The fifth version (HTML 5) is the latest revision which enhances its syntax structure and additionally mixes multiple Web technologies like CSS, Java Script to implement more Web service, API, interactive applications in mobile communications.

1.2.3 Excellent I/O application

- Graphical Control Logic (GCL)
- Peer-to-Peer function
- User-defined Modbus address

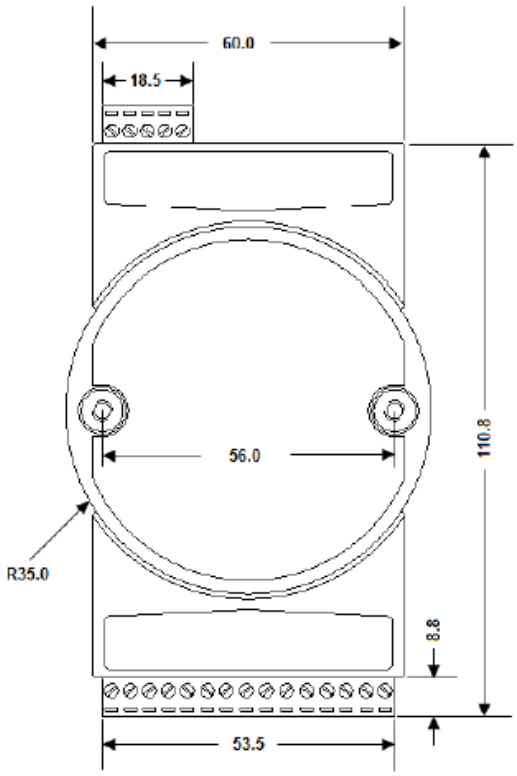
1.2.4 Effective Management

- User Access Control
- Configuration file backup
- Reset to factory default

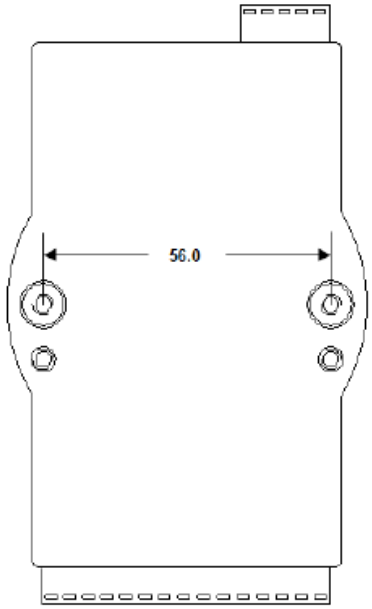
1.3 ADAM-6200 Series Family

Category	Model	Description
Analog I/O	ADAM-6217	8-ch Isolated Analog Input Modbus TCP Module
	ADAM-6224	4-ch Isolated Analog Output Modbus TCP Module
Digital I/O	ADAM-6250	15-ch Isolated Digital I/O Modbus TCP Module
	ADAM-6251	16-ch Isolated Digital Input Modbus TCP Module
	ADAM-6256	16-ch Isolated Digital Output Modbus TCP Module
Relay	ADAM-6260	6-ch Relay Output Modbus TCP Module
	ADAM-6266	4-ch Relay Output Modbus TCP Module with 4-ch DI

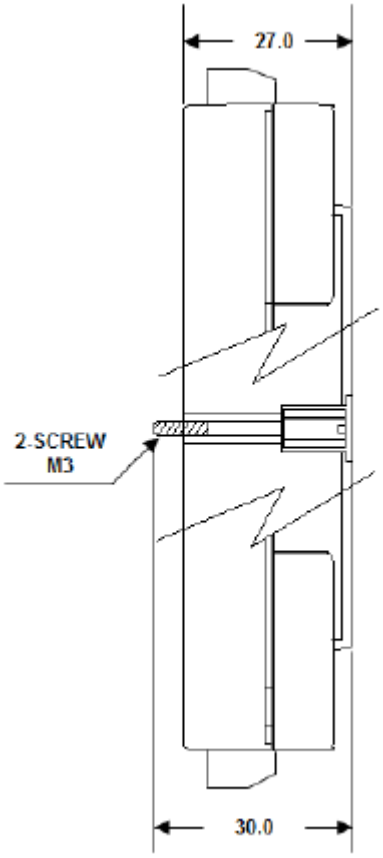
1.4 Mechanical Design and Dimensions



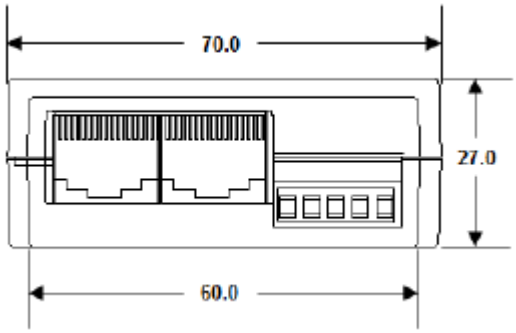
Front View



Bottom View



Side View



Top View

1.5 Switch

It's non-functional now, but reserved for future use.



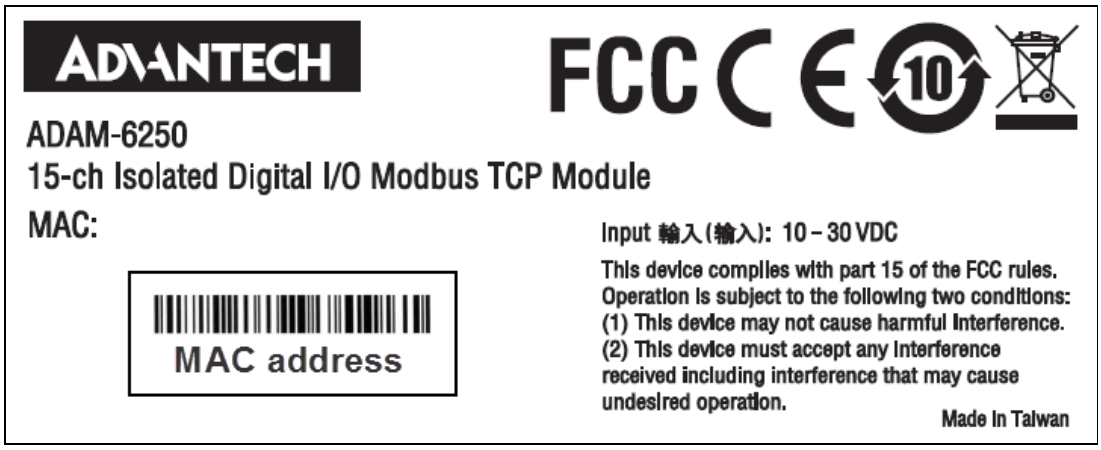
1.6 LED Definition

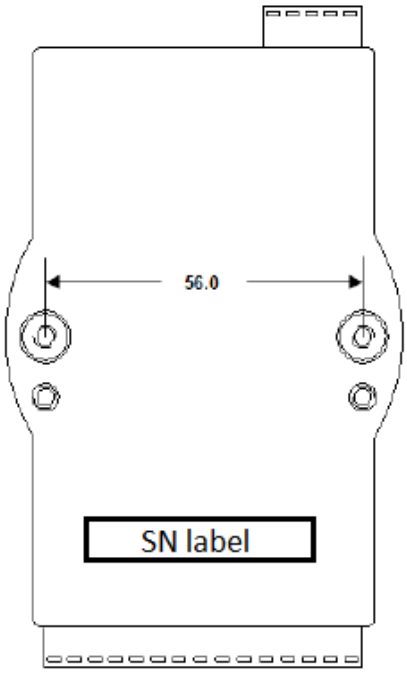
LED	Color	Indication	Behavior
Status/Com	Green (Status)	0.5 second ON 0.5 second OFF	Module is normally at work.
		Always 10 sec ON	When user enable LOCATE function.
	Yellow (Com)	Blink	When TX/RX data in transmission.
Link/Speed 1	Green (Link 1)	ON	Ethernet is connected.
		Blink	When TX/RX is in transmission
	Yellow (Speed 1)	ON	ON: Ethernet speed is at 100 Mbps OFF: Ethernet speed is less than 100 Mbps
Link/Speed 2	Green (Link 2)	ON	Ethernet is connected.
		Blink	When TX/RX is in transmission
	Yellow (Speed 2)	ON	ON: Ethernet speed is at 100 Mbps OFF: Ethernet speed is less than 100 Mbps
DI	Green	ON / OFF	Logic 1 / Logic 0
DO	Orange	ON / OFF	Logic 1 / Logic 0
RL	Orange	ON / OFF	Logic 1 / Logic 0

1.7 Certification and Safety Standard

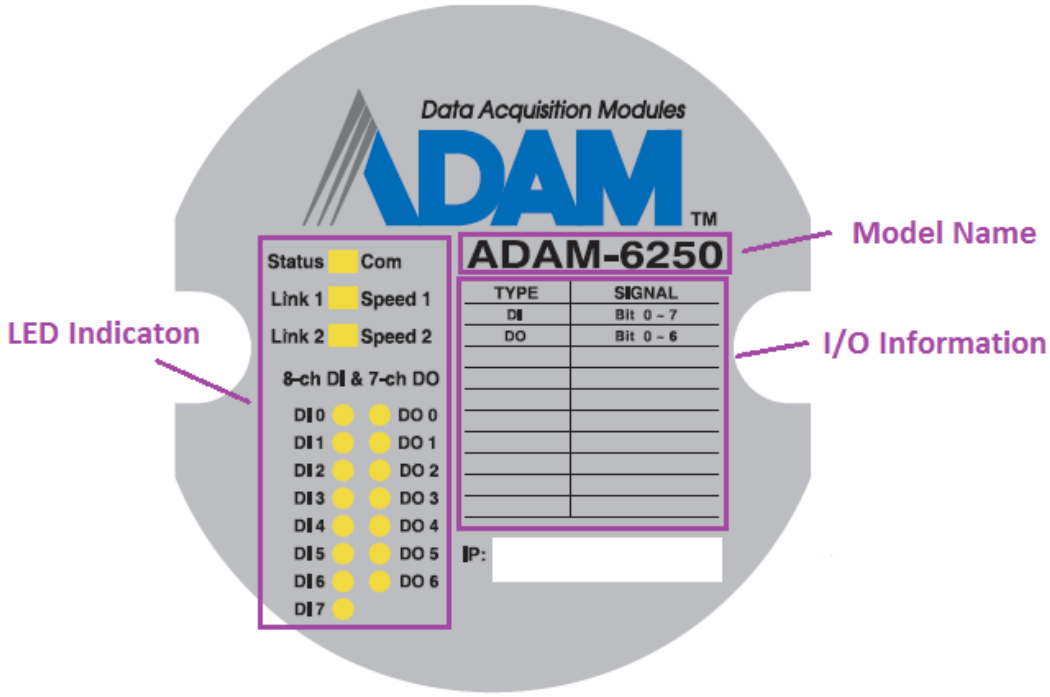
- FCC
  - FCC 47 CFR PART 15 (Class A)
  - IC ICES-003
- CE
  - EN 55011 / 55022 (Class A)
  - EN 61000-6-4
  - EN 61000-3-2
  - EN 61000-3-3
  - EN 55024
  - EN 61000-6-2
  - IEC 61000-4-2
  - IEC 61000-4-3
  - IEC 61000-4-4
  - IEC 61000-4-5
  - IEC 61000-4-6
  - IEC 61000-4-8
  - IEC 61000-4-11
  - RoHS
- China RoHS
- WEEE

1.8 Product Label





1.9 Front Name Plate



1.10 Package Information

The package of ADAM-6200 series module will contain the following items.

Please check and feel free to contact us if any part missing or damaged after purchasing ADAM-6200 product.

- ADAM-6200 module (assembled with DIN Rail)
- Product CD
- Panel mounting bracket
- Start-up manual
- China RoHS declaration

Chapter 2

Product Specifications

2.1 General Specification

Item	Description
Ethernet	2-port 10/100 Base-TX (for Daisy chain)
Protocol	Modbus TCP, TCP/IP, UDP, HTTP, DHCP
LED Indication	ADAM-6217: Status/Com, Link/Speed ADAM-6224: Status/Com, Link/Speed ADAM-6250: Status/Com, Link/Speed, 8x DI, 7x DO ADAM-6251: Status/Com, Link/Speed, 16x DI ADAM-6256: Status/Com, Link/Speed, 16x DO ADAM-6260: Status/Com, Link/Speed, 6x RL ADAM-6266: Status/Com, Link/Speed, 4x RL, 4x DI
Connector	Plug-in 5P/15P screw terminal blocks
Power Input	10 - 30 V_{DC} (24 V_{DC} standard)
Watchdog Timer	System (1.6 second) and Communication (Programmable)
Dimension	110 x 60 x 27 mm
Protection	Built-in TVS/ESD protection Power Reversal protection Over Voltage protection: +/- 35 V_{DC} Isolation protection: 2500 V_{DC}
Power consumption	ADAM-6217: 3.5W @24 V_{DC} ADAM-6224: 6.0W @24 V_{DC} ADAM-6250: 3.0W @24 V_{DC} ADAM-6251: 2.7W @24 V_{DC} ADAM-6256: 3.2W @24 V_{DC} ADAM-6260: 4.5W @24 V_{DC} ADAM-6266: 4.2W @24 V_{DC}
Operating temperature	-10 ~ 70°C (14 ~ 158°F)
Storage temperature	-20 ~ 80°C (-4 ~ 176°F)
Operating Humidity	20 ~ 95% RH (Non-condensing)
Storage Humidity	0 ~ 95% RH (Non-condensing)
Feature Highlights	Daisy chain connection with auto-bypass protection. Remote monitoring and control with smart phone/pad. Group configuration capability for multiple module setup. Flexible user-defined Modbus address. Intelligent control ability by Peer-to-peer and GCL function Multiple protocol support: Modbus TCP, TCP/IP, UDP, HTTP, DHCP Web language support: XML, HTML 5, Java Script System configuration backup User Access Control

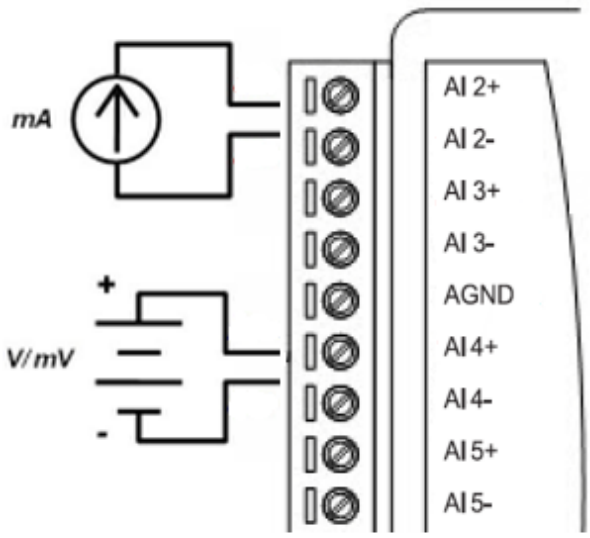
2.2 Analog Input Module (ADAM-6217)

Analog Input	Description
Channel	8 (Differential)
Voltage Input Range	±150 mV, ±500 mV, ±1 V, ±5 V, ±10 V
Current Input Range	±20 mA, 0 ~ 20 mA, 4 ~ 20 mA
Input Impedance	> 10 MΩ (Voltage) 120 Ω (Current)
Accuracy	±0.1% of FSR (Voltage) ±0.2% of FSR (Current)
Resolution	16-bit
Sampling Rate **(Total)**	10 samples/second
CMR @ 50/60 Hz	92 dB
NMR @ 50/60 Hz	60 dB
Span Drift	±30 ppm/°C
Zero Drift	±6 μV/°C
Common-Mode Voltage	200 V_{DC}
Other Functions	Independent Channel Range Type Independent Channel Enable Function Current Input Open-wired Detect Function Analog Input High/Low Alarm Function Historical Max/Min Value, Average Value

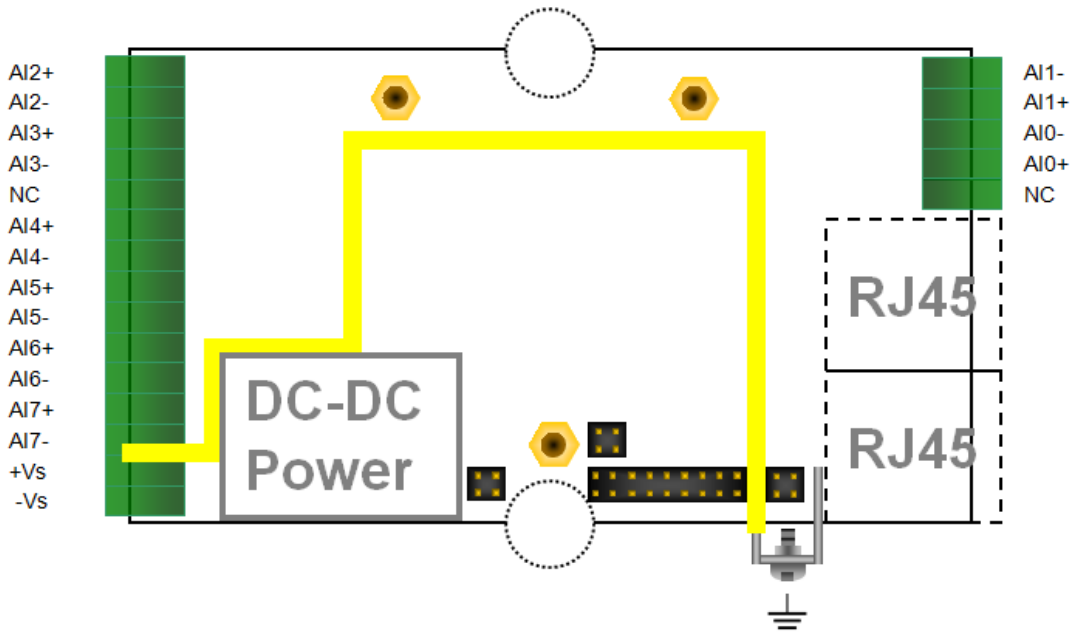
Warning! *1. The maximum input voltage range is +/- 10V. In order to avoid causing system errors and damaging the ADAM module, make sure the input voltage is within +/- 11V.*

2. The default setting of switch is voltage. To prevent system damage, make sure the switch and input signal are consistent.

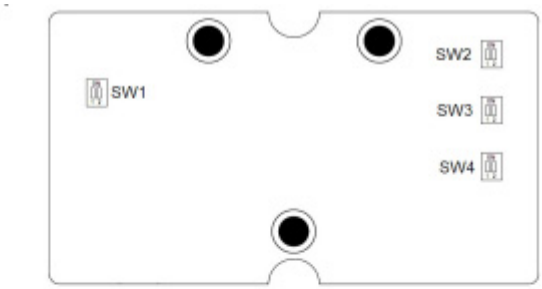
2.2.1 Application Wiring



2.2.2 Pin Assignment:



2.2.3 Switch Setting



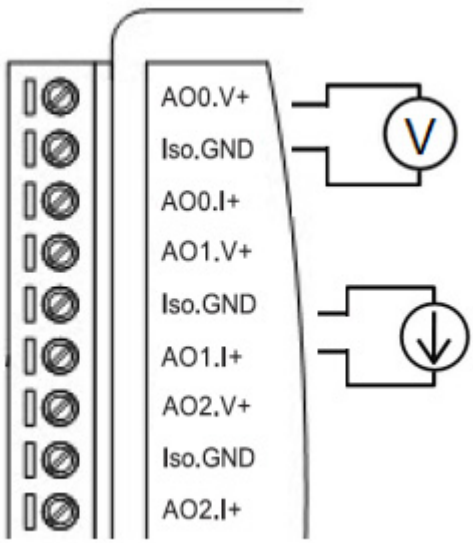
Switch	SW1		SW2		SW3		SW4	
Position	1	2	1	2	1	2	1	2
Channel	Ch1	Ch0	Ch3	Ch2	Ch5	Ch4	Ch7	Ch6
ON	Current Mode							
OFF	Voltage Mode (Default)							

2.3 Analog Output Module (ADAM-6224)

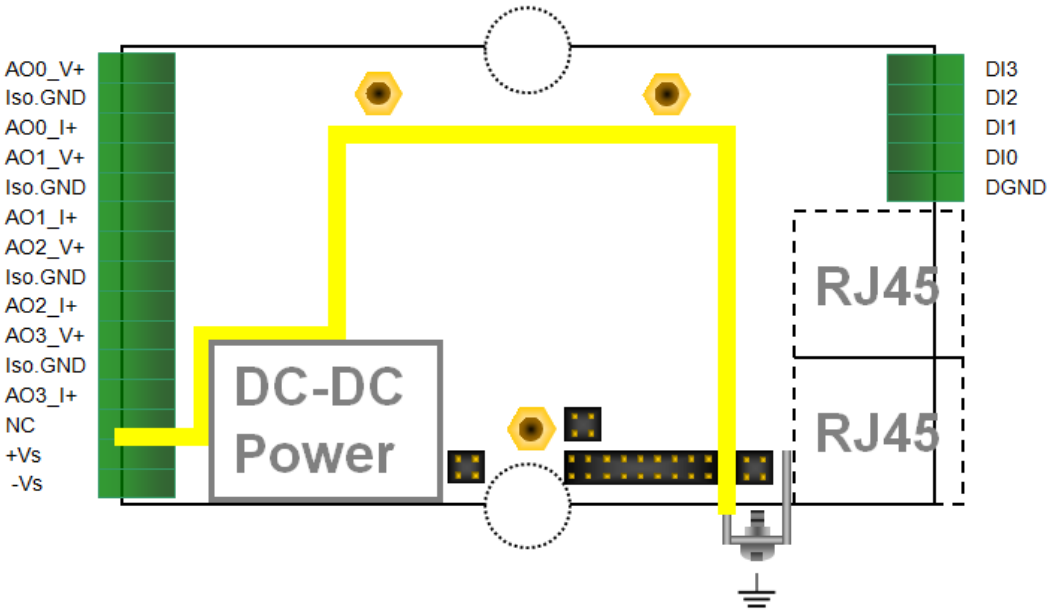
Analog Output	Description
Channel	4
Voltage Output Range	±5 V, ±10 V, 0~5 V, 0~10 V
Current Output Range	0 ~ 20 mA, 4 ~ 20 mA
Driving Load	Voltage: 2kΩ Current: 500 Ω
Output Impedance	2.1 Ω
Accuracy	0.3% of FSR (Voltage) at 25°C 0.5% of FSR (Current) at 25°C
Resolution	12-bit
Output settling Time	20 μs
Programmable Output Slope	0.125 – 128 mA/sec 0.0625 – 64 V/sec
Other Functions	Independent Channel Range Type Current Output Open-wired Detect Function Startup Value Setting Communication Safety Value Setting Digital Input Trigger Analog Output Function

Digital Input	Description
Channel	4
Dry Contact	Logic 0: Open Logic 1: Closed to Iso.GND
Other Functions	Support DI Filter Support Inverted DI Status Support Trigger to Startup or Safety Value

2.3.1 Application Wiring



2.3.2 Pin Assignment



2.4 Digital Input/Output Modules (ADAM-6250/6251/6256)

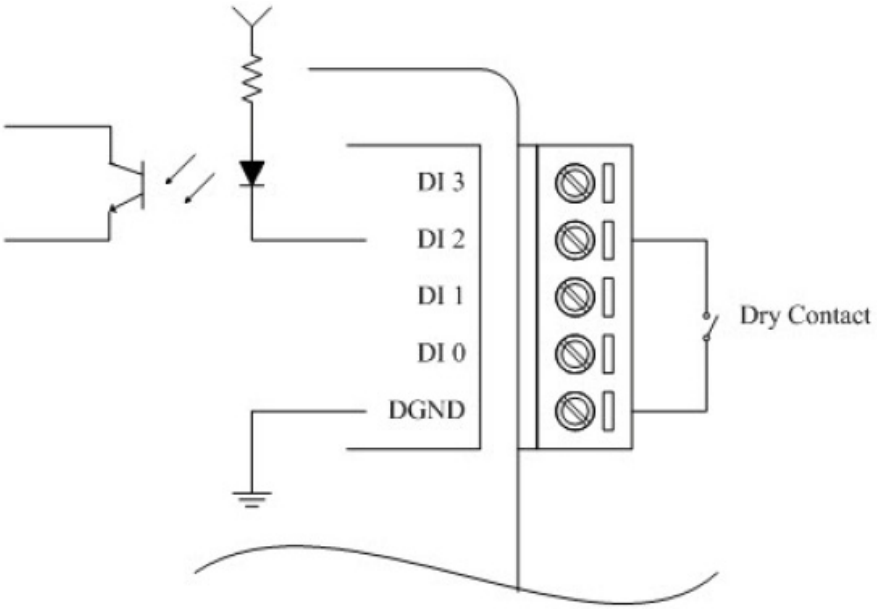
Digital Input	Description
Channel	ADAM-6250: 8 ADAM-6251: 16
Dry Contact	Logic 0: Open Logic 1: Closed to DGND
Wet Contact	Logic 0: 0 ~ 3 V_{DC} or 0 ~ -3 V_{DC} Logic 1: 10 ~ 30 V_{DC} or -10 ~ -30 V_{DC}
Input Impedance	5.2 kΩ (wet contact)
Transition Time	0.2 ms
Frequency Input Range	0.1 ~ 3kHz
Counter Input	3kHz (32-bit+1-bit overflow)
Other Functions	Keep/Discard Counter value when power off Support Inverted DI status

Digital Output	Description
Channel	ADAM-6250: 7 (Sink Type) ADAM-6256: 16 (Sink Type)
Output Voltage Range	10 ~ 30 V_{DC}
Normal Output Current	100 mA (per channel)
Pulse Output	Up to 5kHz
Delay Output	High-to-Low and Low-to-High

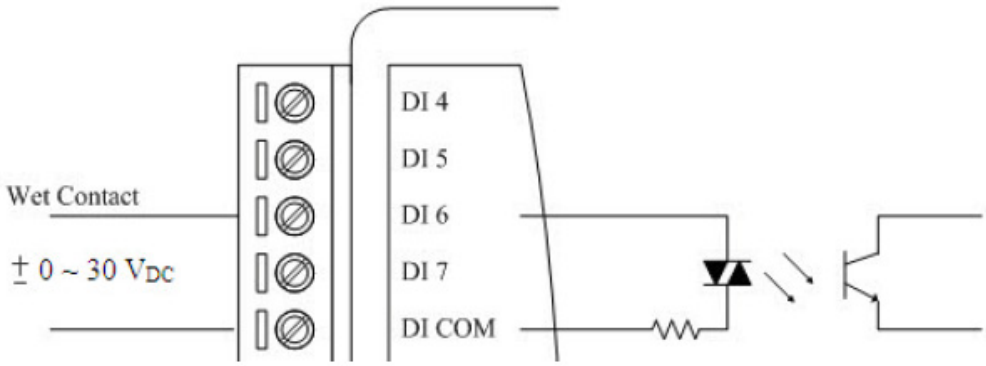
2.4.1 Application Wiring

- **Isolated Digital Input (6250/6251)**

Dry Contact

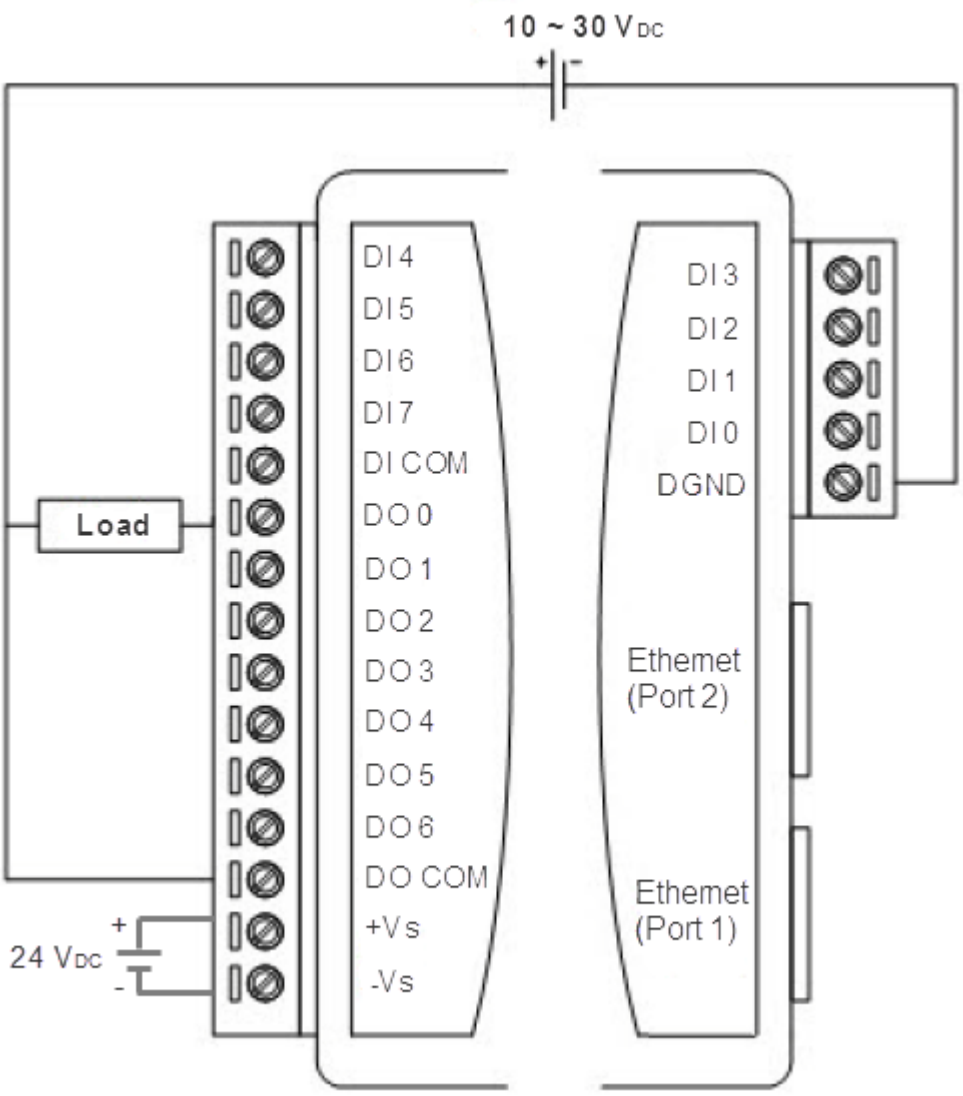


Wet Contact

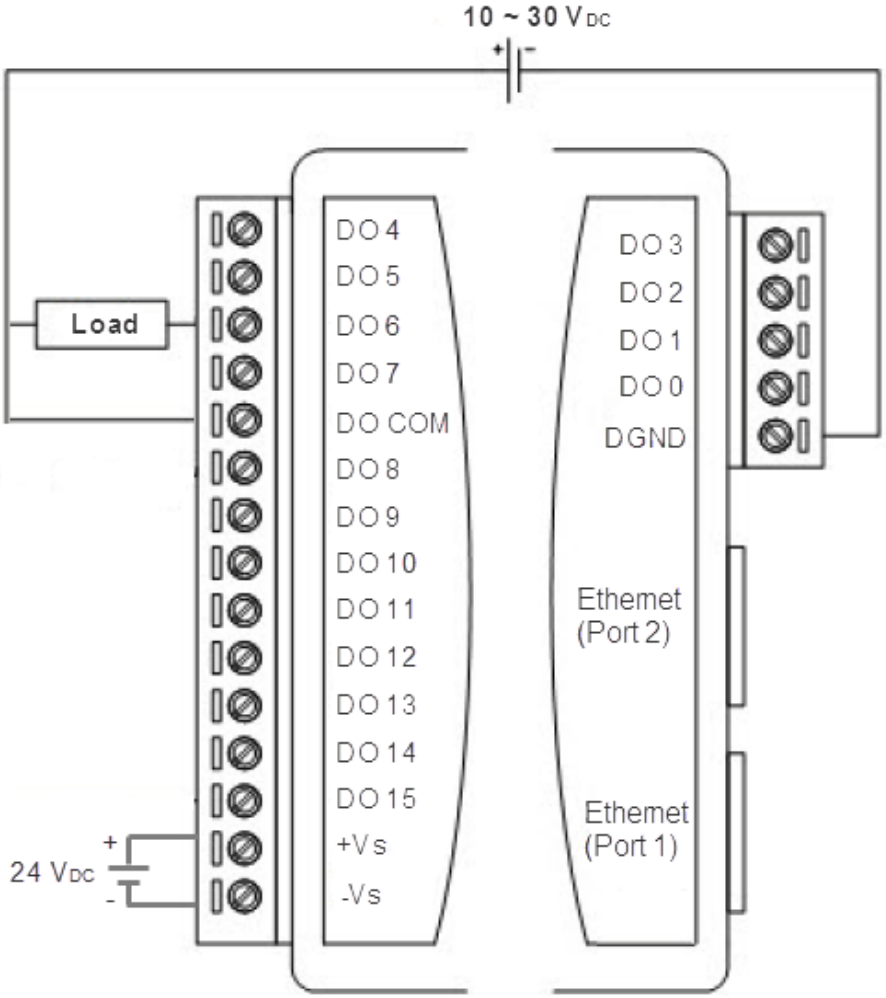


Isolated Digital Output

ADAM-6250

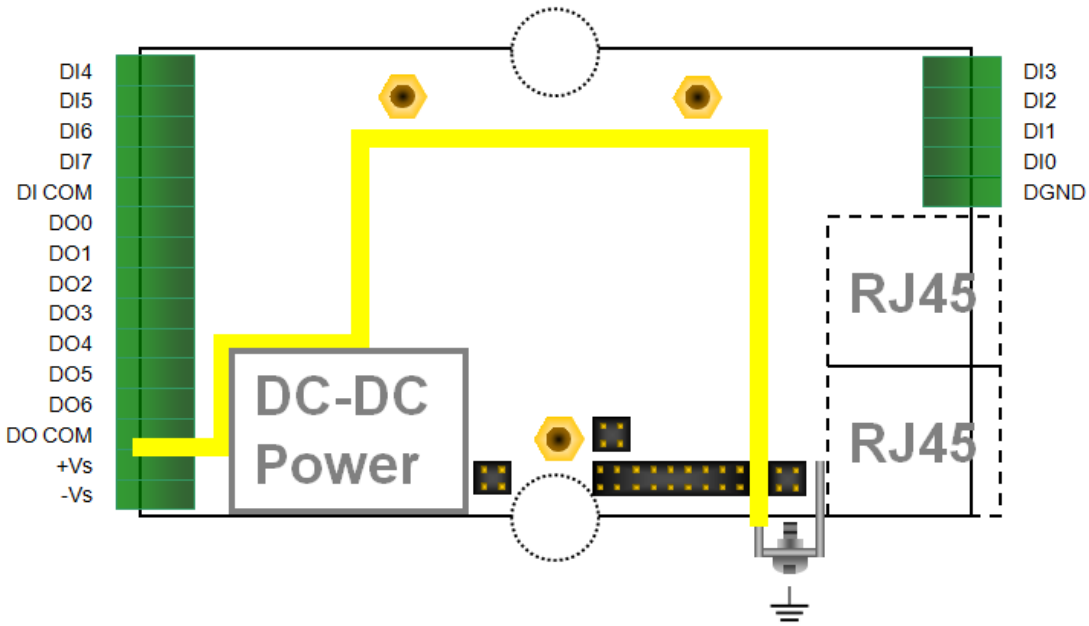


ADAM-6256

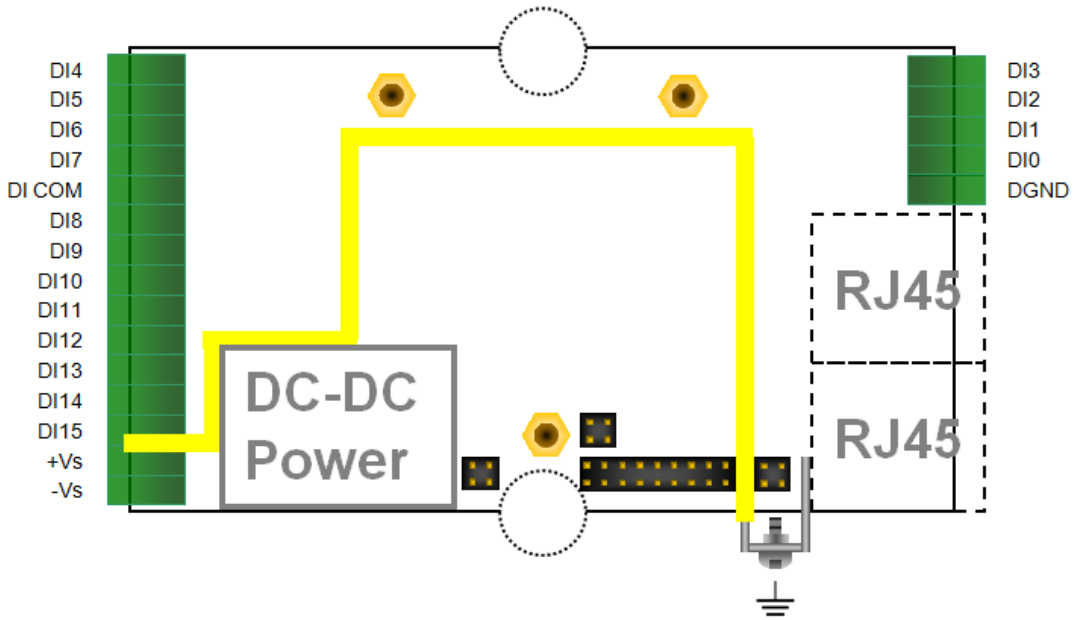


2.4.2 Pin Assignment

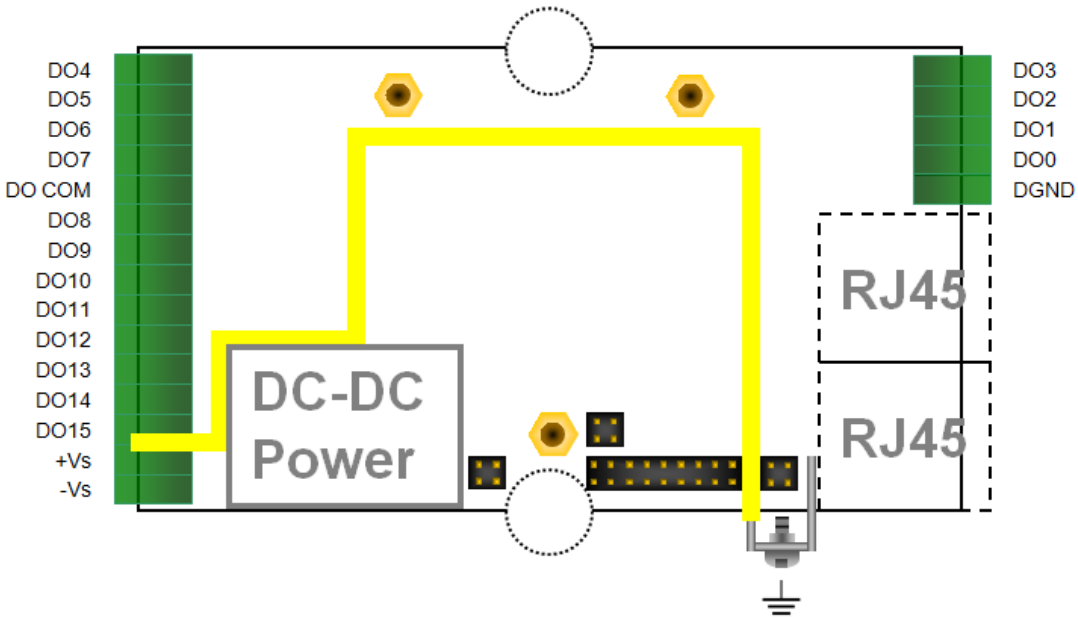
■ **ADAM-6250**



ADAM-6251

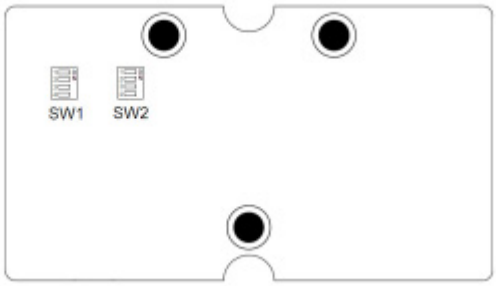


ADAM-6256



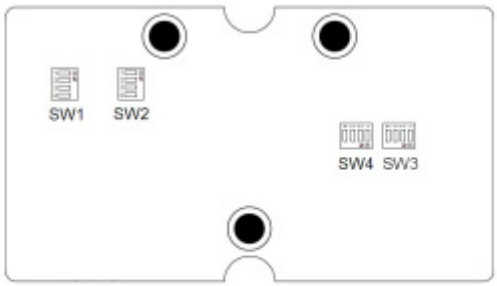
2.4.3 Switch Setting

- **ADAM-6250**



Switch	SW1				SW2			
Position	1	2	3	4	1	2	3	4
DI Channel	Ch3	Ch2	Ch1	Ch0	Ch7	Ch6	Ch5	Ch4
ON	Dry Contact (Default)							
OFF	Wet contact							

- **ADAM-6251**



Switch	SW1				SW2			
Position	1	2	3	4	1	2	3	4
DI Channel	Ch3	Ch2	Ch1	Ch0	Ch7	Ch6	Ch5	Ch4
Switch	SW3				SW4			
Position	1	2	3	4	1	2	3	4
DI Channel	Ch11	Ch10	Ch9	Ch8	Ch15	Ch14	Ch13	Ch12
ON	Dry Contact (Default)							
OFF	Wet contact							

2.5 Relay Output Modules (ADAM-6260/6266)

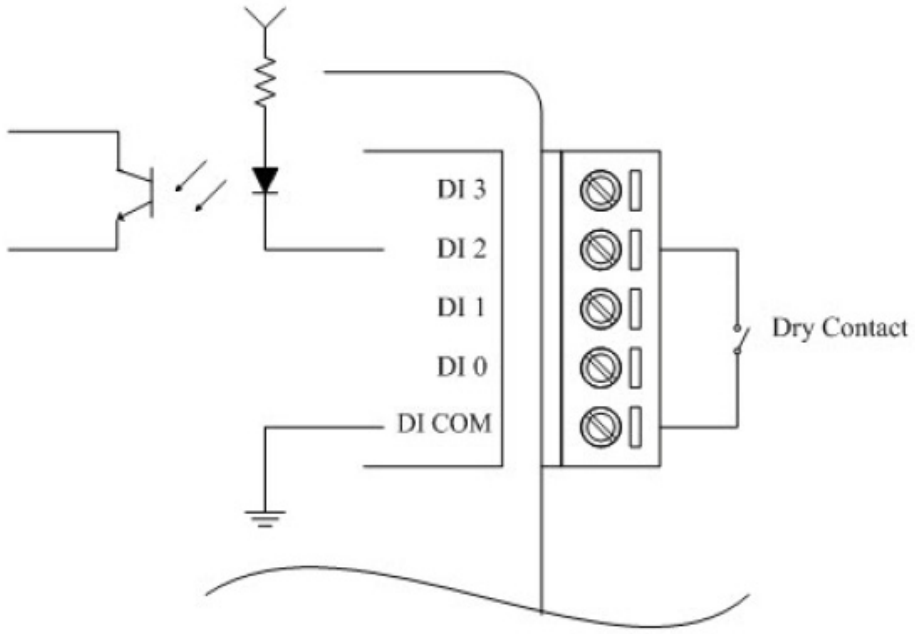
Relay Output	Description
Channel	ADAM-6260: 5 Form C and 1 Form A ADAM-6266: 4 Form C
Contact Rating	250 V_{AC} @ 5A 30 V_{DC} @ 5 A
Max. Switching Voltage	400 V_{AC} 300 V_{DC}
Max. Breaking Capacity	1250 VA
Frequency of Operation	360 operations/hour with load 72,000 operations/hour without load
Set/Reset Time	8 ms/8 ms
Mechanical Endurance	> 15 x 10^6 operations
Isolation between Contact:	1000 V_{rms}
Contact Resistance	< 30 mΩ
Insulation Resistance	> 10 GΩ @ 500 V_{DC}

Digital Input	Description
Channel	ADAM-6266: 4
Dry Contact	Logic 0: Open Logic 1: Closed to DI COM
Wet Contact	Logic 0: 0 ~ 3 V_{DC} or 0 ~ -3 V_{DC} Logic 1: 10 ~ 30 V_{DC} or -10 ~ -30 V_{DC}
Input Impedance	5.2 kΩ (wet contact)
Transition Time	0.2 ms
Frequency Input Range	0.1 ~ 3kHz
Counter Input	3kHz (32-bit+1-bit overflow)
Other Functions	Keep/Discard Counter Value when power off Support Inverted DI Status

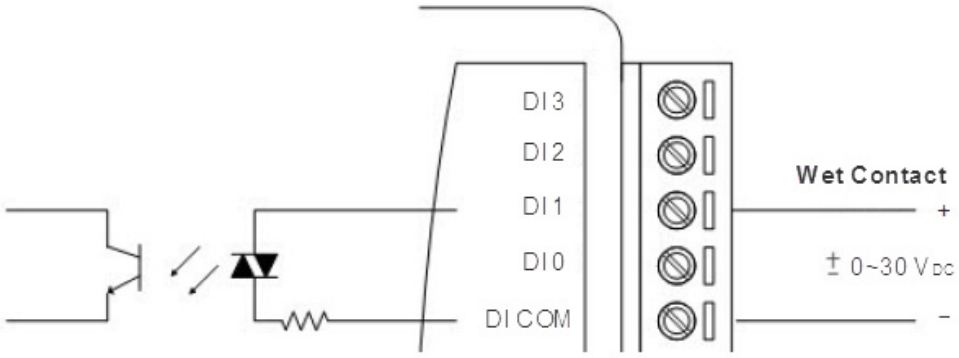
2.5.1 Application Wiring

- **Isolated Digital Input**

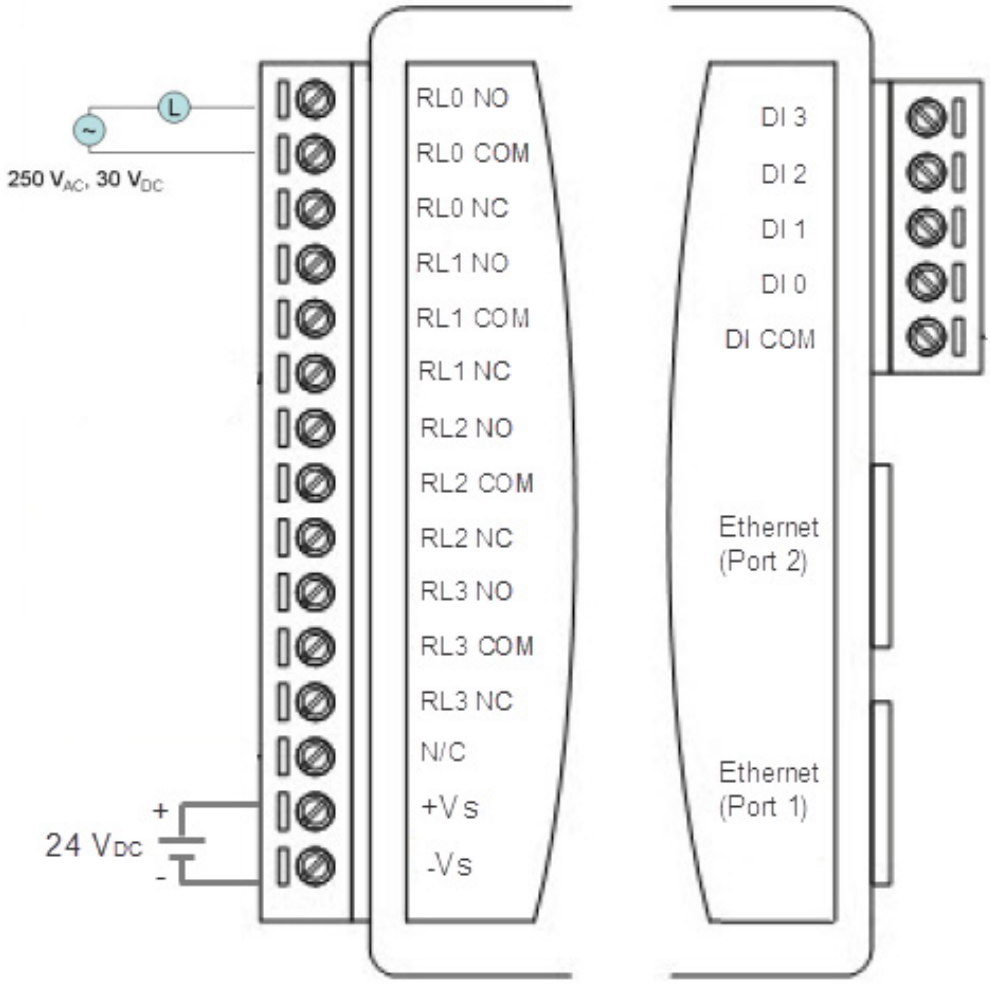
Dry Contact



Wet Contact

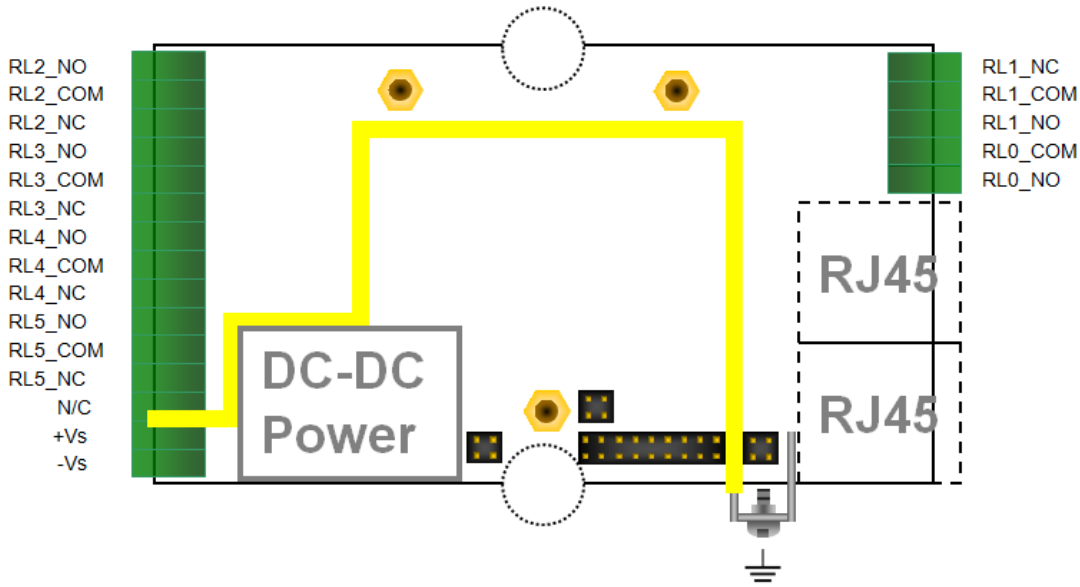


- **Isolated Relay Output**

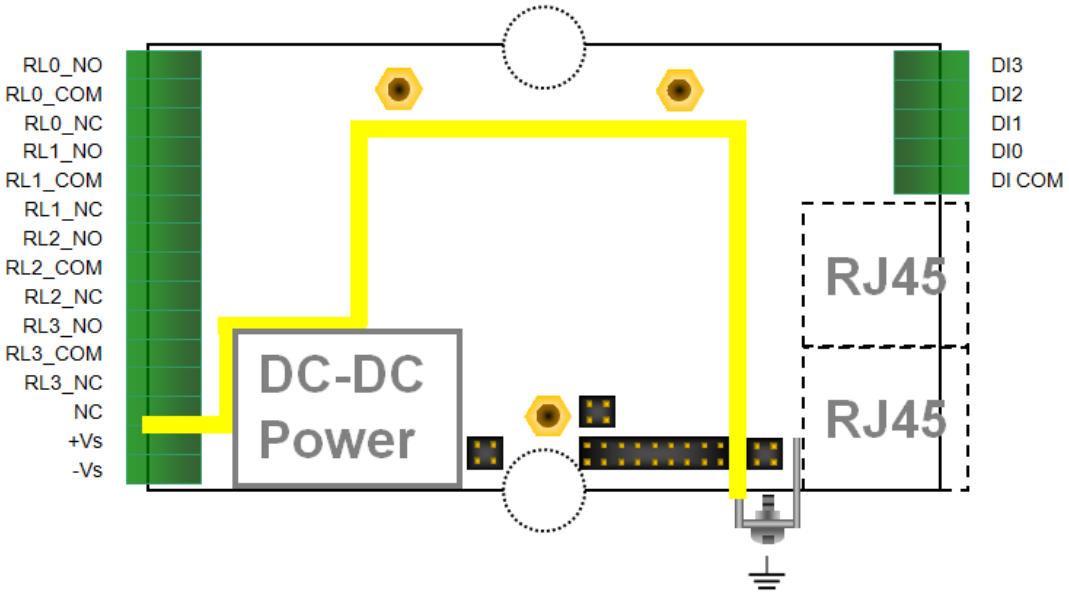


2.5.2 Pin Assignment

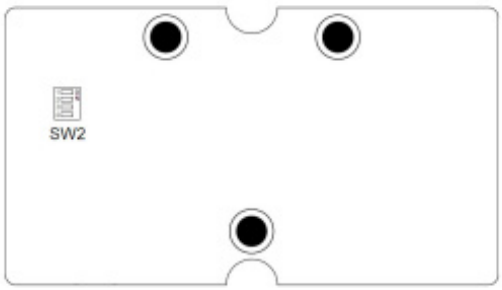
- **ADAM-6260**



■ **ADAM-6266**



2.5.3 Switch Setting

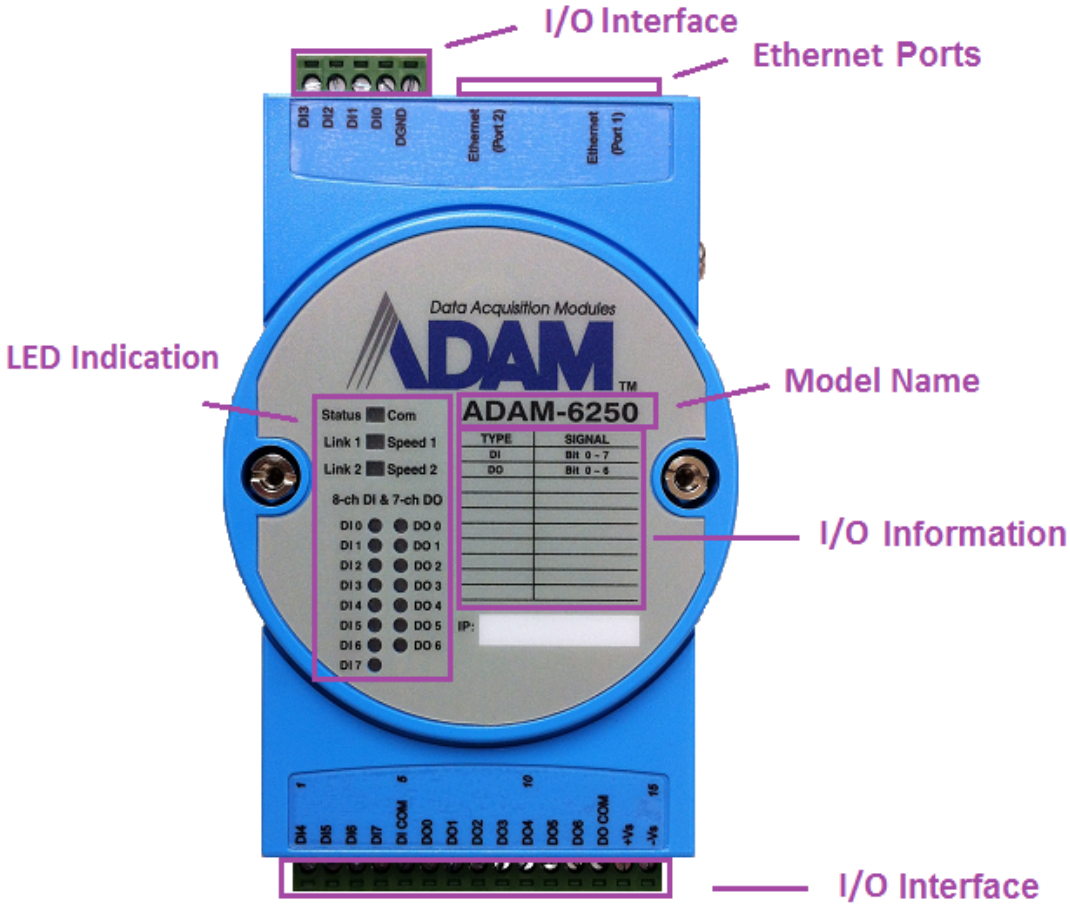


Switch	SW2			
Position	1	2	3	4
DI Channel	Ch0	Ch1	Ch2	Ch3
ON	Dry Contact (Default)			
OFF	Wet contact			

Chapter 3

Hardware Installation

3.1 Interface Introduction



3.2 Mounting

ADAM-6200 modules are designed as compact units and are allowed to be installed in the field site under the following methods.

3.2.1 Panel Mounting

Each ADAM-6200 Module is packed with a plastic panel mounting bracket. User can refer the bracket dimension and assembling figure to configure an optimal placement in a panel or cabinet.

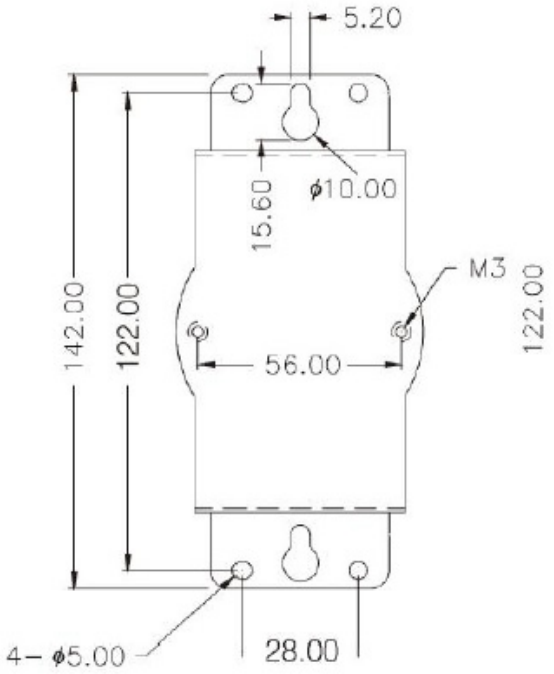


Figure 3.1 Bracket Dimension

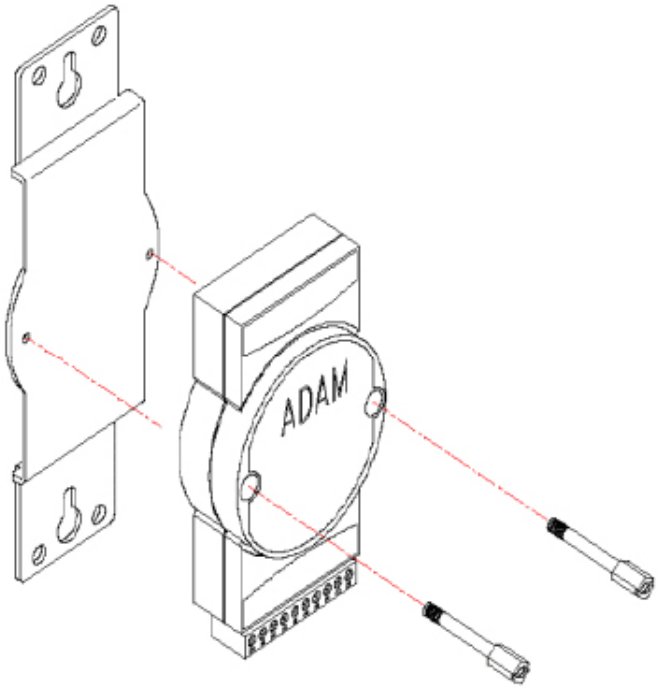


Figure 3.2 Assembly with Bracket

3.2.2 DIN-Rail Mounting

The ADAM-6200 module can also be fixed to the cabinet by using mounting rails. You need to assemble the DIN rail adapter to ADAM-6200 module with flathead screwdriver as below. When the module is mounted on a rail, you may also consider using end brackets at each end of the rail to keep the module from sliding horizontally along the rail.

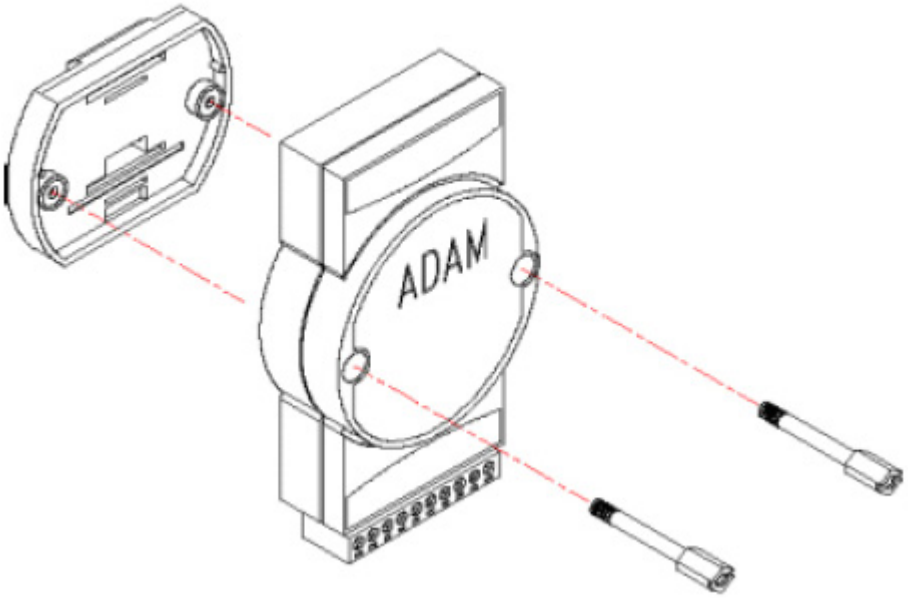


Figure 3.3 Assembly With Din Rail Adapter

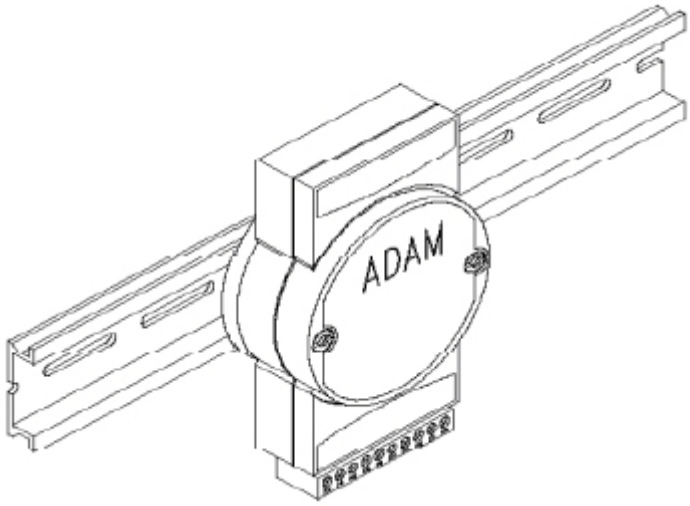


Figure 3.4 ADAM Module Mounted On A Rail

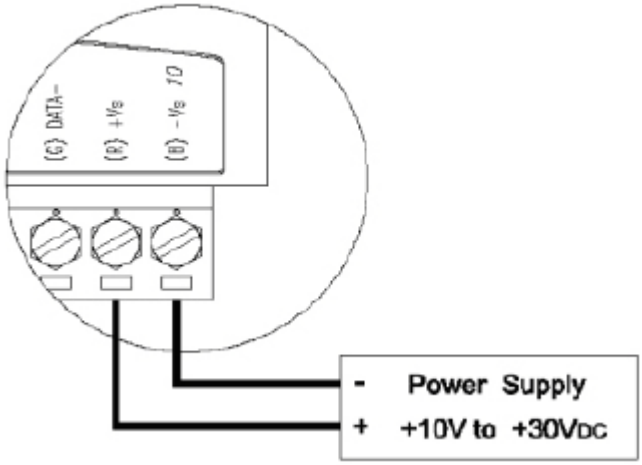
3.3 Wiring & Connections

This section introduces basic information on wiring the power supply, I/O units, and Ethernet connection.

3.3.1 Power Supply Wiring

The system of ADAM-6200 is designed for a standard industrial unregulated 24 V_{DC} power supply. For further application, it can also accept +10 to +30 V_{DC} of power input, 200mV peak to peak of power ripple, and the immediate ripple voltage should be maintained between +10 and +30 V_{DC}.

Screw terminals +Vs and -Vs are for power supply wiring



Note! *The wires used should be at least 2 mm.*

3.3.2 I/O Units

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

1. The terminal block accepts wires from 0.5 mm to 2.5 mm.
2. Always use a continuous length of wire. Do not combine wires.
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.
7. Avoid creating sharp bends in the wires.

3.3.3 Daisy Chain Connection

In early stage of industrial automation field, most users used to deploy their automation environment as serial communication RS-422/RS-485 interface. The typical connection mode with RS-422/RS-485 is field connection. That presents each device can be connected one after one in one line.

Since each ADAM-6200 module has already built in Ethernet switch, user can easily connect each ADAM-6200 module with standard Ethernet cable as following roughly diagram shows.

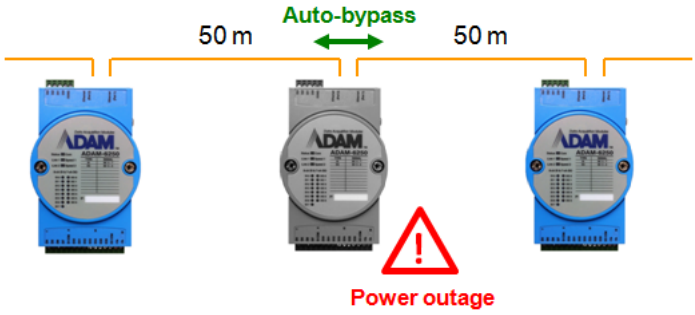


The benefits of Daisy chain topology

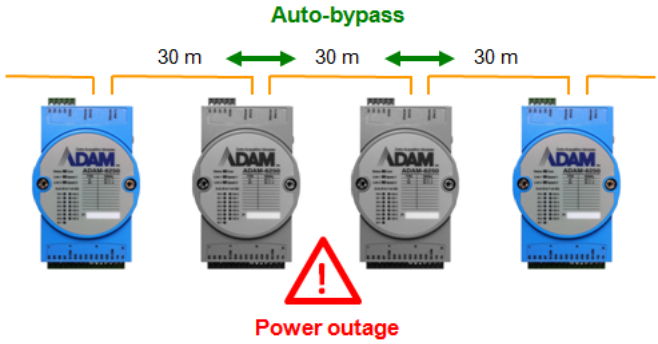
- Easily expands the module in existing deployment.
- Helps improve flexibility of I/O deployment, save wiring cost and space.
- Auto-bypass provides reliable communication networking.

Note!

1. The average latency from one module to another: 15 - 20 µs:
2. Typically, the maximum cabling distance of each 100BASE-TX network segment is 100 meters. Based on this limitation, the maximum total connection length of daisy chain wiring should also be 100m when auto-bypass protection active. See the following figure as an example, the distance from the first to the second module is 50m, so is the second to third. When the power fails on second (middle) module, the auto-bypass will activate to bridge the network connection. The total distance from first to the third will become 100m, that means the total network segment is close to its limit.



- Furthermore, due to the characteristics of signal transmission, the continuous auto-bypass nodes should be less than two, so that the network communication in the daisy chain will work.



3. The purpose of the auto-bypass protection is to help users minimize the damage caused by a module power outage. Note that when a power outage occurs on ADAM-6200 modules, there is around a 2.5 second buffer time for internally switching to the auto-pass circuit. This means that Ethernet packets can't reach the other devices in the daisy-chain for around 2.5 seconds. The valid connection duration after the auto-bypass function is enabled is at most around four days, otherwise, the Ethernet communication will be interrupted until power is restored.

Chapter 4

System Configuration

4.1 ADAM.NET Utility (software)

ADAM.NET Utility, which is designed with graphical operation interface, is aimed to offer users directly configure, control ADAM module, and monitor the real-time status of remote ADAM module via Ethernet or Wireless connection.

You can find ADAM.NET utility and user manual from ADAM product CD. To keep you informed with latest update, you also can check it from the following download link on Advantech website.
http://support.advantech.com.tw/Support/DownloadSRDetail.aspx?SR_ID=1-2AKUDB

Note!

- *Before installing ADAM.NET Utility, you need to install .NET Framework 2.0 or higher version.*
- *System requirement*
  - *Microsoft Windows XP/7*
  - *At least 32 MB RAM*
  - *20 MB of hard disk space available*
  - *VGA color or higher resolution monitor*
  - *2x or higher speed CD-ROM*
  - *Mouse or other pointing devices*
  - *10/100 Mbps or higher Ethernet Card*

4.2 Connection

1. Plug DC power source in +Vs, -Vs pin of ADAM module and turn the power on.
2. Connect your computer to Ethernet port (either 1 or 2) of ADAM module with RJ-45 cross-over Ethernet cable. (see below figure.)



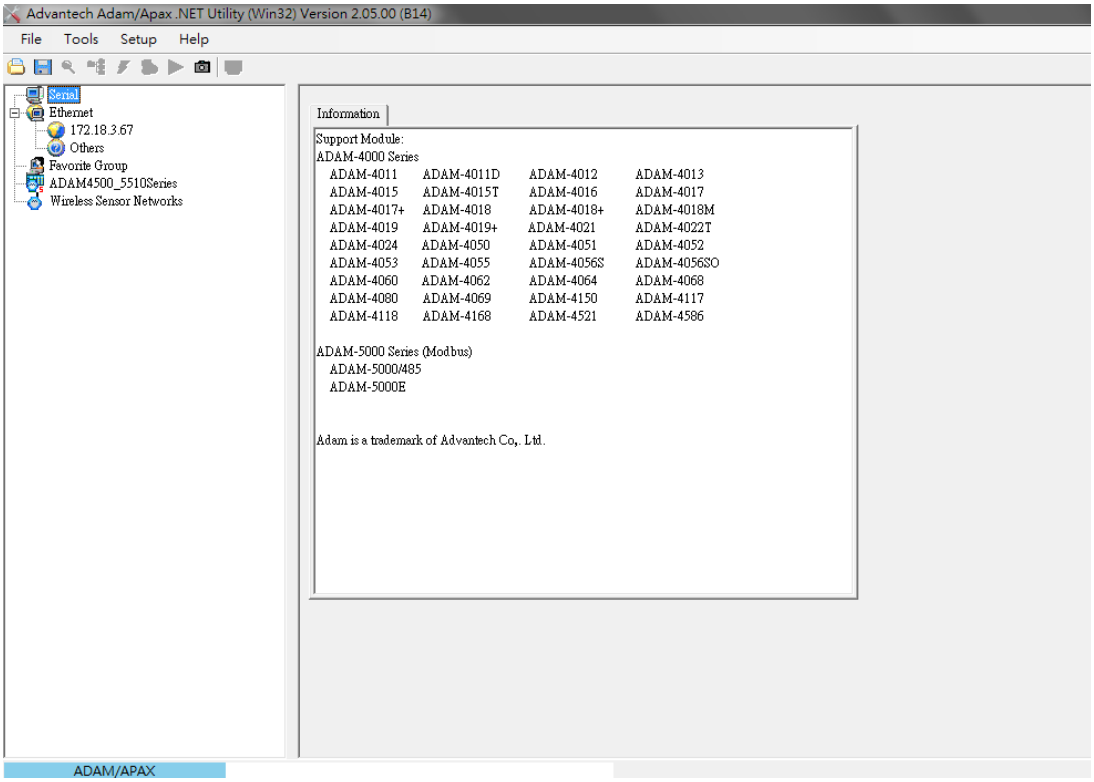
4.3 Configure ADAM-6200 with ADAM.NET Utility

After you've finished the connection between ADAM-6200 module and Host PC, you can use ADAM.NET utility to configure.

1. Install ADAM.NET Utility in your computer.
   (After successfully installation, there will be a shortcut generated on the screen)

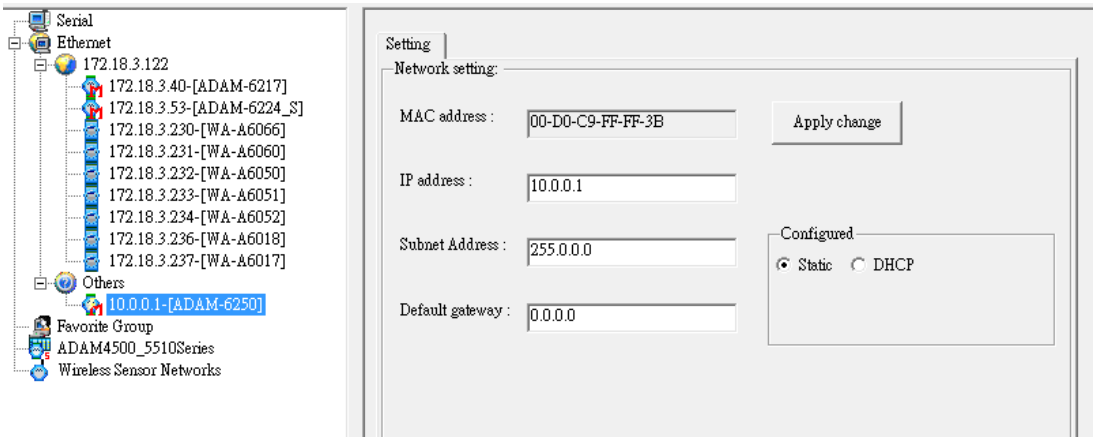


2. Double click the shortcut icon, and then you will see the main operation window as below.



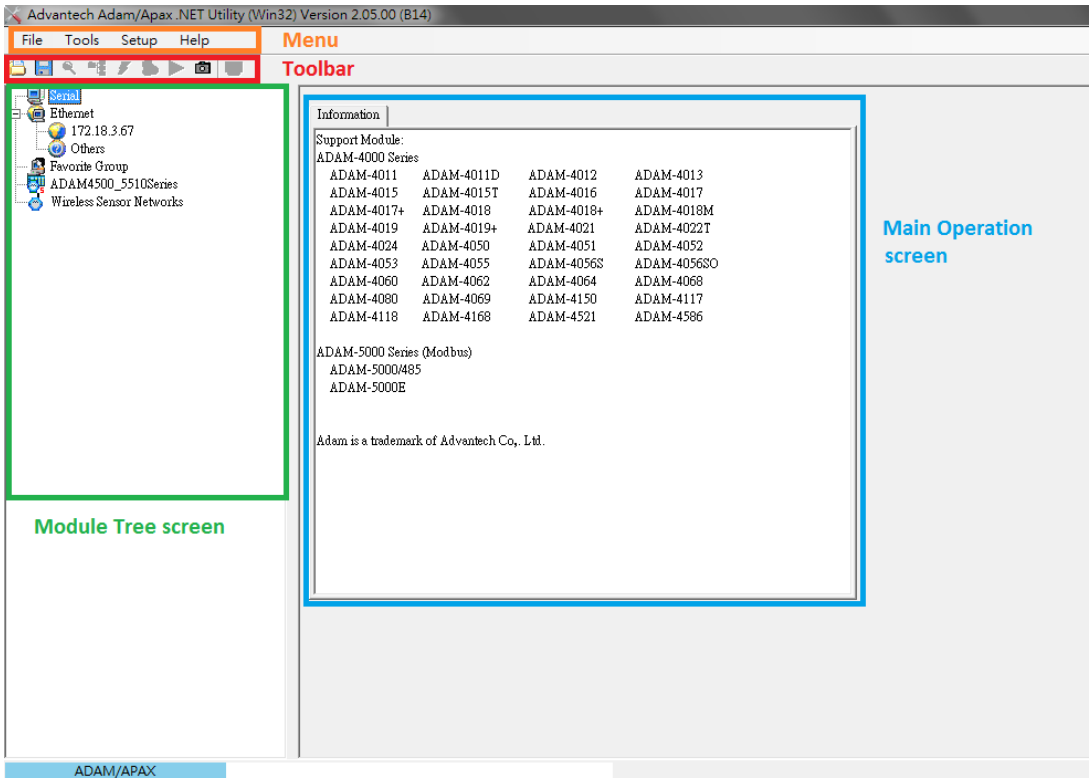
3. Click Search Module icon in Toolbar. You will see all online modules in the left Module Tree screen and an unconfigured new module, whose default password is 00000000, will appear on the Others section as below. Now you can define the network mode of the module in the beginning. After that, you will be able to perform other settings.

Note! *The default password is 00000000*



4.3.1 Operation Framework

The operation window mainly contains 4 areas, including Menu, Toolbar, Module Tree screen and Main Operation screen.



4.3.1.1 Menu

a. File

- **Open Favorite Group**
  You can import the favorite configuration group file (.XML) from your computer.
- **Save Favorite Group**
  You can save the favorite group configuration group as XML file to your computer.
- **Auto-Initial Group**
  If you want to have the same favorite group configuration when you exit ADAM.NET utility and launch it again, you need to check this option.
- **Exit**
  Exit ADAM.NET Utility.

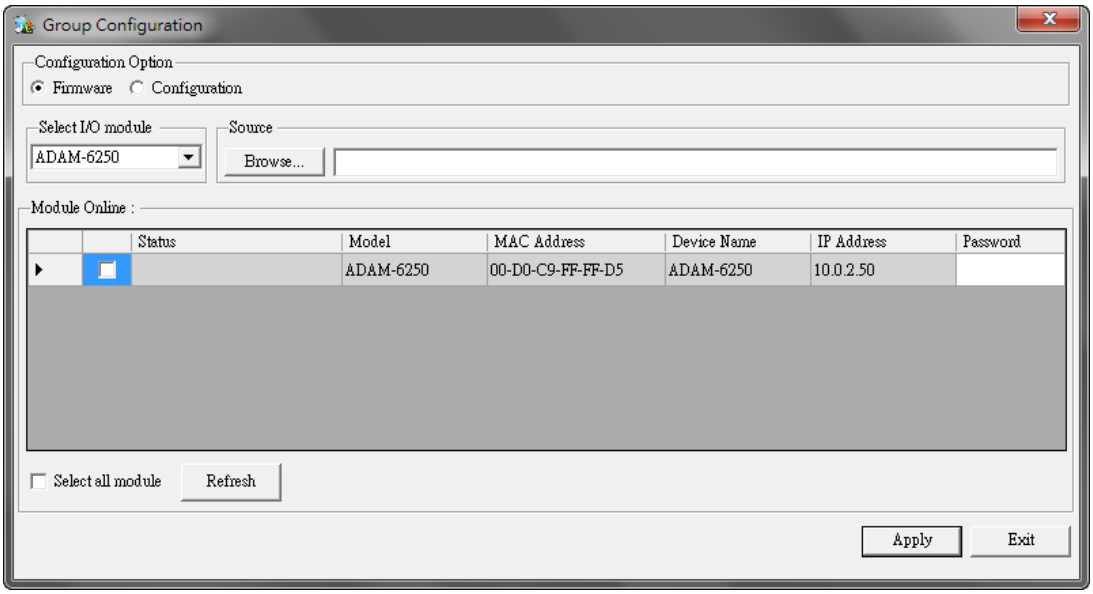
b. Tools

- **Search Device**
  Search all the ADAM modules you connected in local Ethernet.
- **Add Devices to Group**
  It's used to add ADAM modules to your favorite group. After activating search function, all online modules will show on Module Tree Screen area. Now you can enable this function to select the device you want to add in the Module Tree Screen.
- **Group Configuration**
  Group Configuration is on ADAM-6200 series module. It can help you efficiently configure or maintain massive ADAM modules with the same configuration file or firmware upgrade at one time in the local network. The following steps will instruct you how to operate it.

Case 1. If you want to upgrade your module's firmware.

1. Select Firmware.
2. Select I/O module
3. Click Browse button to import the firmware file from your computer.
4. Choose which module you're going to change its setting and enter password.
5. Click Apply button to activate it, and then you will see the operating progress on the Status area.

Note! *The default password is 00000000*

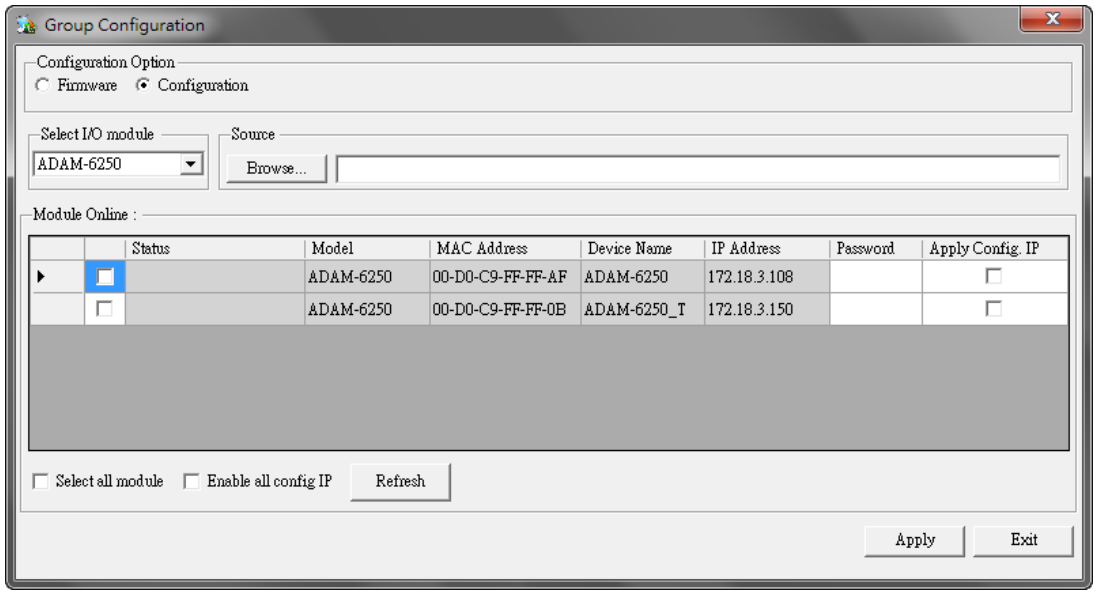


Note! *Please do not remove the power of your module when group configuration function is processing. Otherwise, the module system will probably crash.*

Case 2. If you want to duplicate your module's configuration to others.

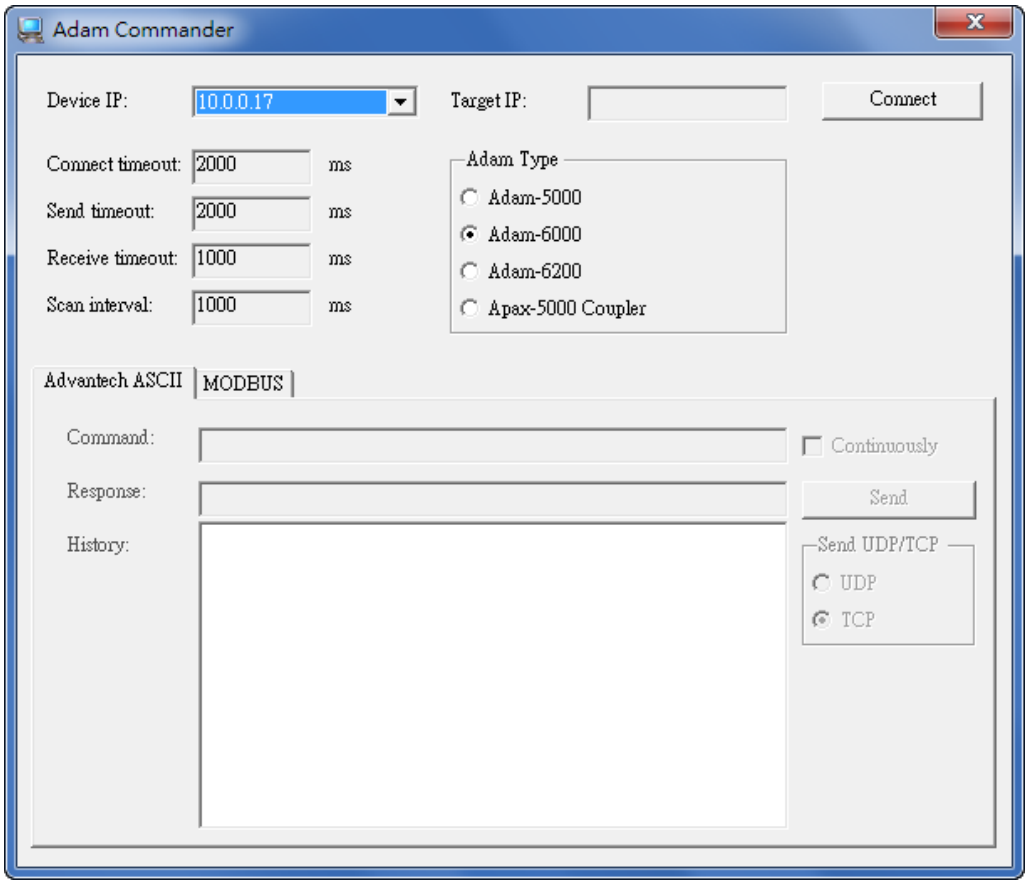
(Before launching it, you need to export the configuration file to your computer in advance)

1. Select Configuration.
2. Select I/O module
3. Click Browse button to import the configuration file from your computer.
4. Choose which module you're going to change its setting and enter password.
5. Click Apply button to activate it, and then you will see the operating progress on the Status area.

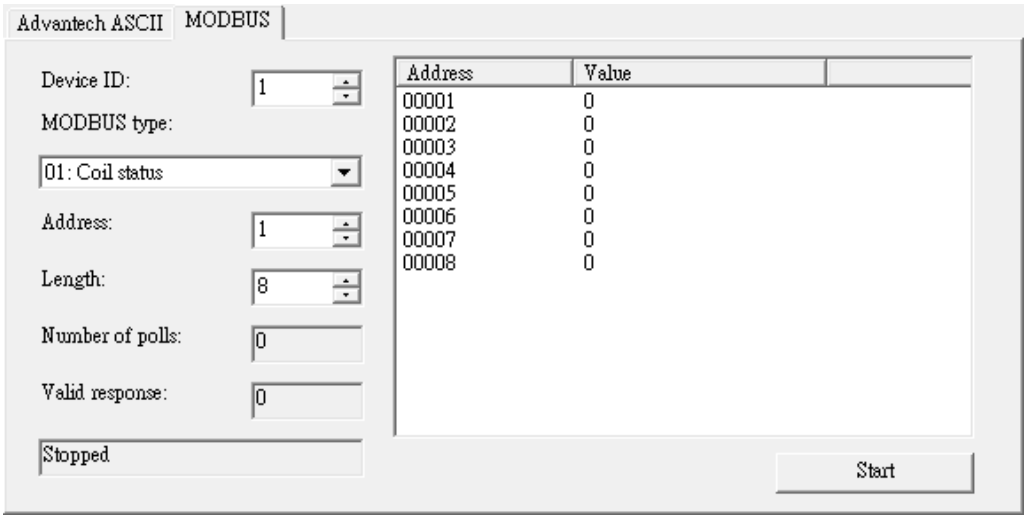


Note! *Do not remove the power of your module when group configuration function is processing. Otherwise, the module system will probably crash.*

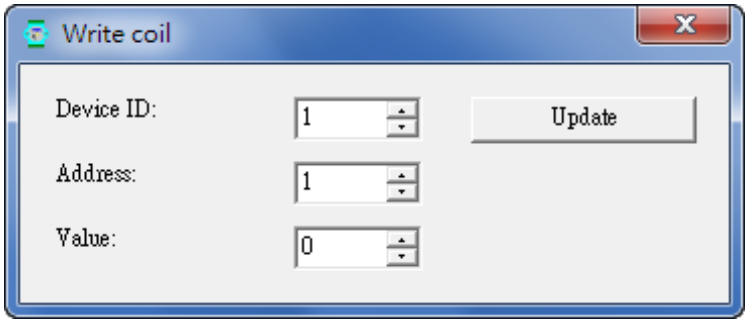
- **Terminal for Command Testing**
  ADAM-6200 series module support ASCII command and Modbus/TCP as communication protocol, so you can launch the terminal to directly communicate with ADAM-6200 series module by these two protocols.

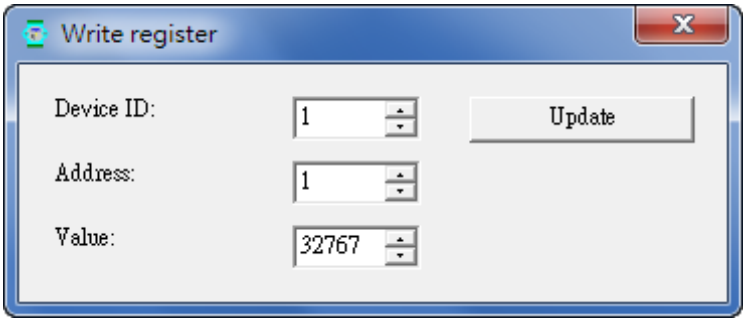






Double click the address can modify the value of coil or register:

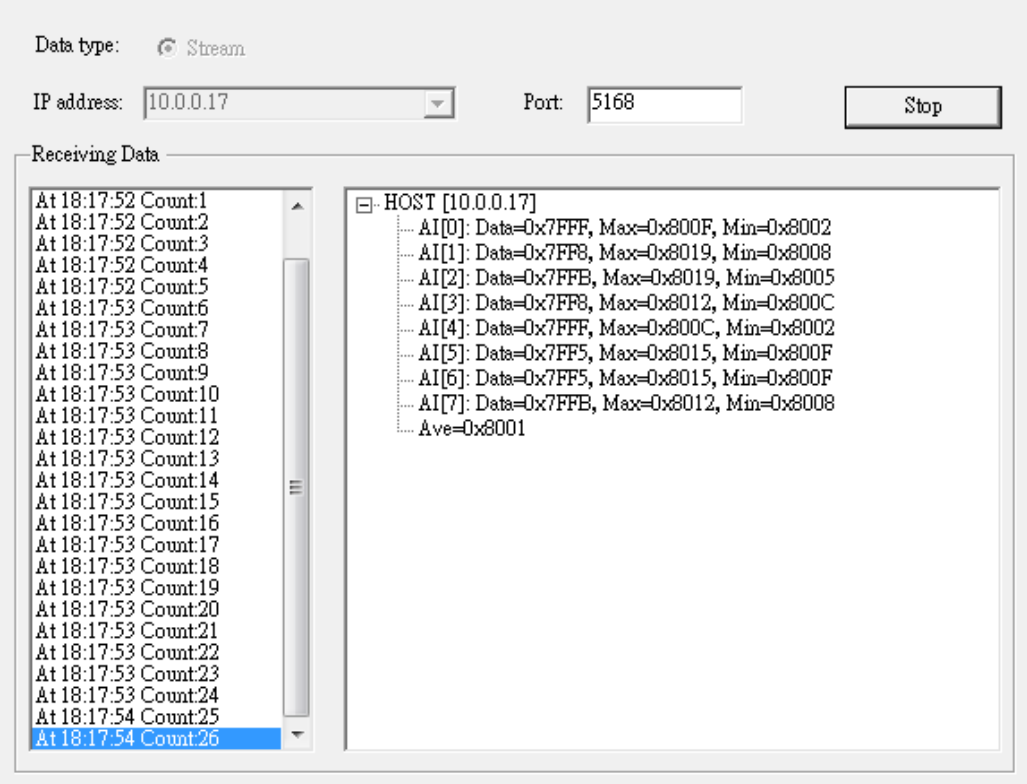




- **Print Screen**
  You can save current ADAM.NET Utility screen into an image file by this option.

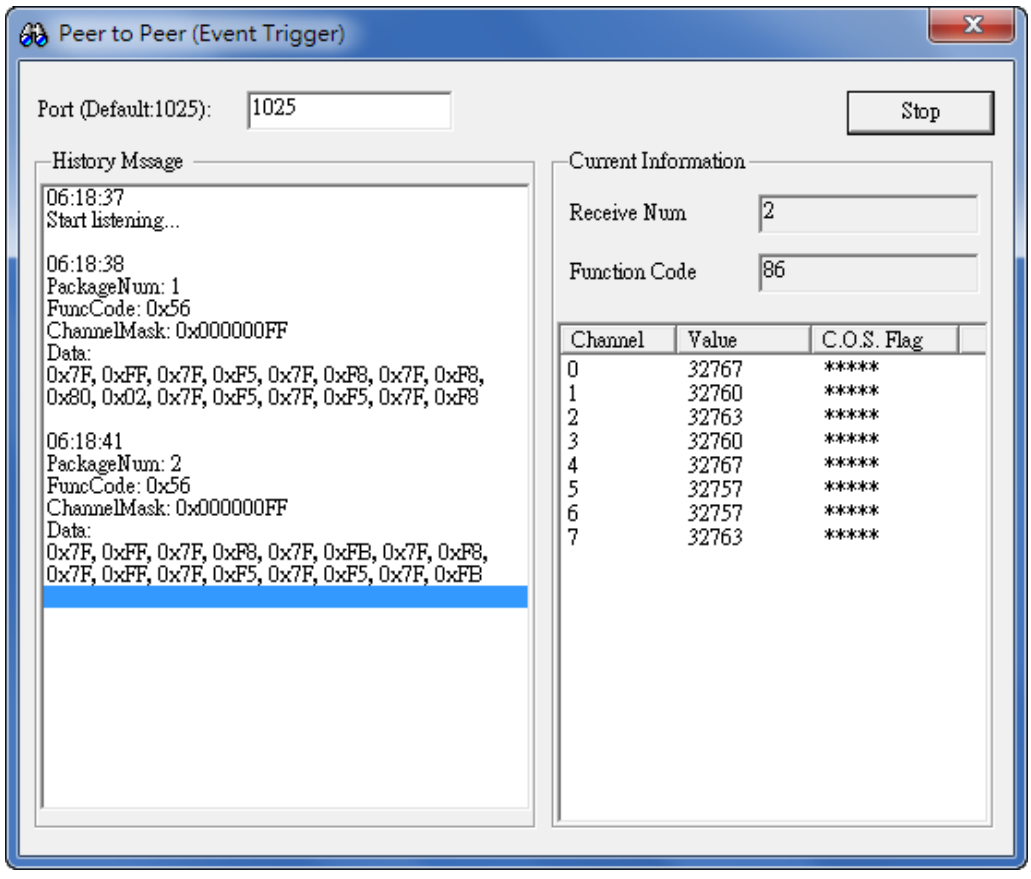
- **Monitor Stream/Event Data**

  This function will indicate real-time I/O data status of one module on your preset host computers/servers. (The Host IP address and data update schedule can be configured in the Stream tab of Status display area.)

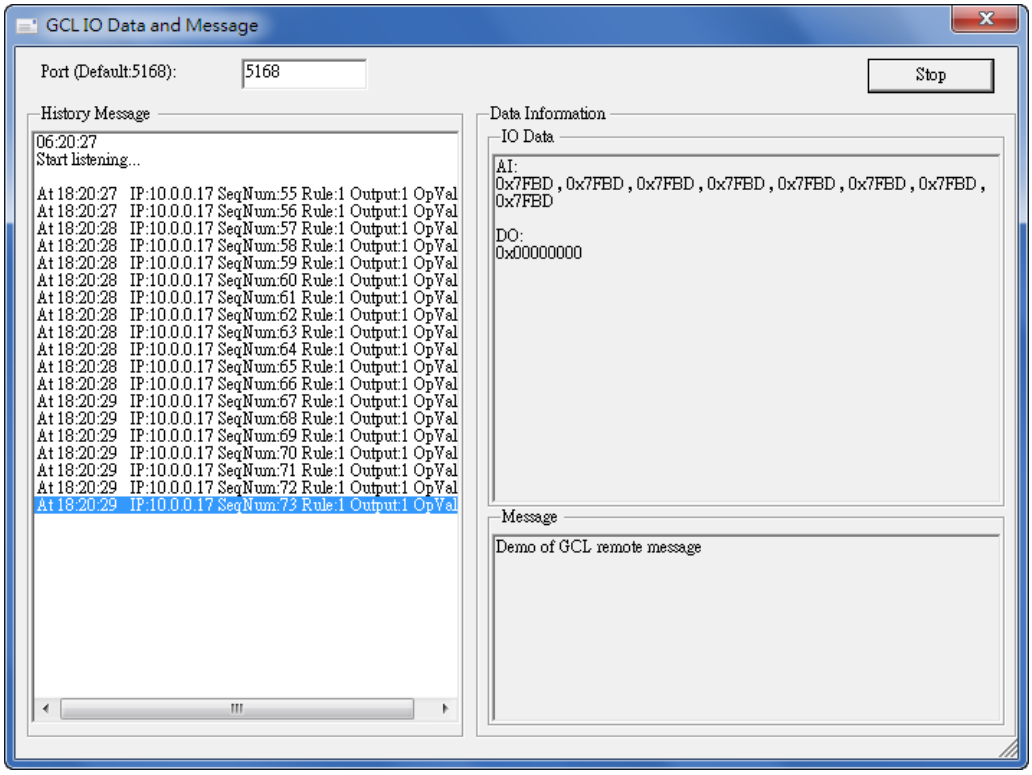


- **Monitor Peer-to-Peer (Event Trigger)**

  This function will indicate the Peer-to-Peer status and historical message when it enabled.



- **Monitor GCL IO Data Message**
  This function will indicate GCL I/O data status when it enabled.



c. Setup

- **Favorite Group**
  You can configure your favorite group including add one new device, modify or delete one current device, sort current devices and diagnose connection to one device.
- **Refresh Serial and Ethernet**
  ADAM.NET utility will refresh the serial and LAN network connection situation.
- **Add COM Ports**
  This option is used to add serial COM ports in ADAM.NET Utility. You won't need to use this option for ADAM-6000/6100/6200 modules.
- **Show TreeView**
  Check this option to display the Module Tree Screen area.
- **Allow Calibration**
  Check this option to allow calibration function enabled on AI/O module.

d. Help

- **Check Up-to-Date on the Web**
  It will automatically connect to support and download page of Advantech web-site when it enabled. You can find and download the latest version of ADAM utility there.
- **About Adam.NET Utility**
  The current version of ADAM.NET Utility is installed on your computer.

4.3.1.2 Toolbar

There are 8 graphical icons for common used options of Menu on the toolbar.



Definition (from left to right)

1. Open favorite group
2. Save favorite group
3. Search Modules
4. Add Devices to Group
5. Terminal for Command Testing
6. Group Configuration
7. Monitor Data Stream/Event
8. Print Screen

4.3.1.3 Module Tree Screen

The Module Tree Screen locates on the left part of ADAM.NET utility operation window. There are four categories in this area:

Serial

All serial I/O Modules (ADAM-4000 and ADAM-5000 RS-485 serial modules) connected to the host PC will be listed in this category.

Ethernet

All Ethernet I/O Modules (ADAM-6000, ADAM-6100, ADAM-6200 and ADAM-5000 TCP modules) connected to the host PC will be listed in this category.

Favorite Group

You can define which devices listed in the three categories above into your personal favorite group. This will make you easier to find your interested modules. Right click on the ADAM device item under the Favorite Group item and you can select Add New Group to create a new group. After you create your own group, right click on your group and Add New Device into your group. You can also select Diagnose connection to check the communication.

ADAM-4500_5510 Series

This is a DOS interface utility for remote controllers such as ADAM-4500 and ADAM-5510 series.

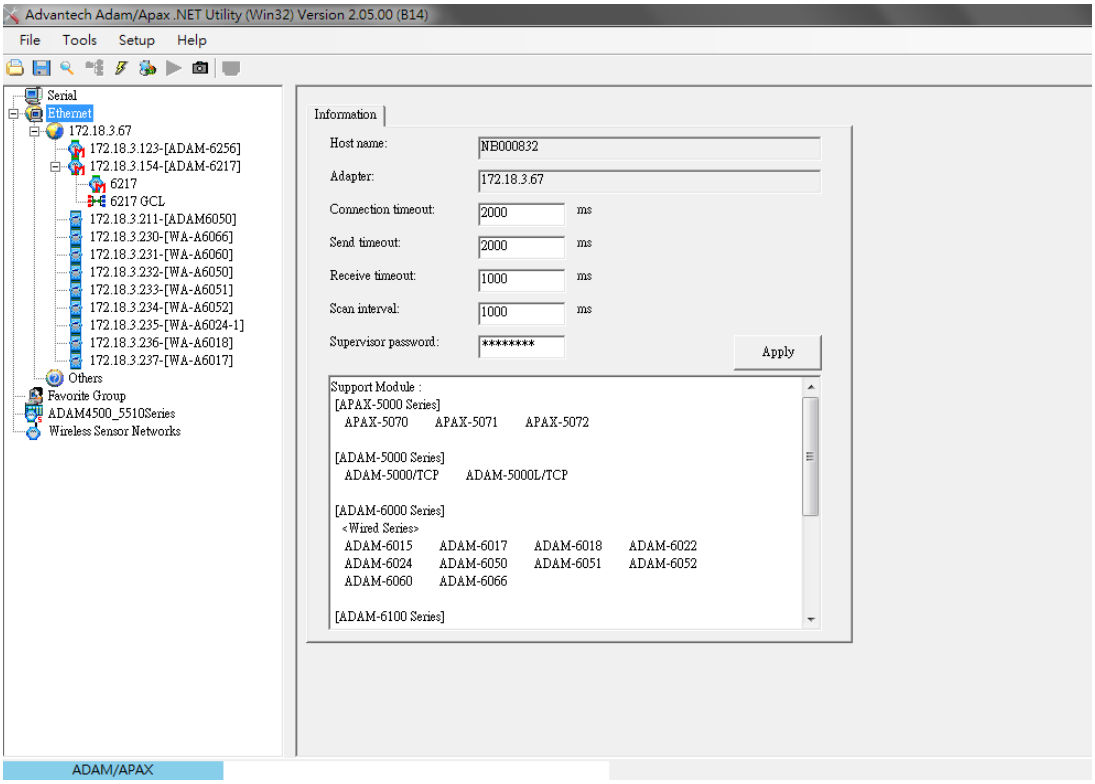
Wireless Sensor Networks

All wireless I/O Modules (ADAM-2000 modules) connected to the host PC, through wireless gateway, will be listed in this category.

4.3.1.4 Main Operation Screen

Main Operation Screen located on the right side of utility includes I/O status display and function setting. You can select different items in Module Tree Screen, and then Main Operation Screen will change dependently. You can do all configurations and test in this area.

In Information page (after clicking Ethernet), you can configure Connection/Send/ Receive/Scan Timeout. The supervisor password is a shortcut to let you enter a password at one time which's applied for certain modules, so you don't need to enter the same password for each module when you check it.



4.3.2 General Configuration

4.3.2.1 Information

It indicates basic information of ADAM-6200 module.



Firmware Version

Indicates the current Firmware version of ADAM-6200 module.

Device Name

Means model name of ADAM-6200 module. You also can rename it for recognition if required.

Device Description

You can add comments on this module for recognition.

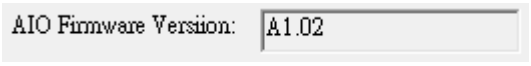
Locate

It can help user search ADAM module with light sign. (Status LED will be constantly on for 10 sec when it enabled.)

ADAM Web Page

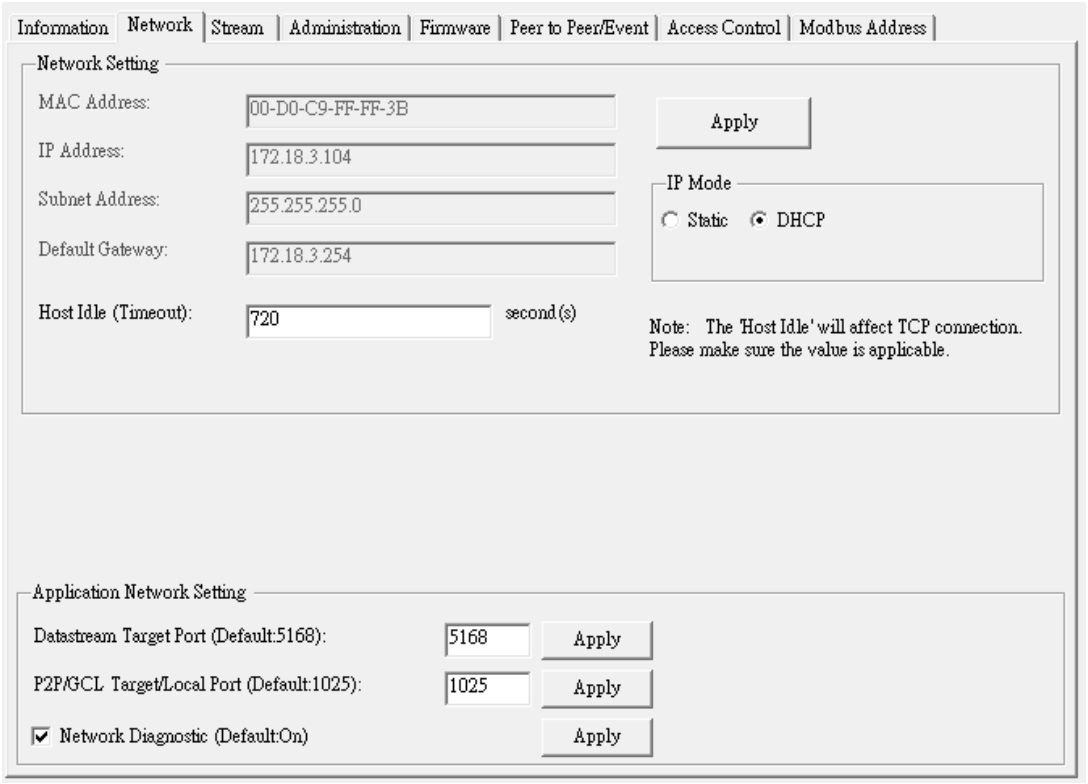
It allows users to access Web page of ADAM module by QR code application software (APP) on smart phone/pad.

Note! *For all AI/O modules, its AIO firmware version will also show in this page.*



4.3.2.2 Network

You can set up required network connection on this page.



Network Setting

You can select the Connection mode as DHCP or Static IP and configure the MAC address, IP address, Subnet address, Default gateway and Host Idle (timeout).

Port Setting

You can set up the network port for DataStream and P2P/GCL function.

Note!

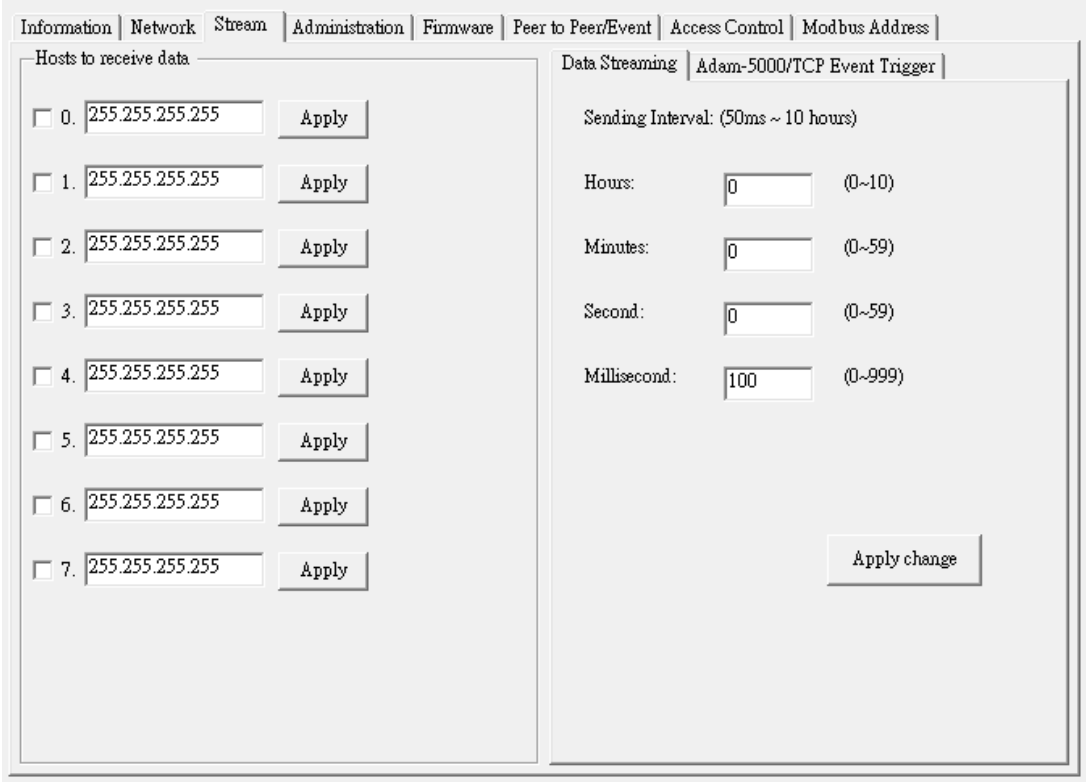
1. *Static IP mode is set 10.0.0.1 as default.*
2. *Network Diagnostic (Default: ON) - ADAM module will actively monitor and diagnose the Ethernet switch periodically when it's enabled. However, if you have no need to use Ethernet ports for communication, please notice to disable this function.*

Module Ethernet Protocol and Port

TCP Modbus	TCP	502	
Download	TCP	5450	
Config. Upload	TCP	5452	
Web Server	TCP	80	
Search Engine	UDP	5048	
ASCII CMD	UDP	1025	
P2P/GCL	UDP	1025	Configurable
Data Stream	UDP	5168	
GCL remote Message	UDP	5168	

4.3.2.3 Stream

ADAM-6200 modules can actively send its data to multiple hosts periodically. It is called Data Stream. In this tab, you can define IP address of the hosts receiving the data transferred by ADAM-6200 modules, as well as the period how often ADAM-6200 modules will send data to the hosts.



Host IP address

There're up to 8 sets of Host IP address for you to run data streaming.

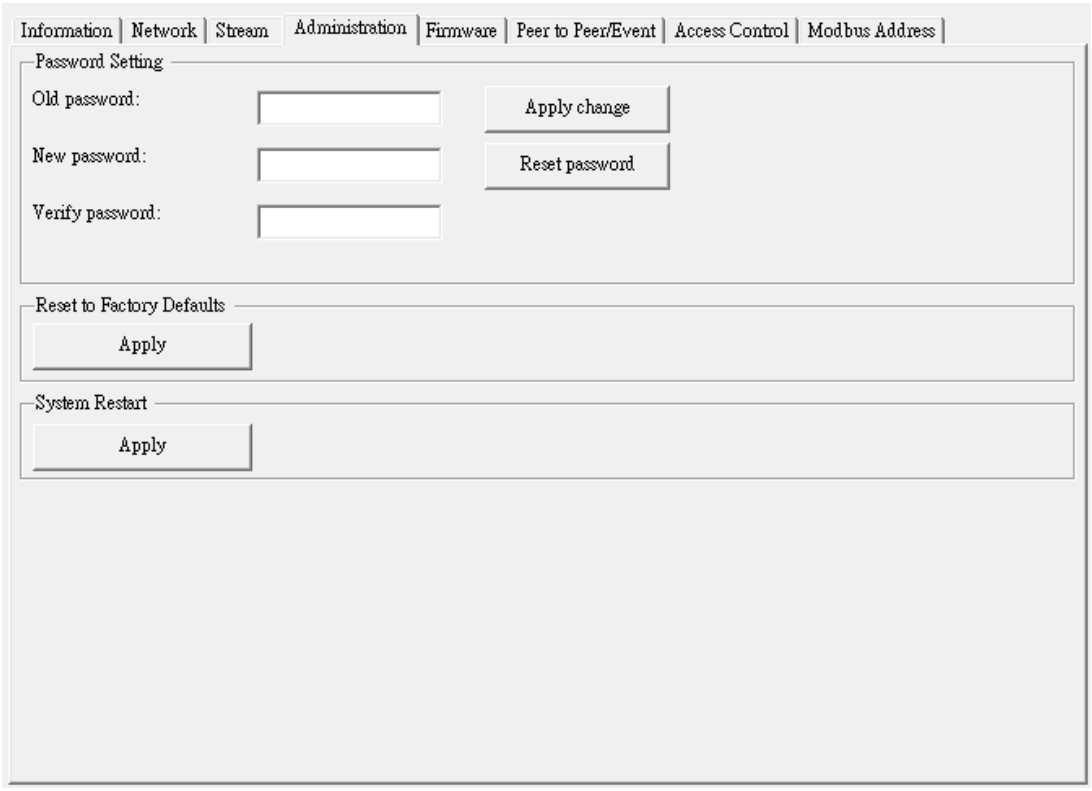
Data Streaming

It used to set up the data streaming schedule.

Note! *Set the period by Data Streaming tab at right for ADAM-6200 streaming function. ADAM-5000/TCP Event Trigger tab is for ADAM-5000.*

4.3.2.4 Administration

The administration page includes Password, reset, and restart function.



Password

You can enter the password on this page to keep the module from losing unknown control.

Note! *The default password is 00000000*

Reset to factory default

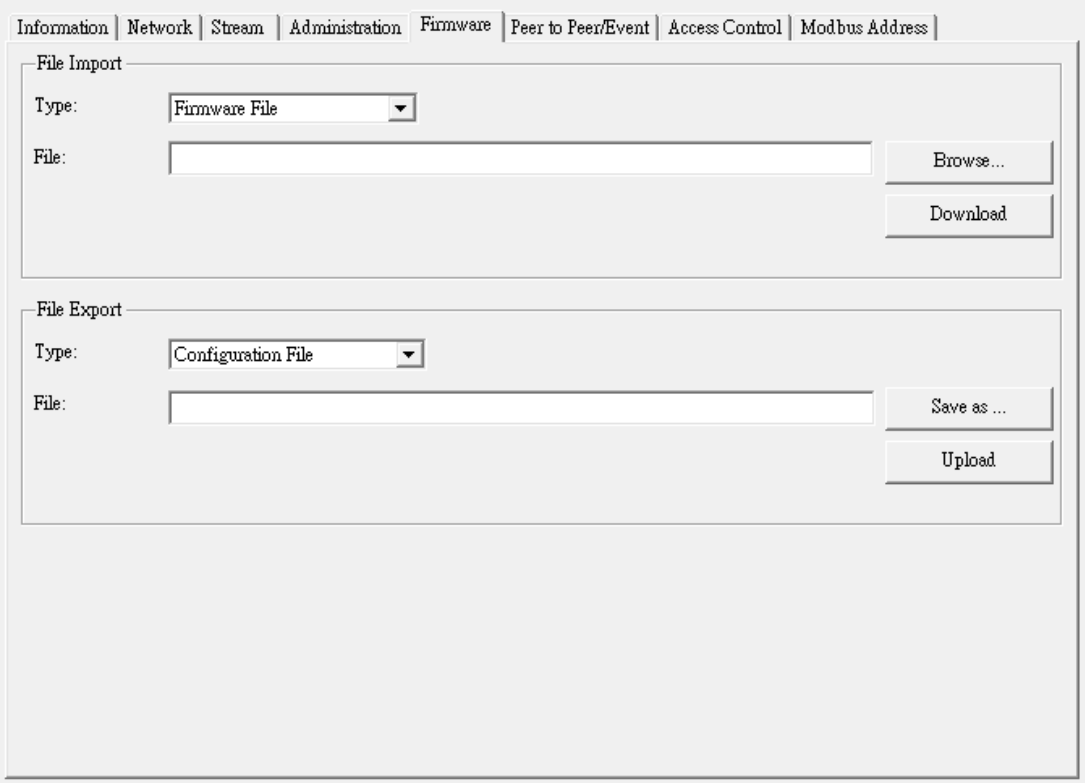
The system configuration of ADAM-6200 module will be clear and restored to factory default when it enabled.

Restart the module

The system of ADAM-6200 module will reboot when it enabled.

4.3.2.5 Firmware

On this page, you can import and export your configuration file and upgrade firmware.



File Import

There are four file types including firmware, JAR, HTML, Modbus information (described as below) for you to import from the host PC.

Firmware -

It means system firmware, you can easily perform firmware upgrade on this page, but for AI/O module, it will additionally need to involve AI/O firmware if changed.

JAR file -

It's used for the extended application on the Webpage.

HTML file -

It's used for the extended application on the Webpage.

Module information -

It presents the configuration file of module in Utility.

Note!

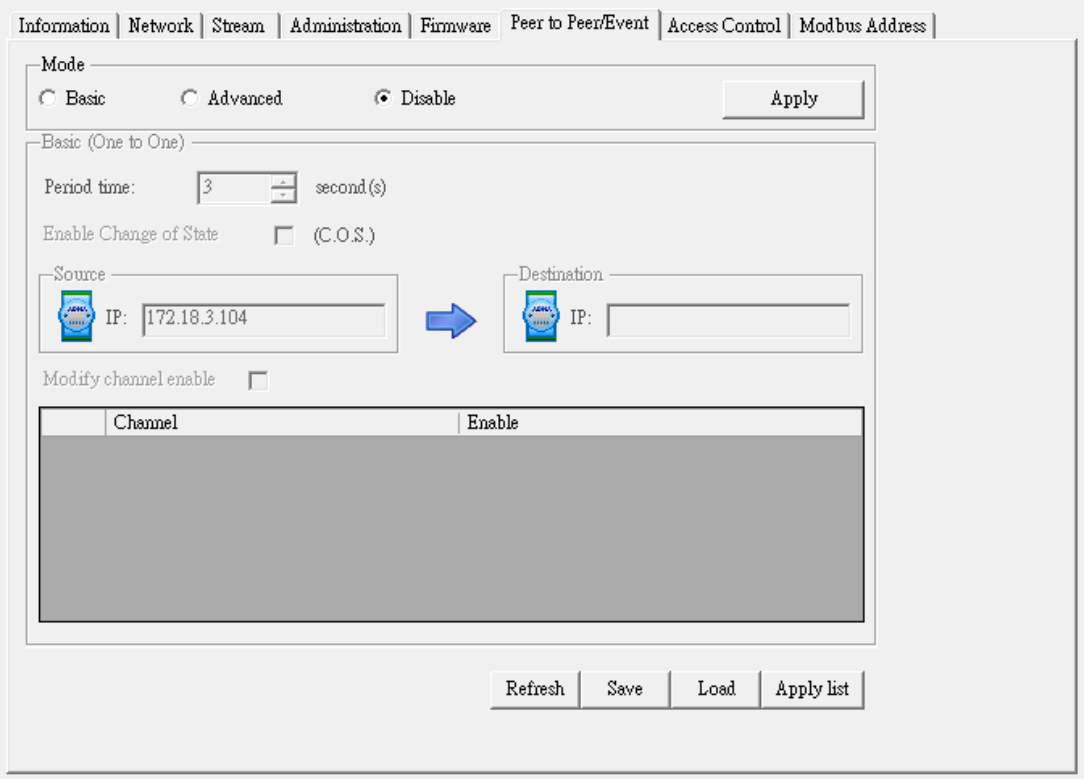
1. *To keep your module system up-to-date, please find the latest firmware version on Advantech's website.*
2. *Which Web browser supports HTML 5: IE 9, Chrome, Firefox.*
3. *Please do not remove the power of your module when group configuration function is processing. Otherwise, the module system will probably crash.*

File Export

Alternatively, you also can export configuration file to your host PC.

4.3.2.6 Peer to Peer / Event

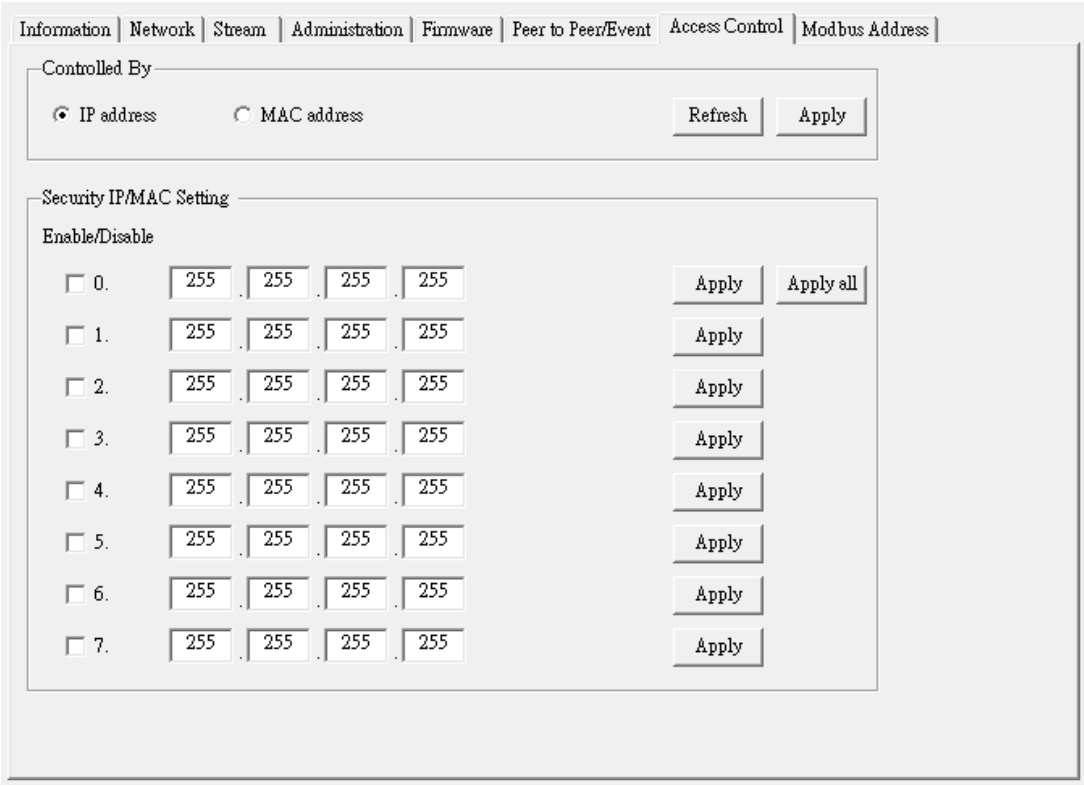
It will be described in following sections.



4.3.2.7 Access Control

For avoiding from unauthorized access, you can manage which host PC or device has been permitted to remotely control ADAM-6200 module by IP or MAC Address.

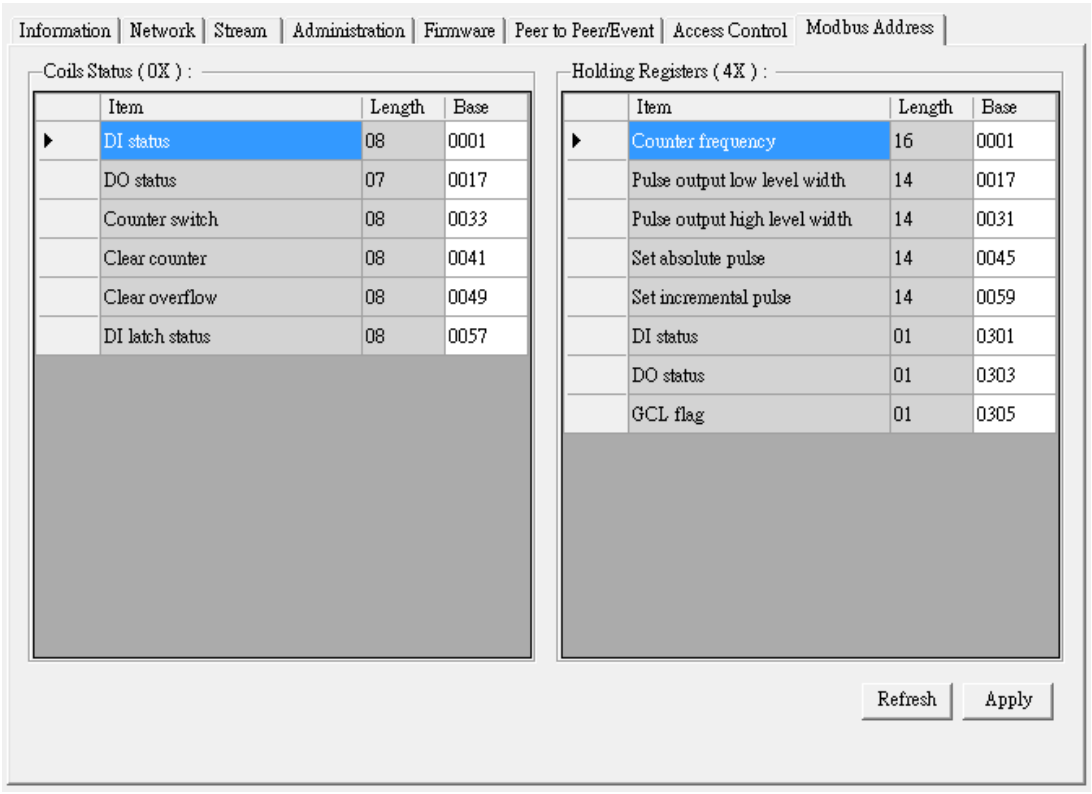
1. Select which type you're going to configure.
2. Enter the IP or MAC address of Host controller.



4.3.2.8 User-defined Modbus Address

In order to provide user with more flexible and scalable in deploying ADAM module, ADAM-6200 module remove the limitation of Modbus address setting and make it configurable as user's actual need. Basically, there're two kinds of Modbus address section (0X and 4X) for you to configure each function item.

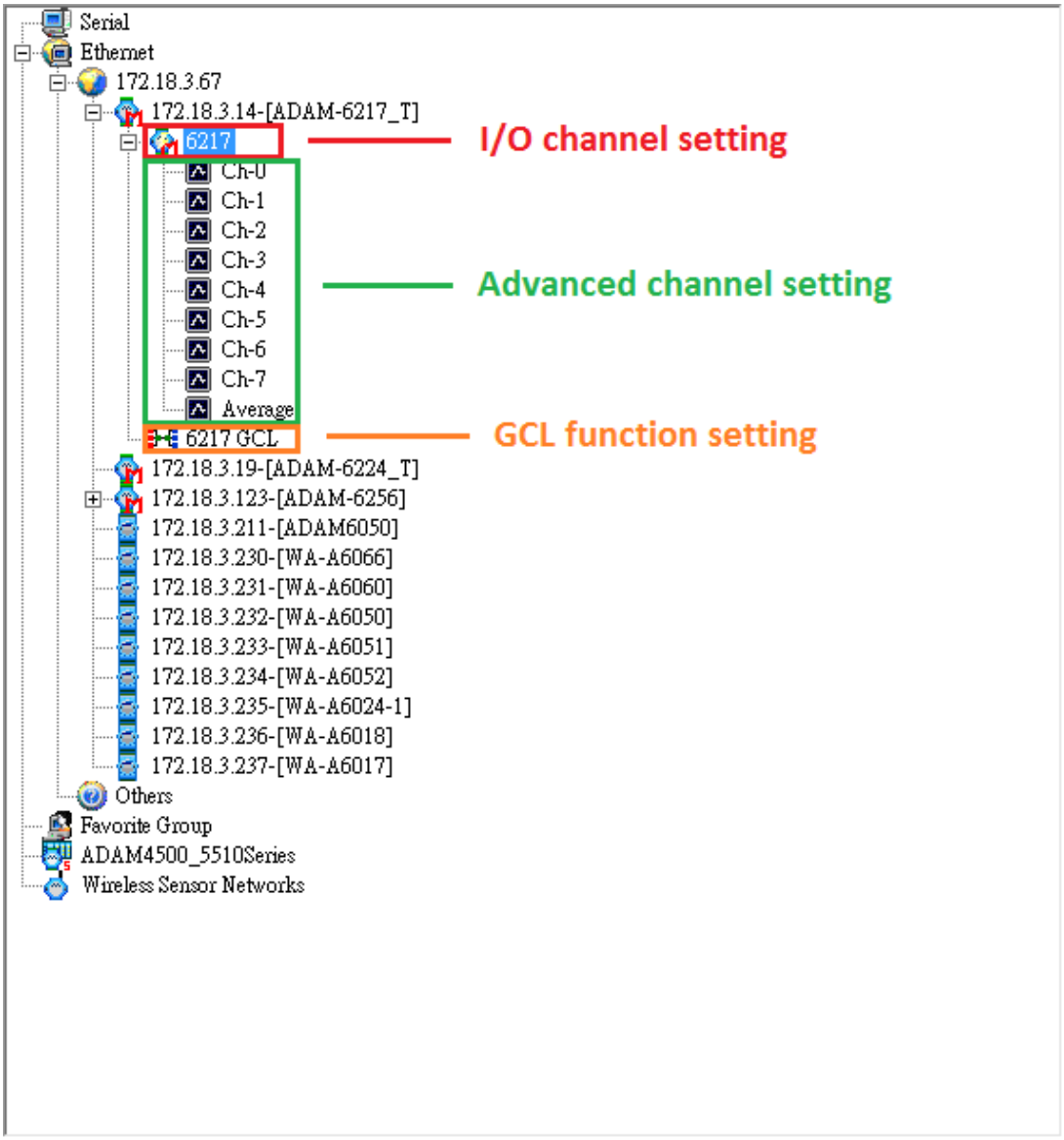
For example, the below screenshot is Modbus address setting page of ADAM-6250.



4.3.3 I/O Configuration

After the introduction of ADAM-6200 general configuration in previous section, you're realized how to operate with several key functions in your application. Furthermore, you will learn the way to set up all I/O channel configuration in details such as channel range, calibration and alarm.

You can select one of ADAM-6200 module you want to configure and spread its channel list by entering the password in the Module Tree screen. As below figure, it will be divided into three parts, I/O channel overview, advanced channel setting and GCL function setting.



I/O channel overview

You will see all I/O channel status and output enable setting.

Advanced channel setting

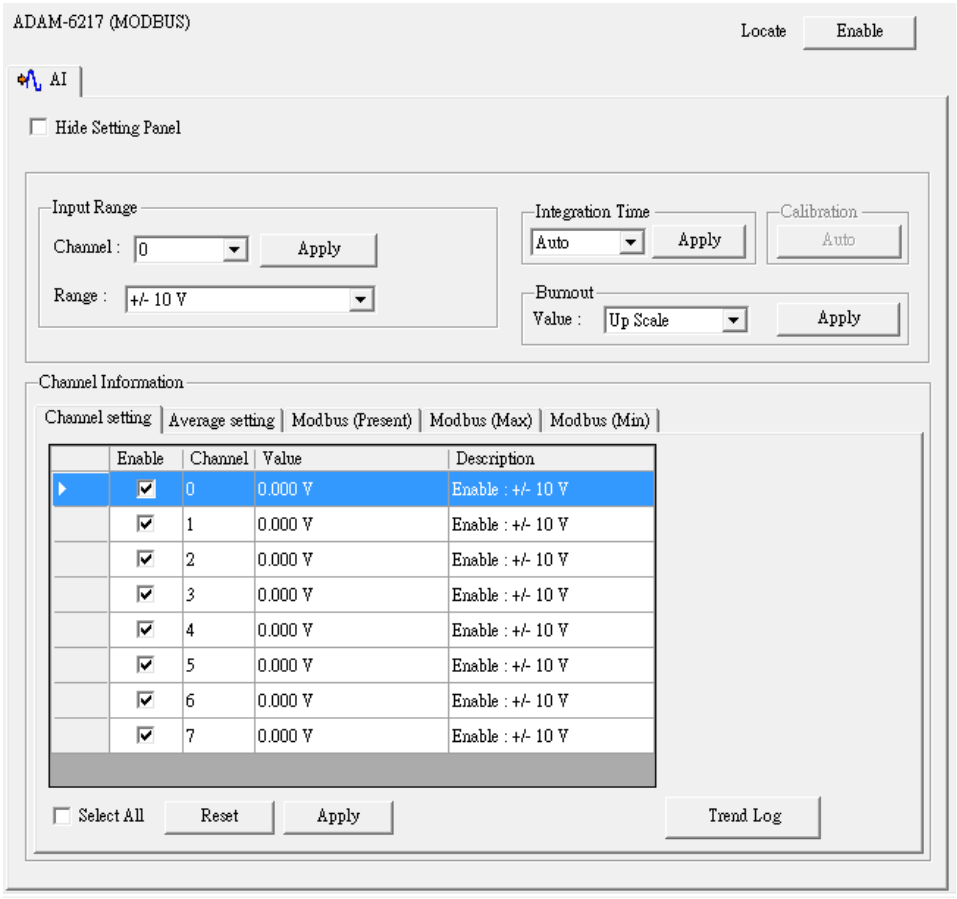
You will be able to specifically configure single channel for details.

GCL function setting

It will be introduced in the following section.

4.3.3.1 ADAM-6217

Analog Input Channel Overview



Locate

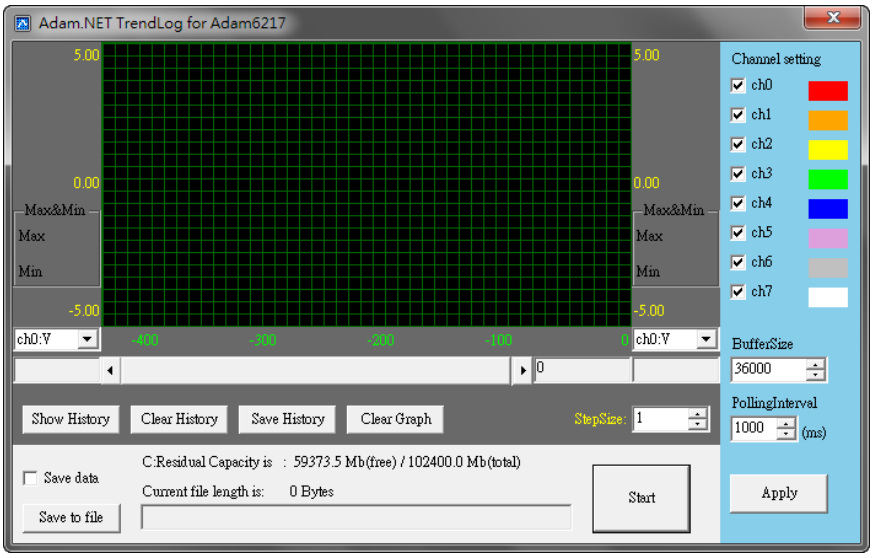
It can help user search ADAM module with light sign. (Status LED will be on for 10 sec when it enabled.)

Hide Setting Panel

The details of channel setting will be hidden when you check on the Hide Setting Panel checkbox.

Trend Log

When you click the trend log button, you will see the real time input channel status (after I/O configuration is done) in pop-up window. In which, you also can check, operate, even save the log as your actual need.

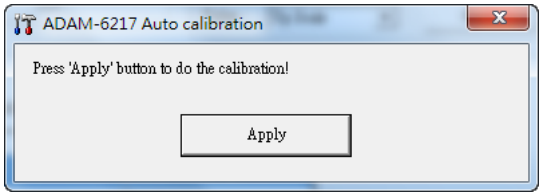


Input Range

You can select input channel and its voltage/current range and click Apply button to take effect.

Calibration

It's for sure that each module is well calibrated when it's in manufacturing process. However, in certain applications, user may need to redo the calibration with their device. If you want to enable this, please go to Setup menu and click Allow Calibration first, then click Auto button to open the calibration page. The calibration process will automatically run without any input signal in around 20 seconds, after that, please click apply button to finish the process.



Integration Time

In order to remove the DC noise from the power supply, there are two kinds of frequency built-in analog input modules to filter it. You can select and apply this feature here as your request.

Burnout

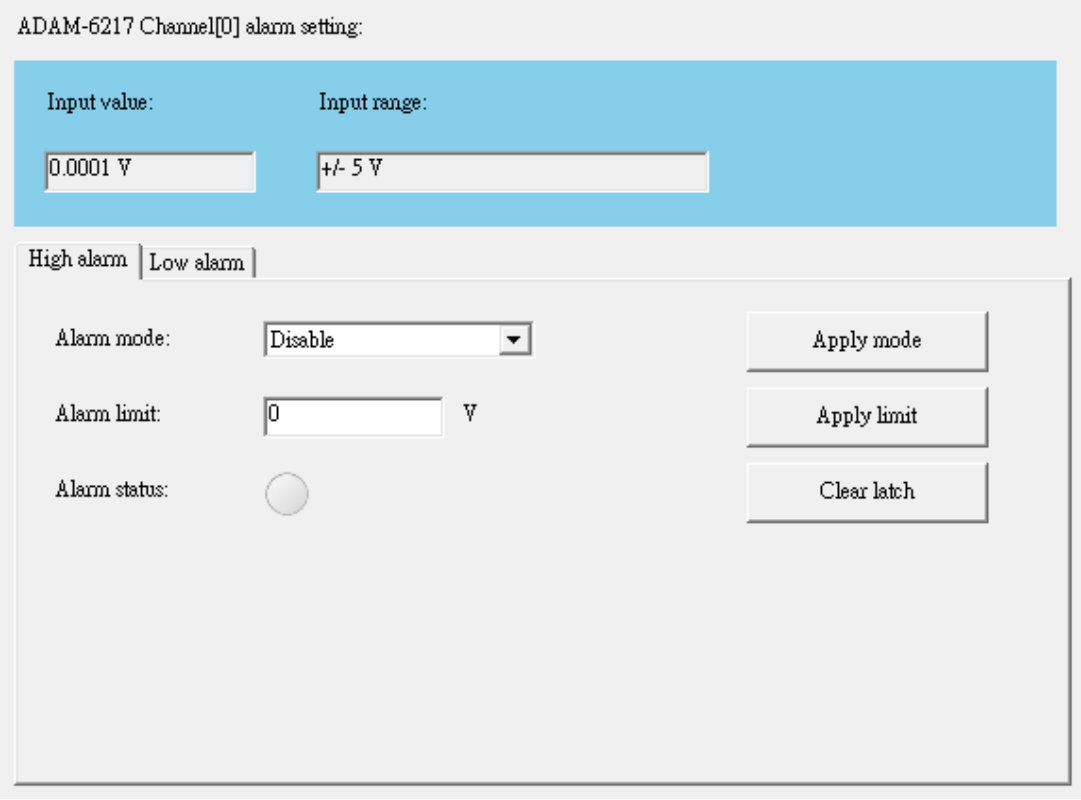
Presently, it's only support for 4 - 20mA input range. If you select up scale, you will see the value FFFF in Modbus address when open circuit happens. Otherwise, it will show 0000 as down scale.

Channel Information

- Overview
  It presents which channel is enabled and its current input value.
- Average setting
  You can decide which channel will be selected to perform average calculation and it will show on right side
- Modbus (Present)
  You can see current analog input value in decimal, hexadecimal, engineer unit, and description for all related Modbus address.
- Modbus (Max)
  You can see historical maximum analog input value presented as decimal, hexadecimal, and engineer unit for all related Modbus address. If you need to recalculate the past maximum value, please click the reset button for each channel to clear the record.
- Modbus (Min)
  You can see historical minimum analog input value presented as decimal, hexadecimal, and engineer unit for all related Modbus address. If you need to recalculate the past minimum value, please click the reset button for each channel to clear the record.

Advanced channel setting

In advanced channel setting, you can easily monitor the present input value/range of each channel, and separately configure the details of channel such as high/low alarm mode, limit and so on.



Alarm Setting

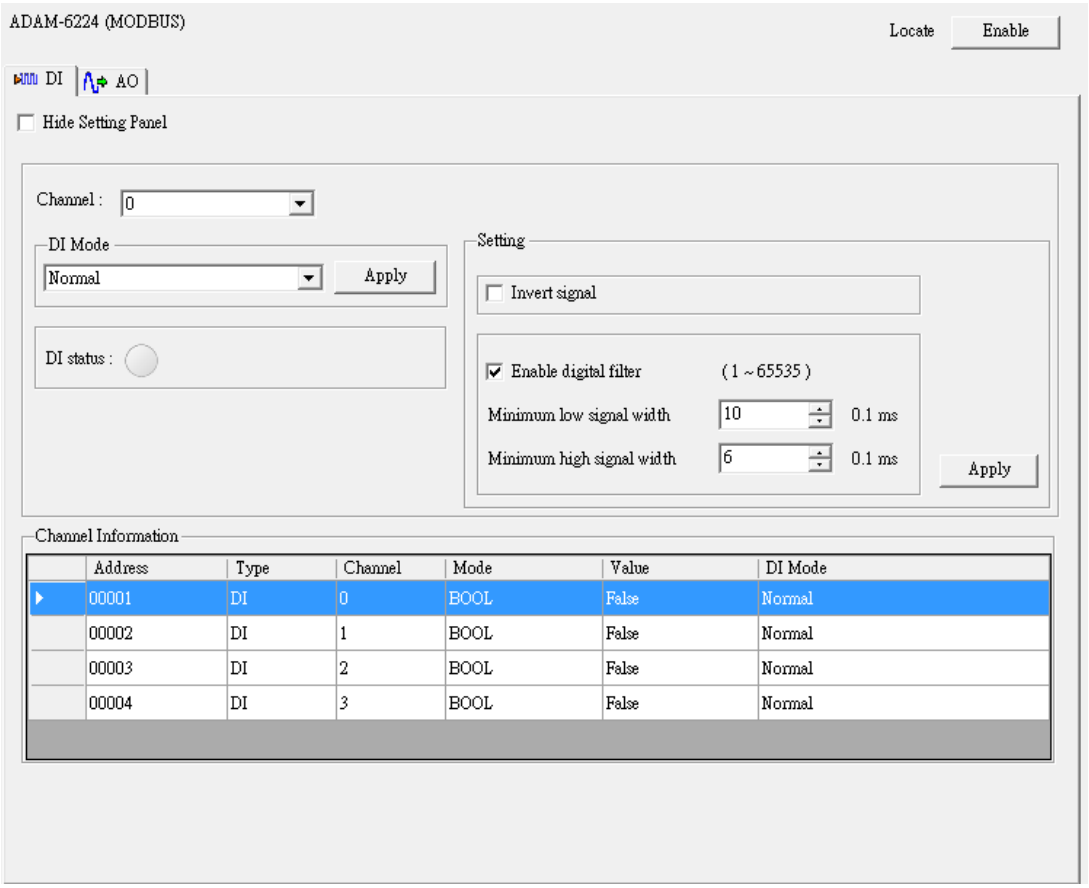
When the analog input value is higher than the high alarm value, or lower than the low alarm value, the alarm condition will be valid. Then the alarm status register will be pulled to logic high and **alarm status** LED will be lit up on the main screen of utility.

There are three alarm modes. You can select the alarm mode by the Alarm mode combo box for the low alarm and high alarm respectively and enter the value as **Alarm limit**.

1. Disable: Alarm is disabled.
2. Latch: Once the alarm condition is valid, the alarm status will be activated to logic high level and will keep the value and alarm LED will also on unless the alarm is clear by clicking the Clear latch button.
3. Momentary: The alarm status will dynamically change according to the alarm condition occurs. When the alarm status is set as logic high, it means alarm condition is valid. Otherwise, the alarm status register will be logic low. Alarm status LED will be on/off so.

4.3.3.2 ADAM-6224

- **Digital Input Channel Overview**



Locate

It can help user search ADAM module with light sign. (Status LED will be on for 10 sec when it enabled.)

Hide Setting Panel

The details of channel setting will be hidden when you check on the Hide Setting Panel checkbox.

DI mode

a. Normal
When DI channel receives input signal, AO will send the value you defined.

b. True trigger to startup value
When the received logic result of DI is true, AO will send the startup value you defined.

c. False trigger to startup value
When the received logic result of DI is false, AO will send the startup value you defined.

d. True trigger to safety value
When the received logic result of DI is true, AO will send the safety value you defined.

e. False trigger to safety value
When the received logic result of DI is false, AO will send the safety value you defined.

Invert signal

It will invert the logic state of input signal. In another words, when no signal pulse come into DI channel, the DI status will be logic high ("1")

DI status LED

When DI channel is activated, it will light up.

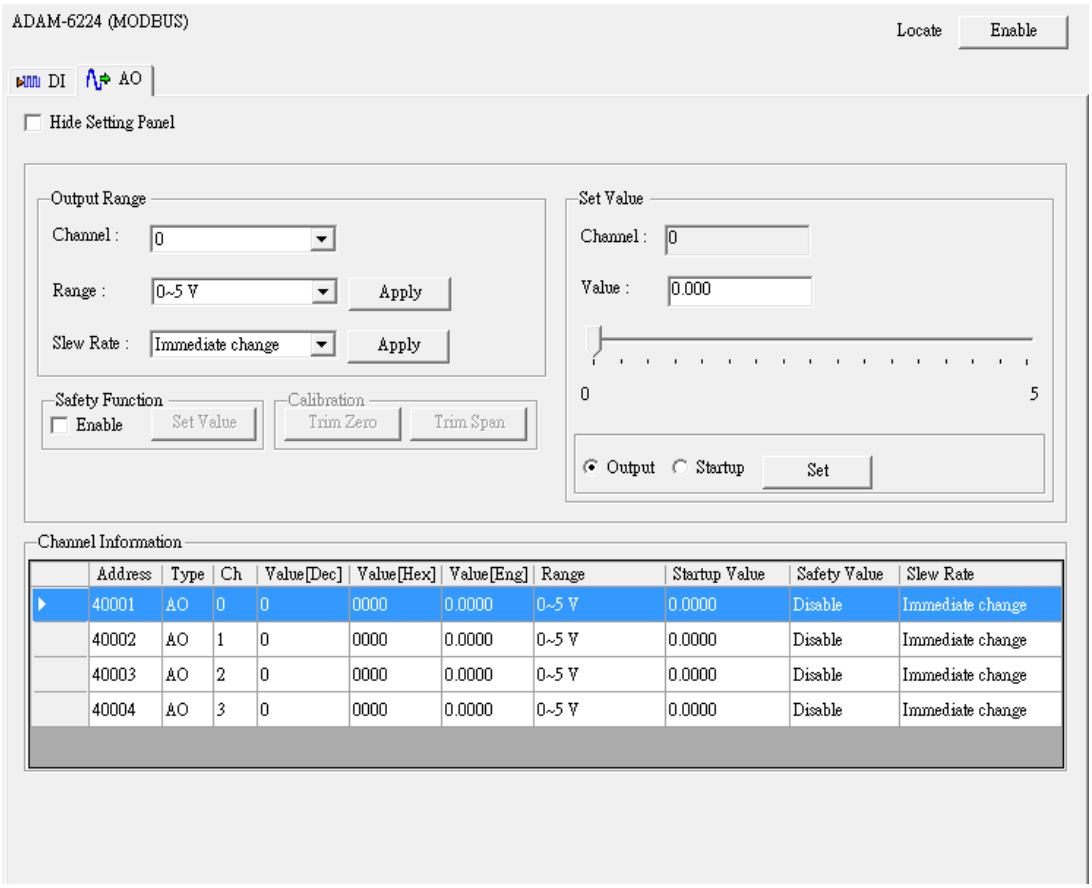
Digital filter

It contains minimum high signal width and minimum low signal width (1-65535) for filtering the noise.

Channel Information

The information including address, type, channel, mode, value, and DI mode will show on the screen.

■ Analog Output Channel Overview



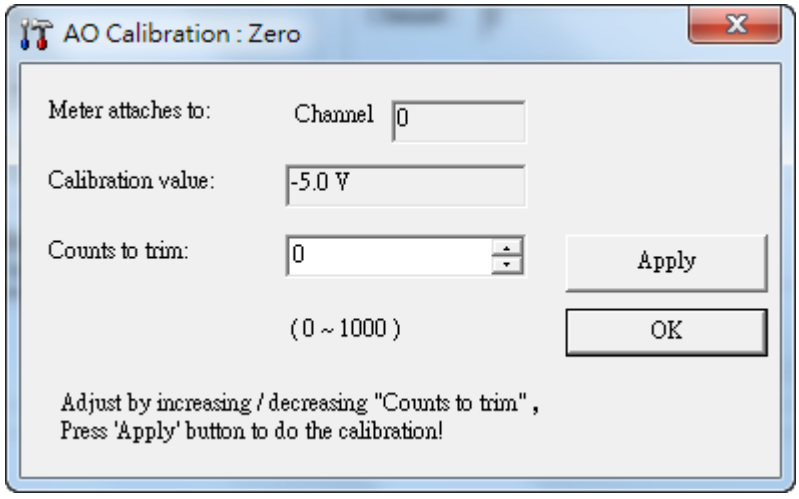
Output Range

You can select input channel and its voltage/current range and click Apply button to take effect.

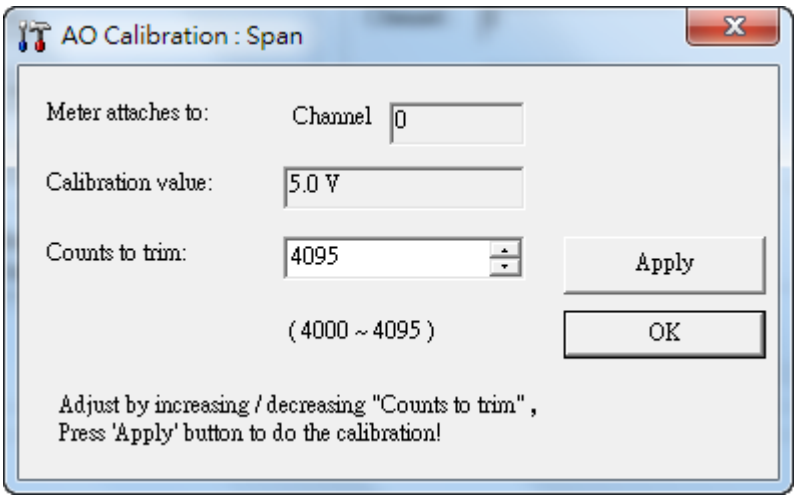
Calibration

It's for sure that each module is well calibrated when it's in manufacturing process. However, in certain applications, user may need to redo the calibration with their device. If you want to enable this, please go to Setup menu and click Allow Calibration first.

1. Trim Zero - It's used to help user calibrate the low limit of output range by entering the value.
   Step 1. Click the Trim Zero button after enabled Allow Calibration.
   Step 2. The output signal will be delivered as the lowest level, for example, if you select +/- 10V range, you will get around -10V.
   Step 3. Please check if the voltage/current meter connected to the module is right to the lowest level. If not, please adjust "Counts to trim" to correct the output value indicated on the meter. After calibrating to correct value, please click Apply button to take it effect.
   Step 4. Click OK button to finish the process.



2. Trim Span - It's used to help user calibrate the high limit of output range by entering the value.
   Step 1. Click the Trim Span button after enabled Allow Calibration.
   Step 2. The output signal will be delivered as the highest level, for example, if you select +/- 10V range, you will get around 10V.
   Step 3. Please check if the voltage/current meter connected to the module is right to the highest level. If not, please adjust "Counts to trim" to correct the output value indicated on the meter. After calibrating to correct value, please click Apply button to take it effect.
   Step 4. Click OK button to finish the process.



Slew Rate

It defines how long the AO value changes according to the type of output range.

Set Value

You can set value by scrolling the meter bar.

Safety function

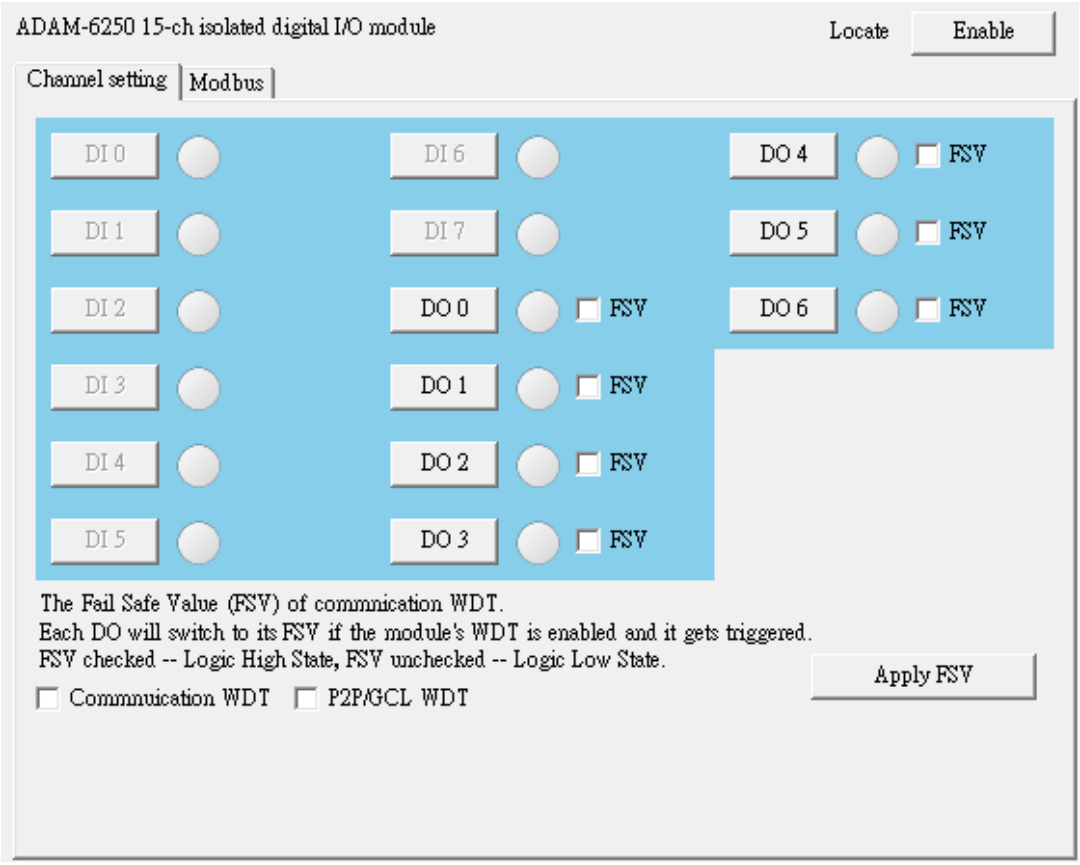
You can define the safety value after it's enabled.

Channel Information

The information including address, type, channel, value [Decimal], value [Hexadecimal], value [Engineering], output range, startup value, safety value, and slew rate will show on the screen.

4.3.3.3 ADAM-6250

- **Digital Input Channel Overview**



Locate

It can help user search ADAM module with light sign. (Status LED will be on for 10 sec when it enabled.)

Channel Setting

You can see value of all digital input channels by related LED display in this tab. Besides, you also can control values of all digital output channels by related button. The LED next to the button will display current value of that digital output channel.

The Fail Safe Value (FSV) of communication WDT

If FSV checkbox is enabled, it presents the module will set the output channel to logic high when WDT timeout. There are two application as below. After all, click the Apply FSV button to take it effect.

a. Communication WDT
When the module hasn't received any TCP network packet from client too long, such as SCADA or Web browser. That means if the waiting time is greater than host idle time, the module will automatically set the safety value to output.

b. P2P/GCL WDT
When the module hasn't received P2P/GCL network packets too long, that means the waiting time is greater than idle time you entered, the module will automatically send the safety value to host PC if you've enabled this function.

Modbus

It indicates the overview of the status (value) of its corresponding Modbus address, channel and mode.

Advanced channel setting

1. DI configuration

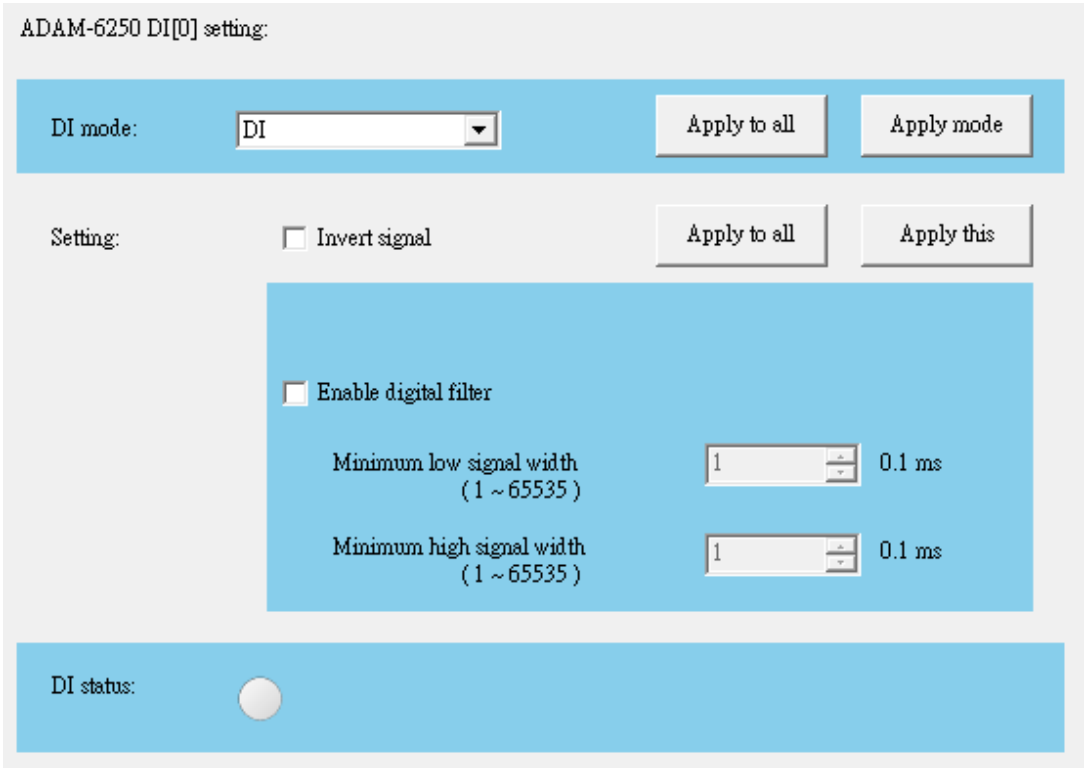
In ADAM-6250 advanced channel setting, you can configure the details of input signal such as enable/disable invert signal, digital filter based on different DI mode and meanwhile, you also can check DI status of each channel. There are five kinds of DI mode including DI, Counter, Low to High Latch, High to Low Latch, and Frequency.

DI mode

a. Invert signal - It will invert the logic state of input signal. In another words, when no signal pulse come into DI channel, the DI status will be logic high (“1”)

b. Digital filter - It contains minimum high signal width and minimum low signal width (1-65535) for filtering the noise.

c. DI status - It will light up when logic high.



Counter mode

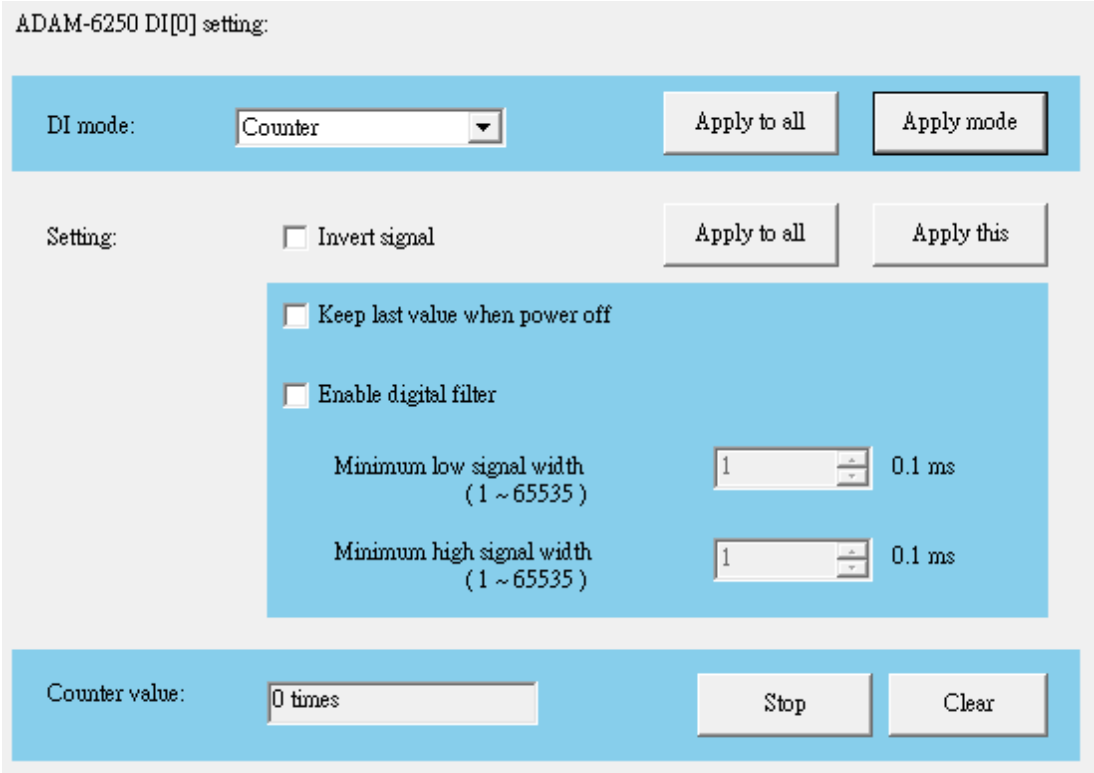
When you choose Counter mode, one counter will count the pulse number of the digital signal from the selected channel, and then record the count number in the register. At the bottom of the screen as below, present count value of the selected channel is displayed by the Counter value text box. You can start or stop the counter to count by clicking the Star/Stop button next to the Counter value text box. You also can reset the counter (the value in the register will be initialized to zero) by clicking the Clear button.

a. Invert signal - It will invert the logic state of input signal. In another words, when no signal pulse come into DI channel, the DI status will be logic high (“1”)

b. Keep last value when power off - It can store the last counter value in the register when the module power off.

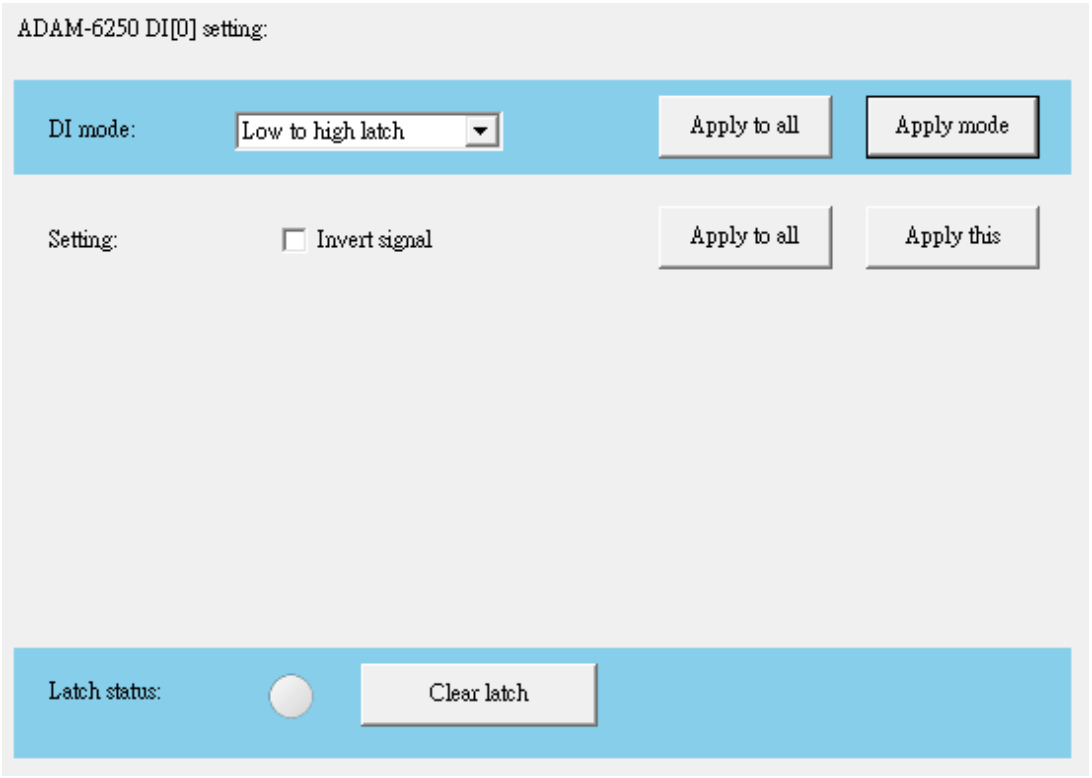
c. Digital filter - It contains minimum high signal width and minimum low signal width (1-65535) for filtering the noise.

d. Counter value - It will continuously display the present counter value unless it's stopped.



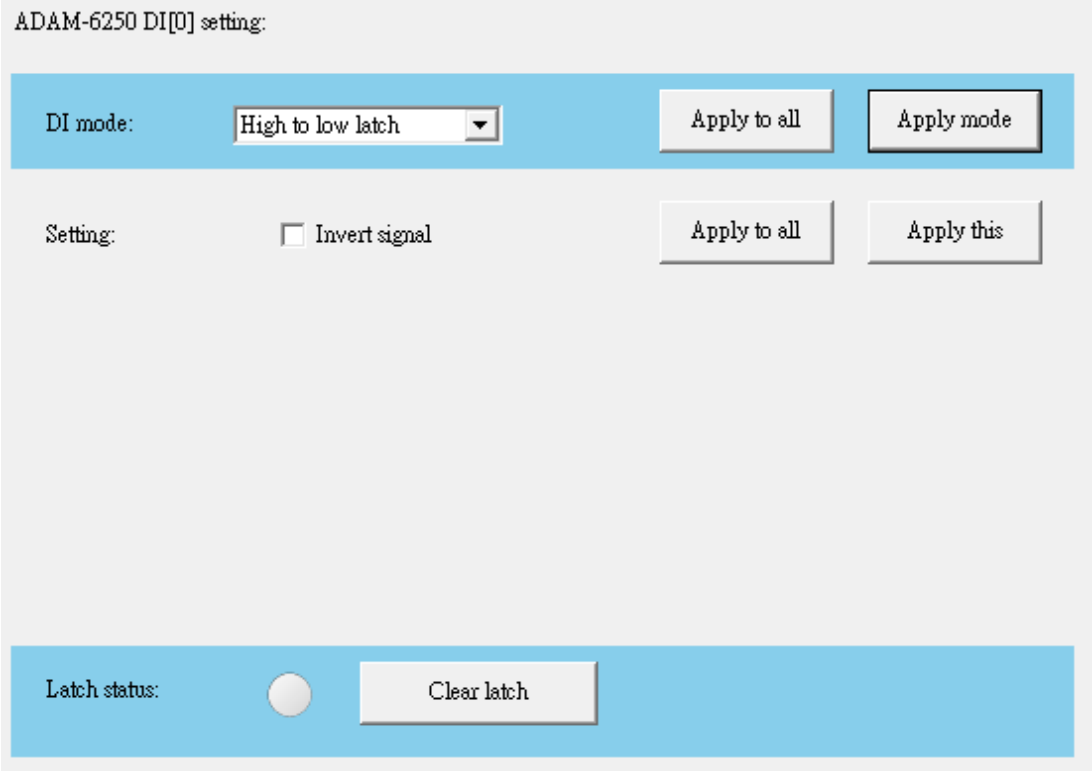
Low to High Latch mode

When you choose Low to High Latch mode, once the digital input channel detects logic level changes from low to high, the logic status will be keep as logic high. The logic status will remain the logic high, until you clear latch manually. Then the logic status will back to logic low. The logic status can be seen by the Latch status LED display. If you want to clear latch, you can do it by clicking the Clear latch button. As for Invert signal setting, it's the same as mentioned above.



High to Low Latch mode

When you choose High to Low Latch mode, once the digital input channel detects logic level changes from high to low, the logic status will be keep as logic High. The logic status will remain the logic High, until you clear latch manually. Then the logic status will back to logic Low. The logic status can be seen by the Latch status LED display. If you want to clear latch, you can do it by clicking the Clear latch button. As for Invert signal setting, it's the same as mentioned above.



Frequency mode

When you choose Frequency mode, ADAM-6200 digital module will calculate the frequency value of the digital input signal from the selected channel and the frequency value will be displayed by the Frequency value text box.



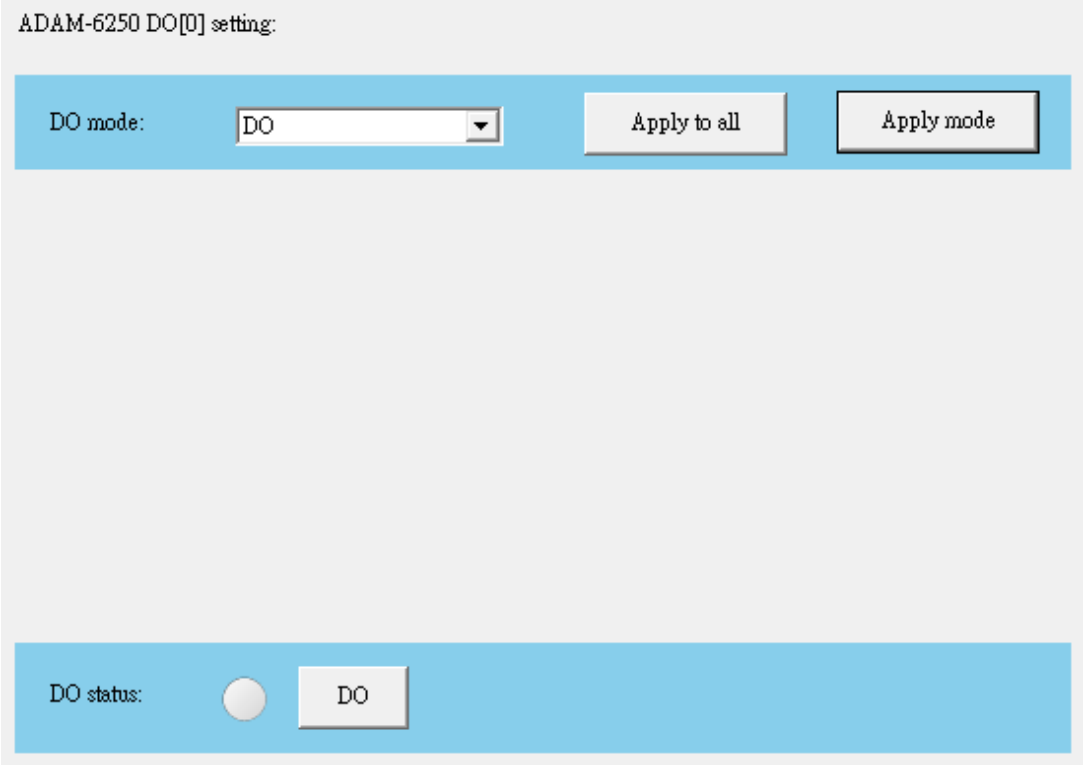
5. DO configuration

In ADAM-6250 advanced channel setting, you can configure the details of output signal such as signal width, frequency, duty cycle based on different DO mode and meanwhile, you also can check DO status of each channel.

There are four kinds of DO mode including DO, Pulse out, Low to high delay, and High to low delay.

DO mode

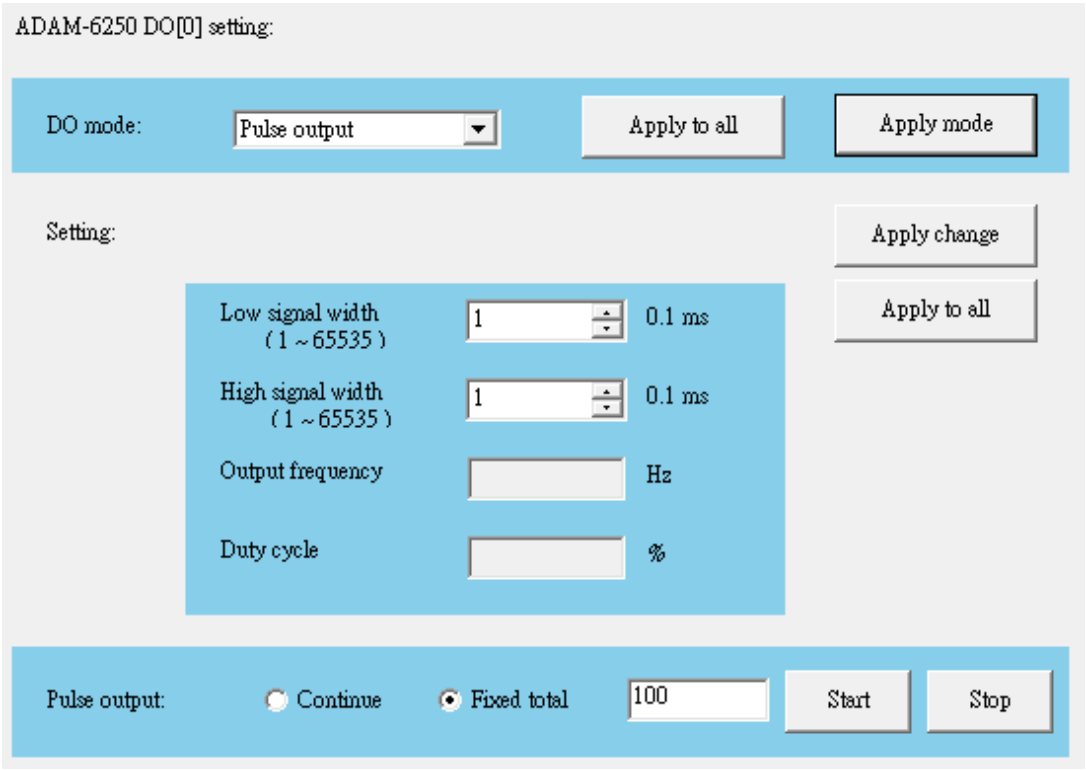
When you choose DO mode, you can control the digital output value of the selected channel by the DO button. The current digital output value will be shown by the DO status LED display.



Pulse Output mode

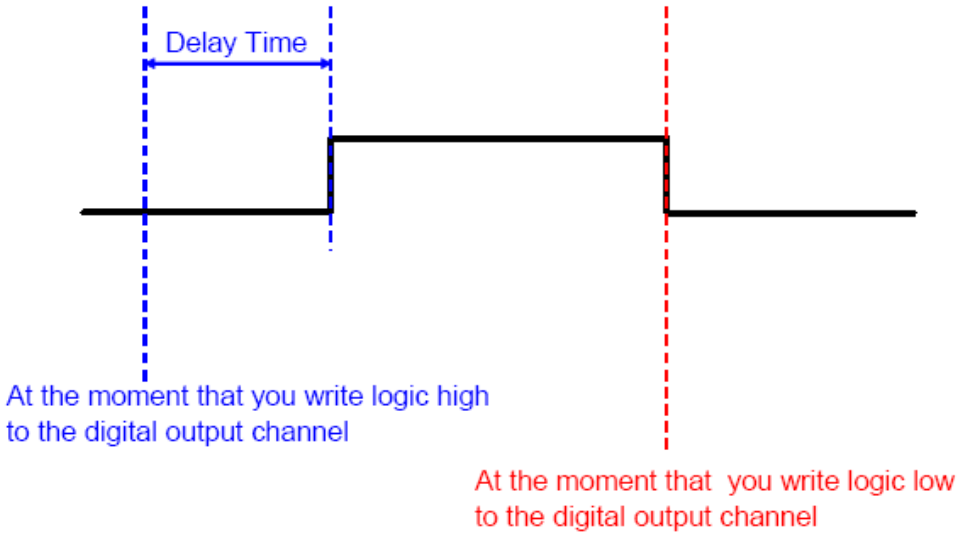
The pulse output is the same as PWR. After you choose the Pulse output mode, the selected digital output channel can generate continuous pulse train or finite pulses.

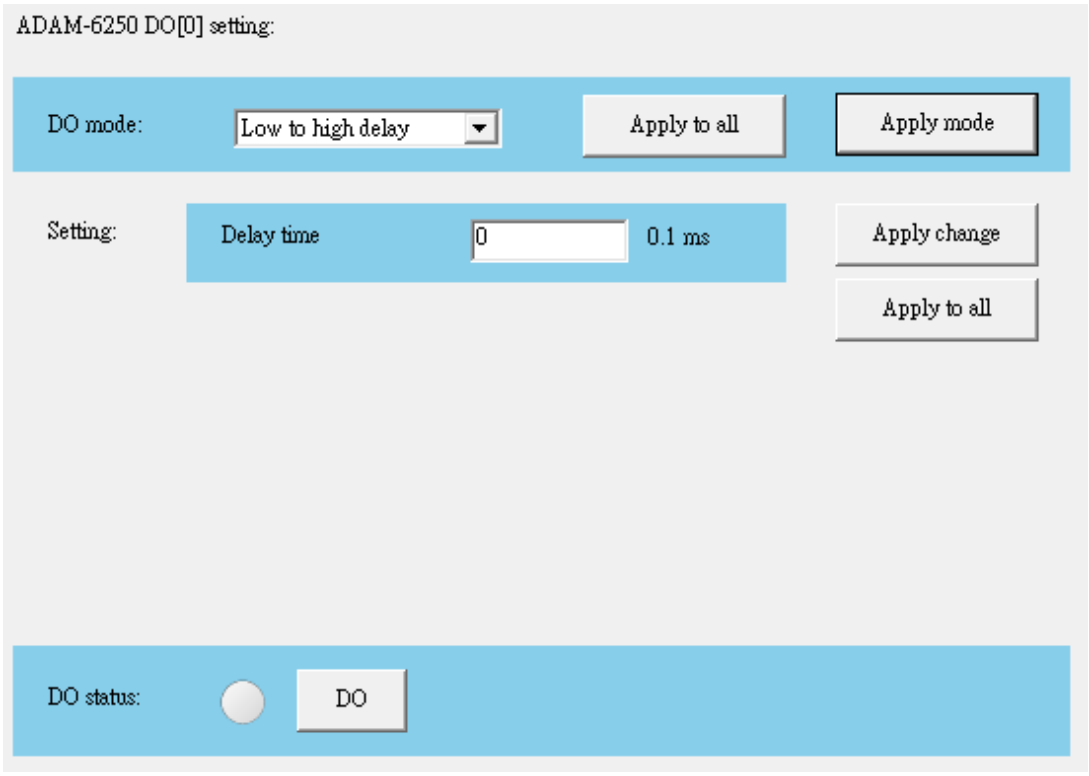
a. Signal width - It contains Low signal width and High signal width (unit: 0.1ms) for configuration
b. Output frequency -It will automatically indicate by the setting of Low/High signal width.
c. Duty cycle - It will automatically indicate by the setting of Low/High signal width.
d. Pulse output - You can select Continuous to generate continuous pulse train or Fixed total to define how many pulse you want to send.



Low to high delay

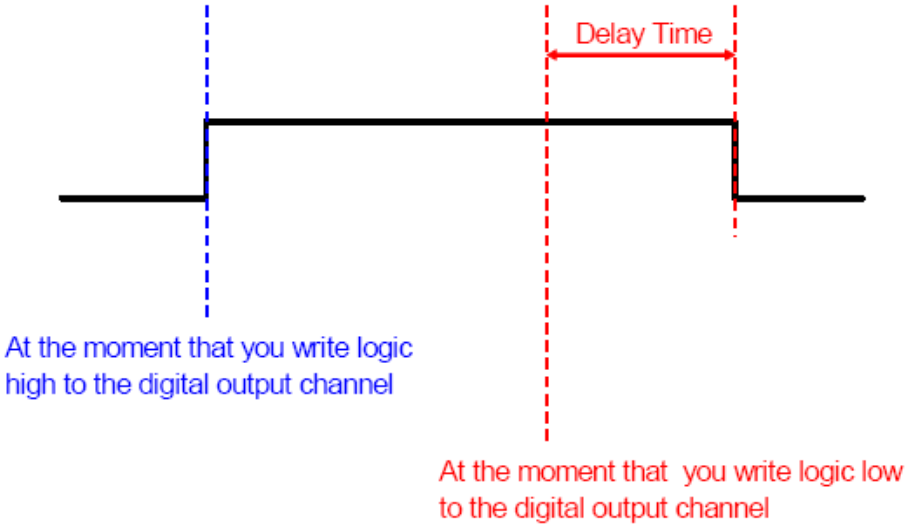
It means there will be certain time delay when the output value changes from logic low to logic high as below. You can define how much Delay time of a pulse as your need, and also control DO status by clicking DO button.

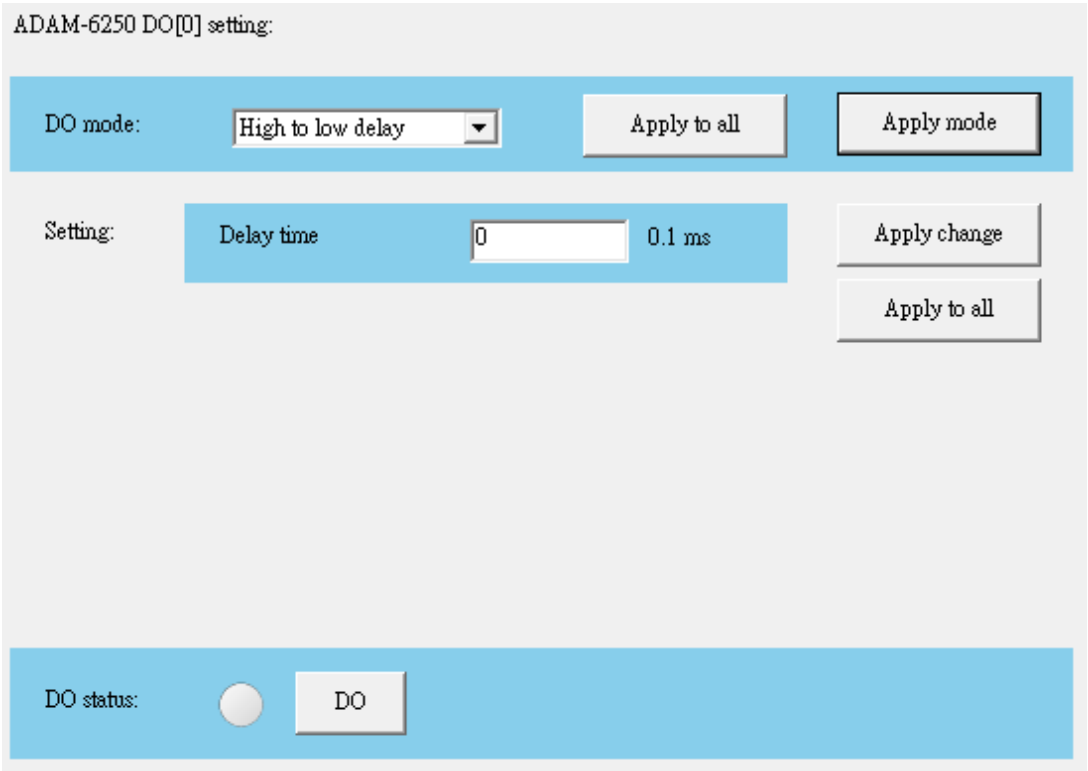




High to low delay

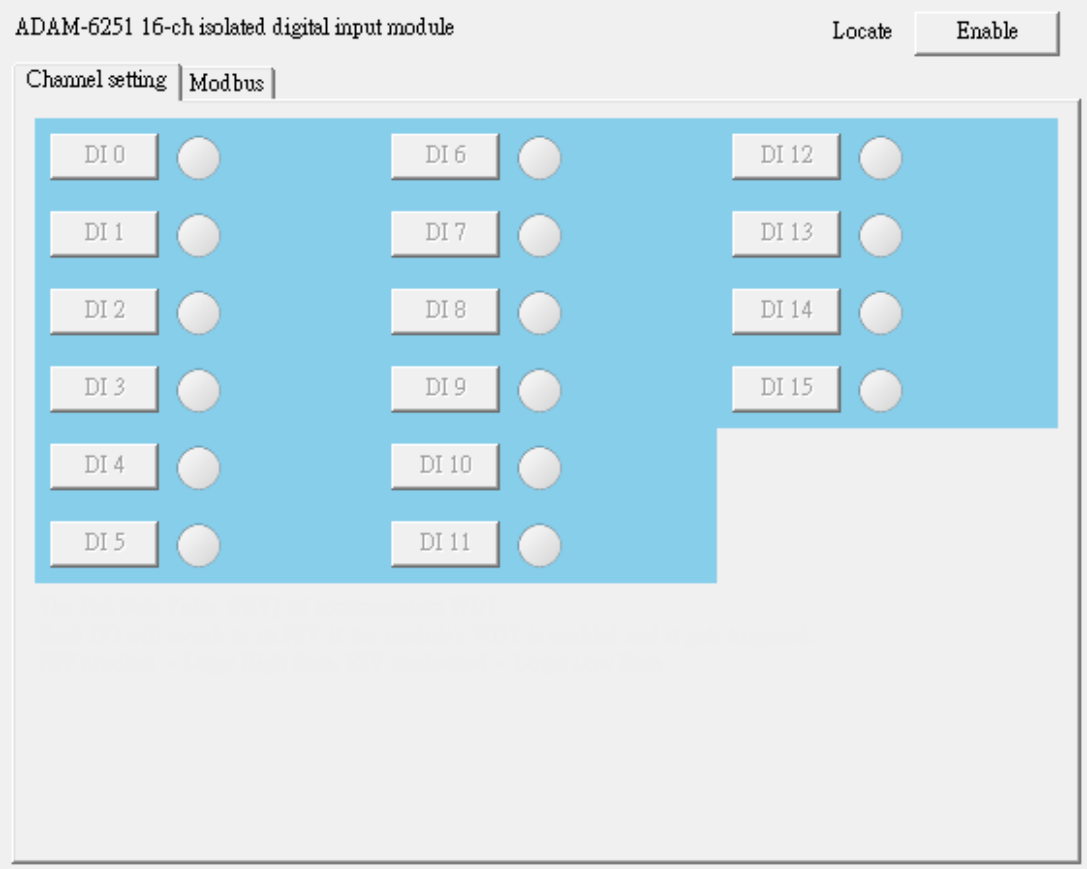
It means there will be certain time delay when the output value changes from logic high to logic low as below. You can define how much Delay time of a pulse as your need, and also control DO status by clicking DO button.





4.3.3.4 ADAM-6251

- **Digital Input Channel Overview**



Locate

It can help user search ADAM module with light sign. (Status LED will be on for 10 sec when it enabled.)

Channel Setting

You can see value of all digital input channels by related LED display in this tab. Besides, you also can control values of all digital output channels by related button. The LED next to the button will display current value of that digital output channel.

Modbus

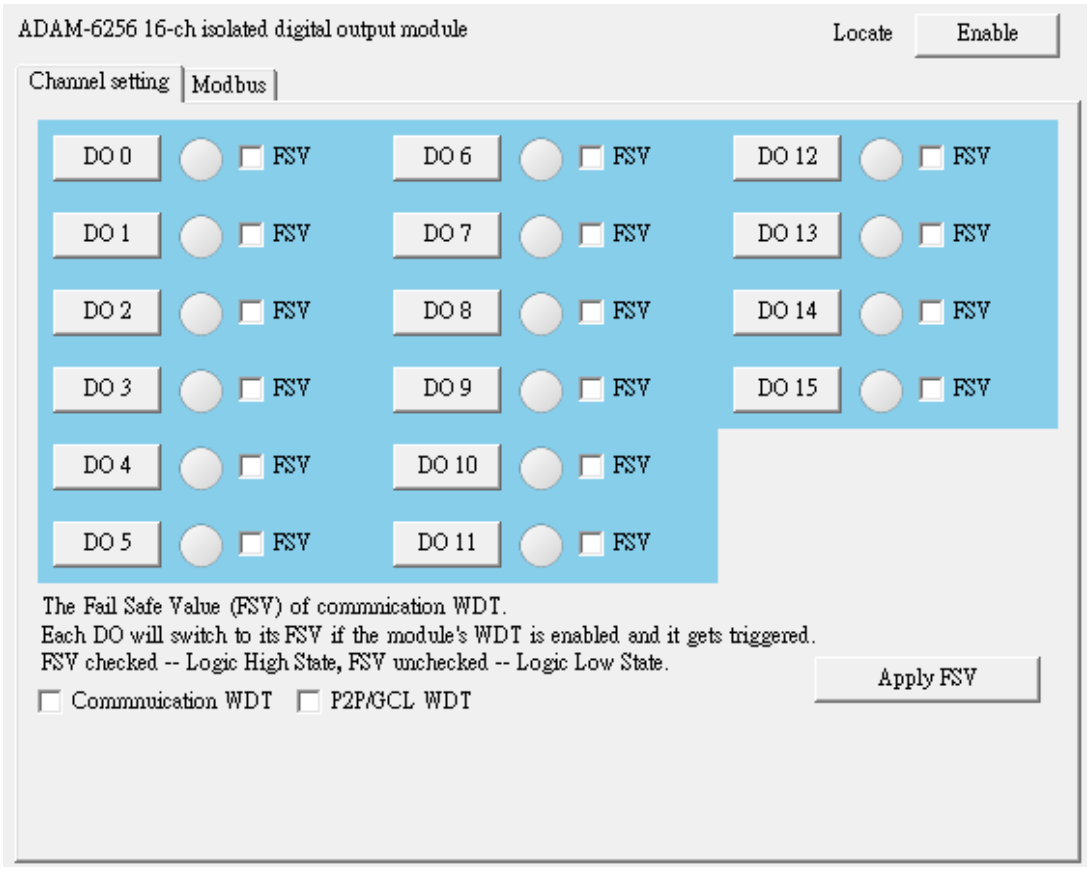
It indicates the overview of the status (value) of its corresponding Modbus address, channel and mode.

Advanced channel setting

Please refer to the DI operating statement of ADAM-6250.

4.3.3.5 ADAM-6256

■ **Digital Output Channel Overview**



Locate

It can help user search ADAM module with light sign. (Status LED will be on for 10 sec when it enabled.)

Channel Setting

You can see value of all digital output channels by related LED display in this tab. Besides, you also can control values of all digital output channels by related button. The LED next to the button will display current value of that digital output channel.

The Fail Safe Value (FSV) of communication WDT

If FSV checkbox is enabled, it presents the module will set the output channel to logic high when WDT timeout. there are two application as below. After all, click the Apply FSV button to take it effect.

a. Communication WDT
When the module hasn't received any TCP network packet from client too long, such as SCADA or Web browser. That means if the waiting time is greater than host idle time, the module will automatically set the safety value to output.

b. P2P/GCL WDT
When the module hasn't received P2P/GCL network packets too long, that means the waiting time is greater than idle time you entered, the module will automatically send the safety value to host PC if you've enabled this function.

Modbus

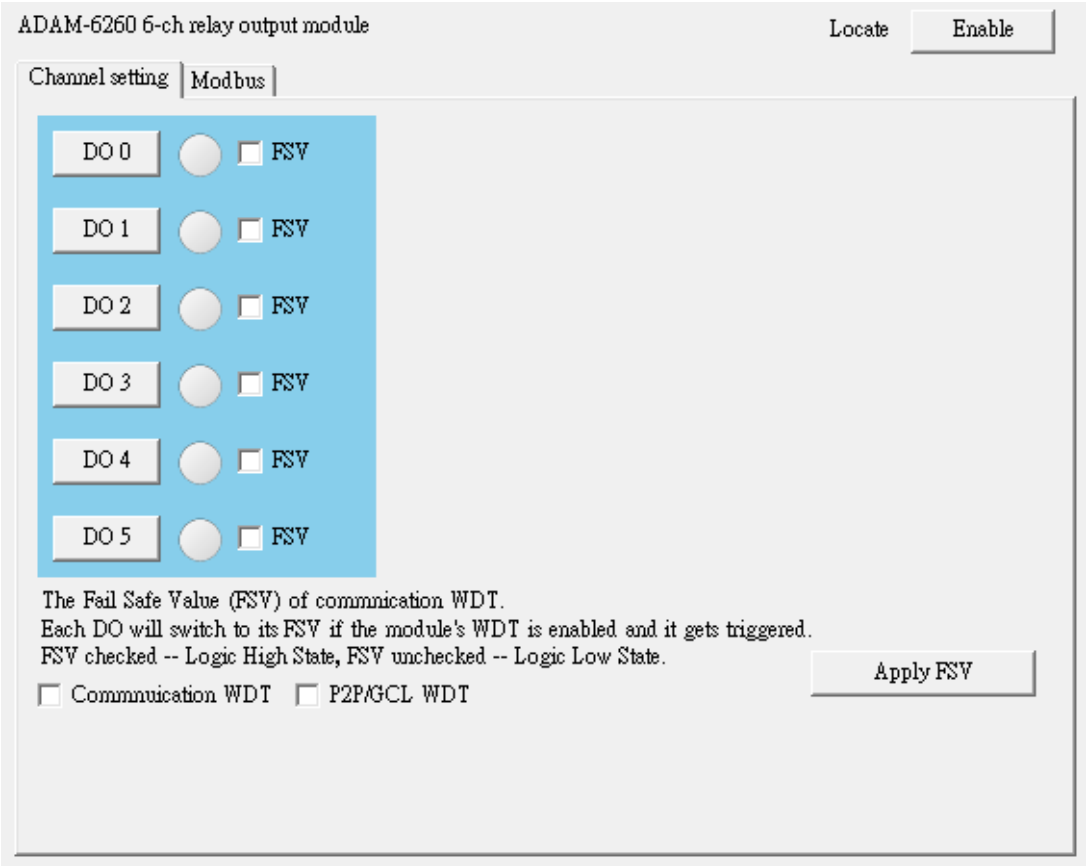
It indicates the overview of the status (value) of its corresponding Modbus address, channel and mode.

■ **Advanced channel setting**

Please refer to the DO operating statement of ADAM-6250.

4.3.3.6 ADAM-6260

- **Digital Output Channel Overview**



Locate

It can help user search ADAM module with light sign. (Status LED will be on for 10 sec when it enabled.)

Channel Setting

You can see value of all digital output channels by related LED display in this tab. Besides, you also can control values of all digital output channels by related button. The LED next to the button will display current value of that digital output channel.

The Fail Safe Value (FSV) of communication WDT

If FSV checkbox is enabled, it presents the module will set the output channel to logic high when WDT timeout. There are two application as below. After all, click the Apply FSV button to take it effect.

a. Communication WDT
   When the module hasn't received any TCP network packet from client too long, such as SCADA or Web browser. That means if the waiting time is greater than host idle time, the module will automatically set the safety value to output.

b. P2P/GCL WDT
   When the module hasn't received P2P/GCL network packets too long, that means the waiting time is greater than idle time you entered, the module will automatically send the safety value to host PC if you've enabled this function.

Modbus

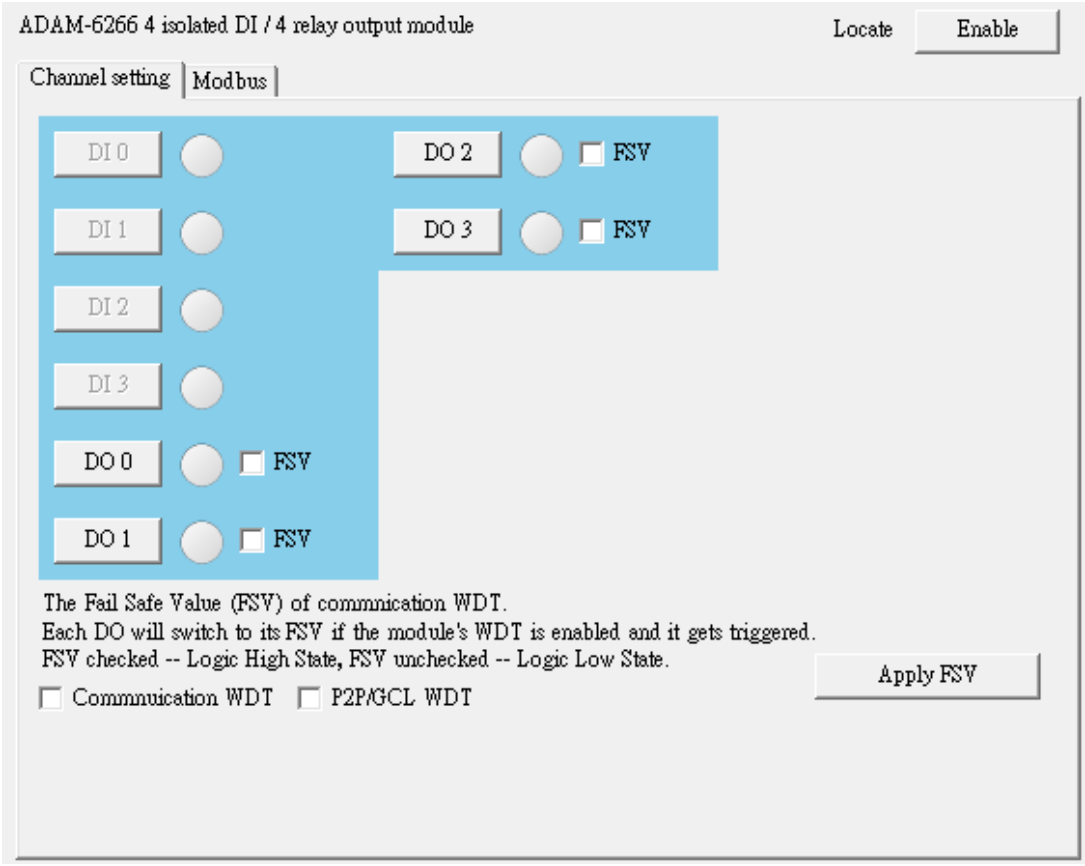
It indicates the overview of the status (value) of its corresponding Modbus address, channel and mode.

- **Advanced channel setting**

Please refer to the DO operating statement of ADAM-6250.

4.3.3.7 ADAM-6266

■ **Digital Output Channel Overview**



Locate

It can help user search ADAM module with light sign. (Status LED will be on for 10 sec when it enabled.)

Channel Setting

You can see value of all digital input channels by related LED display in this tab. Besides, you also can control values of all digital output channels by related button. The LED next to the button will display current value of that digital output channel.

The Fail Safe Value (FSV) of communication WDT

If FSV checkbox is enabled, it presents the module will set the output channel to logic high when WDT timeout. There are two application as below. After all, click the Apply FSV button to take it effect.

a. Communication WDT
When the module hasn't received network packets too long, that means the waiting time is greater than idle time you entered, the module will automatically send the safety value to host PC if you've enabled this function.

b. P2P/GCL WDT
When the module hasn't received P2P/GCL network packets too long, that means the waiting time is greater than idle time you entered, the module will automatically send the safety value to host PC if you've enabled this function.

Modbus

It indicates the overview of the status (value) of its corresponding Modbus address, channel and mode.

■ **Advanced channel setting**

Please refer to the DI/O operating statement of ADAM-6250.

4.3.4 Peer to Peer

When you want to send a signal from one module to another module, Peer-to-Peer is a perfect solution. With Peer-to-Peer function enabled, ADAM-6200 modules can actively update its input value to other devices such as PC or another ADAM-6200 module. One typical application is operating with a pair of ADAM-6200 modules. The value of input channel on one module will be automatically updated to output channel on another module.

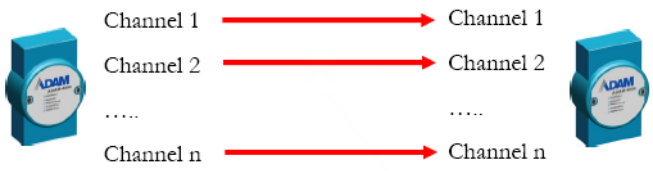
The data will be transferred automatically as long as the connection between the two ADAM-6200 modules is already built. No controller is needed to take care of the communication. ADAM-6200 modules feature two types of Peer-to-Peer function:

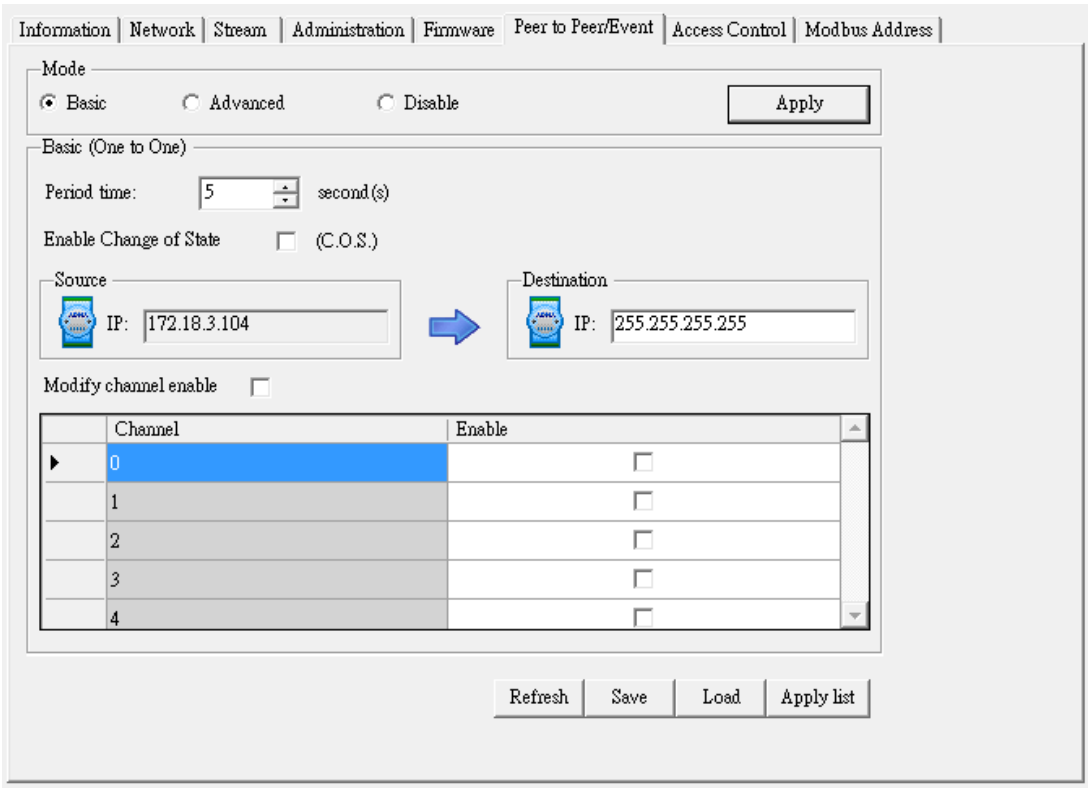
Note!

1. *Please use Ethernet Switch between a pair of Peer-to-Peer modules. Do not use an Ethernet hub. This can prevent data packet collision.*
2. *Peer-to-Peer (Event) and GCL function can't run at the same time.*

4.3.4.1 Basic Mode

In basic mode, it's defined as one-way data transmission. That means only one target device to receive the data transferred from one ADAM-6200 module. You can configure the IP address of target device (Destination) and which channel you're going to send data to its mapping channel.





1. Period Time Function:

The value of the input channel will be updated to the target devices with the defined period.

2. Period Time Function + C.O.S (Change of Status) Function

The value of the input channel will still be updated to the target devices with the defined period. Moreover, when C.O.S happens (the change of the analog input value is greater than specific deviation or digital input status changes), the value of the input channel will also update to the target

devices immediately.

Note!



1. *There will be uncertainty for network communication. Sometimes there might be packet lost when event occurs. This is the reason we Period Time function +C.O.S. function (no C.O.S. function only). When event occurs, even if the packet is lost, the data will be sent again when the next period reaches. This can help to make the system more reliable.*
2. *For preventing input noise also been send as digital input signal, we suggest to activate DI filter function if C.O.S (Change of Status) function enabled.*
3. *How to use Peer-to-Peer function to implement Event Trigger?*
   In many applications, the data will only be sent to a host computer when specific event happens. Typical event is that the digital or analog signal changes. To implement this kind of application, ADAM-6200 modules enabled with Peer-to-Peer function is a perfect solution. The target device of Peer-to-Peer can be a computer, simply by entering the IP of that computer into the Destination text box of Peer-to-Peer/Event configuration tab in ADAM.NET Utility. (The detail information about configuration is described below.) Choose Basic mode and Period Time function + C.O.S. function as communication method. There should be one program running on the host computer to receive the data, and we provide an example C program (VC++ 6.0) with ADAM-6200 module. Although the ADAM-6200 modules will send data to the host computer periodically (the reason is for communication security, see Note below), you can still distinguish the message is sent from Period Time function or C.O.S. function. The message contains which channel has changed. So if you find there is no change for all channels in the message information, then you can realize that there is no event happening.

Basic Mode Configuration

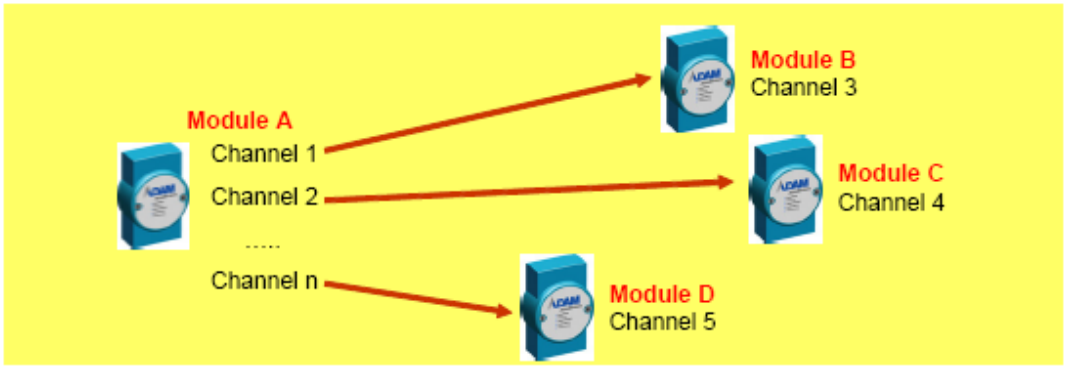
The Status Display of basic mode is shown above. You can define the target device by entering its IP address into the Destination text box in the Basic (One to One) area.

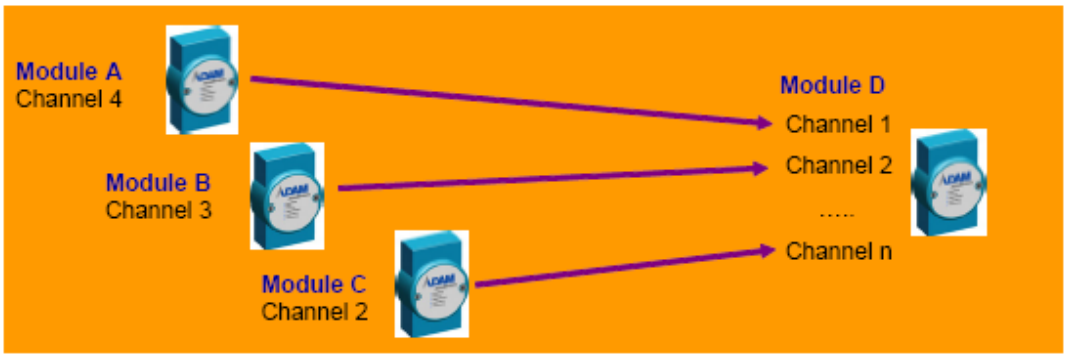
You can choose these two methods by click the Deviation Enable check box (for AI modules) or Enable Change of State check box (for digital modules). If this check box is not checked, the transfer method is Period Time function. The period to transfer data from source to destination is defined by the Period time numeric control in the Basic (One to One) area. If the check box is checked, the transfer method becomes Period Time function + C.O.S. function. You can define the deviation for analog input by the Deviation Rate numeric control (value is percentage unit and represent the change value divided by the total range). For Period Time function + C.O.S function, the data will be transferred from source to destination periodically. Besides, when the analog input value change is greater than what the Deviation Rate defined or the digital input channel value changes, the data will also update from source to destination automatically.

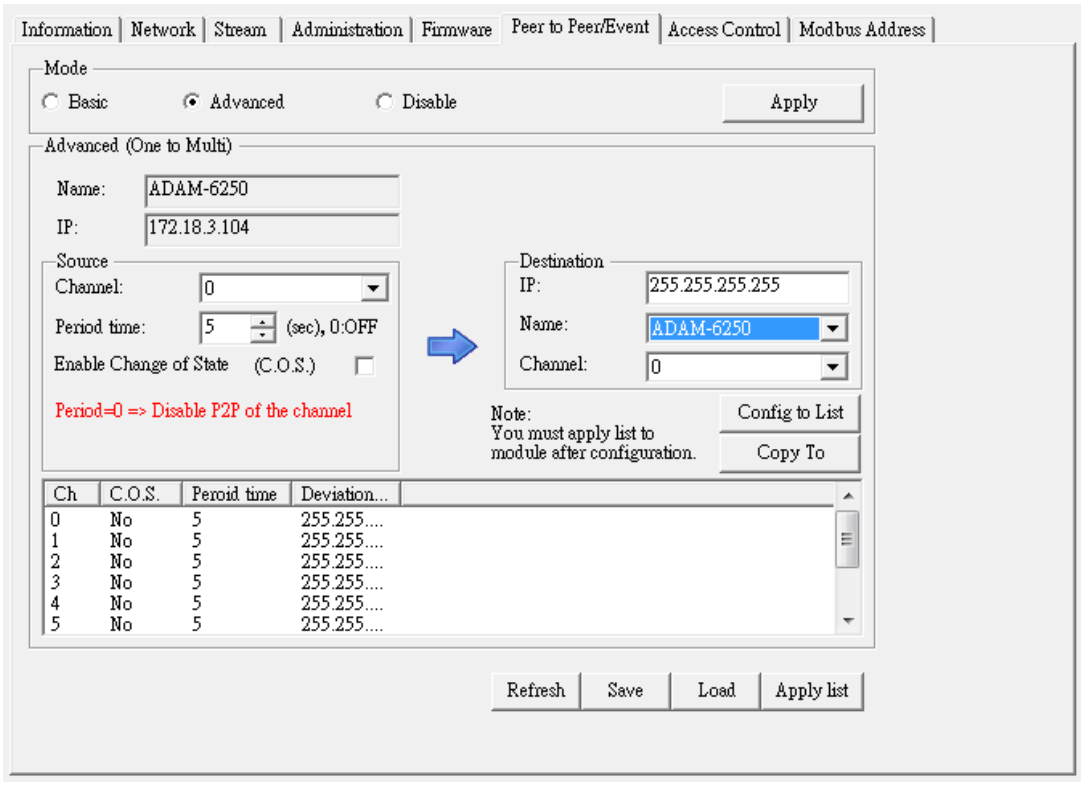
By default, all input channels of the source module will all be mapping to all output channels of the destination module. However, you can manually define which input channels are mapping to output channels, by clicking the Modify channel enable check box. When this check box is checked, you can select which input channels will be mapping to the corresponding output channels by click related Channel check box. In this example, only the value of input channels 0, 1, 2, 3 of the source module will update to the output channels 0, 1, 2, 3 of the destination module. After you have selected the channel, click the Apply list button to download this configuration to the source module. You can save current mapping relation into a configuration file in your computer by clicking the Save button. You also can load previous mapping configuration file by clicking the Load button. If you click the Refresh button, the current mapping configuration on the source module will be displayed in the Channel-Enable table.

4.3.4.2 Advanced Mode

In advanced mode, there will be multiple target devices to receive the data transferred from one ADAM-6200 module (Module A). You can define different target devices (by different IP address) to each channel of module A. For example, you can define the input channel 1 of Module A is mapping to the output channel 3 of Module B, while input channel 2 of Module A is mapping to the output channel 4 of Module C. So value of input channel 1 and 2 on Module A will automatically update to channel 3 on Module B and channel 4 on Module C, respectively. Please refer to the following figures for better understanding.







Advanced Mode Configuration

When you choose the advanced mode, the Main Operation screen area is shown as above. With advanced mode, each channel on the source ADAM-6200 module can be mapping to channel on different target devices. You can configure the mapping relation using the two block areas Source and Destination in the Advanced (One to Multi) area.

The setup of the mapping relationship is shown as below:

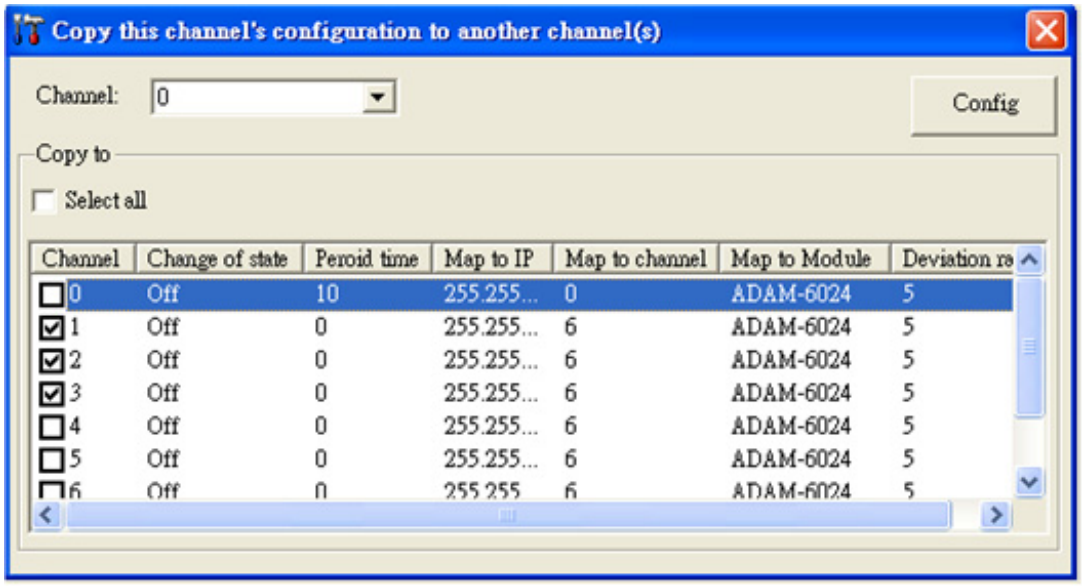
1. Select the input channel by the **Channel** combo box in **Source**.
2. Use Period time numeric control, **Deviation enable (C.O.S)** checkbox (for analog modules) or **Change of state (C.O.S)** check box (for digital modules), and **Deviation Rate** numeric control in the Source area to define when to transfer the data for that channel. The configuration operation is similar to **Basic** mode.
3. Define the target device for that channel by entering its IP address into the IP text box in the **Destination** area. Choose the correct module name from the **Name** combo box, and choose the output channel to receive the data by the **Channel** combo box.
4. After you have completed all these configurations for this channel, click the **Config** to list button. You can see your configuration for that channel is displayed by the mapping table below the **Source** and **Destination** area.

Note!



1. *The mapping setting is only restored in memory of your computer, and it will download to the target ADAM-6200 module after you click the **Apply list** button below the mapping table. It is not suggested to download the mapping configuration immediately if you only complete setting for one channel.*
2. *We suggest you to download all channels mapping configuration together at one time, instead of downloading one-channel setting many times. The reason is that this can save the times to use the flash memory on target module and help to extend the flash memory life.*
3. *Peer-to-Peer Data Transfer Performance*
   For example, the data transfer time of a pair of ADAM-6250 will be less than 1.2 milliseconds.

Repeat the step 1 to step 4 for another input channel. Continuously repeat the configuration until you have configured all the input channels which you want to create the mapping relationship. Click the **Apply list** button to download all mapping configuration to the target module. You can save all configurations in the mapping table into a file by clicking the **Save** button. Or you can load previous configuration from a file by clicking the **Load** button. If you click the **Refresh** button, the real configuration on the source module will be uploaded to your computer and you can see it in the mapping table. In order to save setting time for the mapping configuration, you can copy one channel setting to other channels, and then only change what needs to change. You can do this by clicking the **Copy to** button, then a dialog window will pop-up as below image.



Choose the channel which provides the setting for other channels by the **Channel** combo box at the top of the dialog window. Then select channels which you want to copy setting to by clicking the **Channel** checkbox in the **Copy to** area. Using as an example, setting of channel 0 will copy to channel 1, 2, and 3. If you want to copy the setting to all channels, click the **Select all** check box. After selecting the channels, click the **Config** button. Then you will find the setting of the channels you selected has been copied in the mapping table. After that, you can individually select the channel needed to modify and change the parameters. Therefore, you don't need to do all the configurations and you can concentrate on setting on the parameters needed to be modified.

4.3.5 GCL

4.3.5.1 Overview

In a traditional control and data acquisition system, there must be one controller to manage the system. Remote I/O modules like the ADAM-6200 modules, only acquire data from sensors, or generate signal to control other devices or equipment. There must be a computer (or a controller, such as PLC) responsible to get the data from the input modules, manipulate the data, execute logic operation and process depending on the input data, and generate output data to the output modules based on the logic decision.

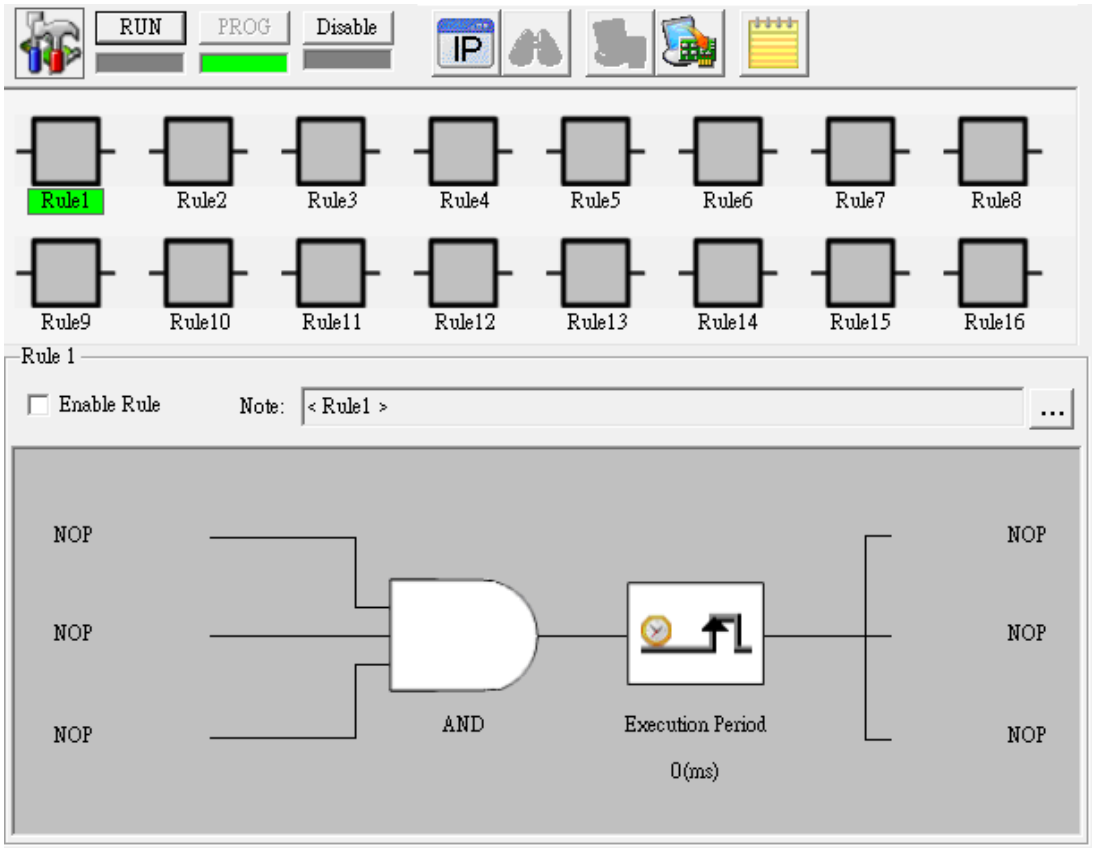
The computer (or controller) and remote I/O modules form a complete control system within the same network. The complexity of logic operation and process depend on the application, and it is implemented by the program written on the computer (or controller). There are plenty of software applications to write programs. Examples are C language, Microsoft Visual Studio for computer, and Ladder language for PLC controller.

In many applications, the logic operation and process is not very difficult that it seems not so necessary to implement a computer or controller which are too powerful than needed. Now, ADAM-6200 modules feature logic operation and process ability by the new design --- **Graphic Logic Condition (GCL)**. This feature makes the ADAM-6200 modules become a smart I/O module that it can play as a standalone control system.

People can define the logic operation and process rules in the ADAM.NET Utility and download the rules to the ADAM-6200 modules. Then ADAM-6200 modules will execute the logic rules to process different action depending on the input conditions.

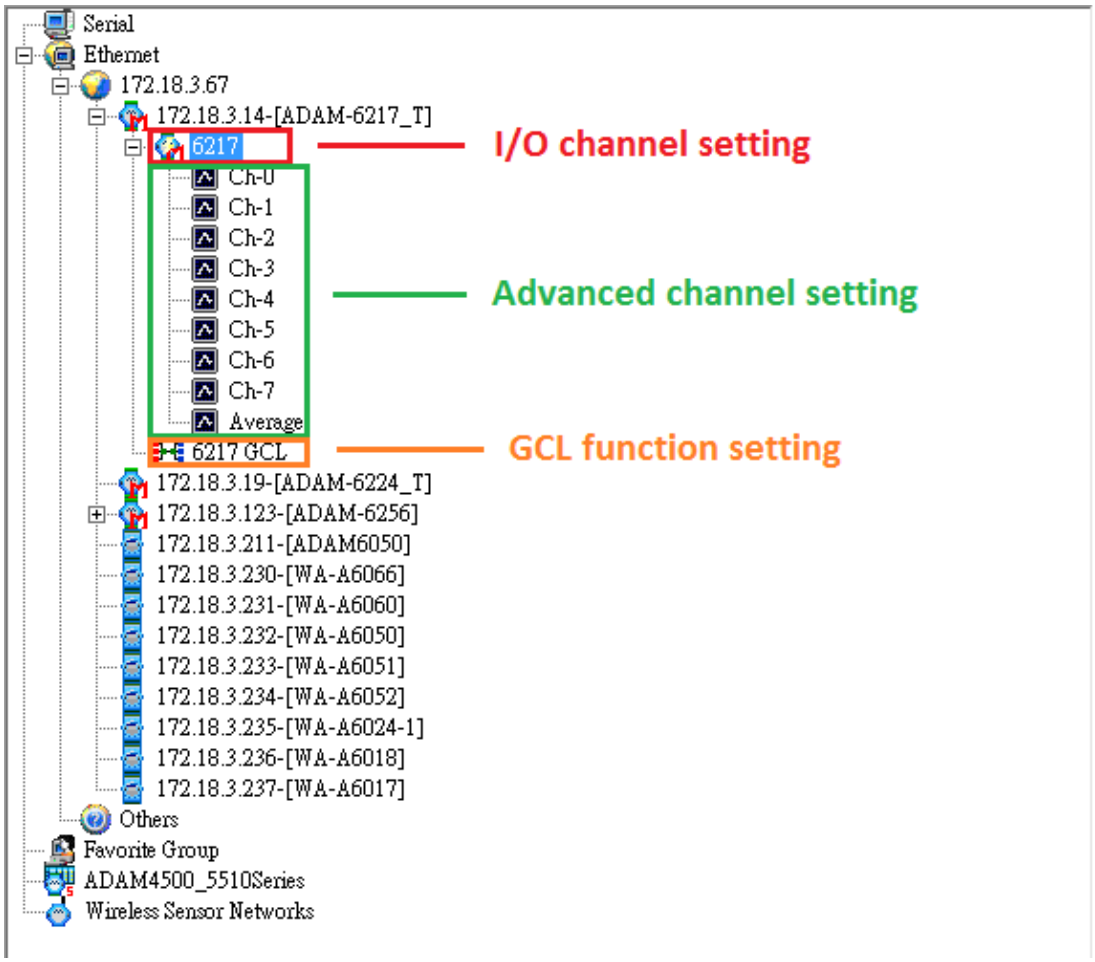
With GCL enabled, a computer (or a controller) can be removed from the control system since the ADAM-6200 modules can play as controller by themselves.

The configuration environment for GCL in ADAM.NET Utility is completely graphical, making it very easy and intuitive to complete the logic rule configuration. After completing the logic rule configuration and download, engineers can see the real-time execution situation and input value in ADAM.NET Utility on line. We will introduce these features in more detail by following content.



4.3.5.2 Configuration Environment

As previously mentioned, when you try to spread one of ADAM-6200 module, you will be allowed to check the channel setting and GCL function by entering correct password in the Module Tree screen.



Note! *You can configure all GCL related setting by clicking the GCL Configuration item list. For the two features Peer-to-Peer and GCL, only one of them can be enable at one time. If you enable Peer-to-Peer function before, when you click the GCL Configuration and launch the GCL configuration environment, you will find that it is disabled by default. Once you click the Program GCL button (refer to table below) to enable the GCL feature, the Peer-to-Peer function will be disabled.*

The top icon definition in GCL Menu:

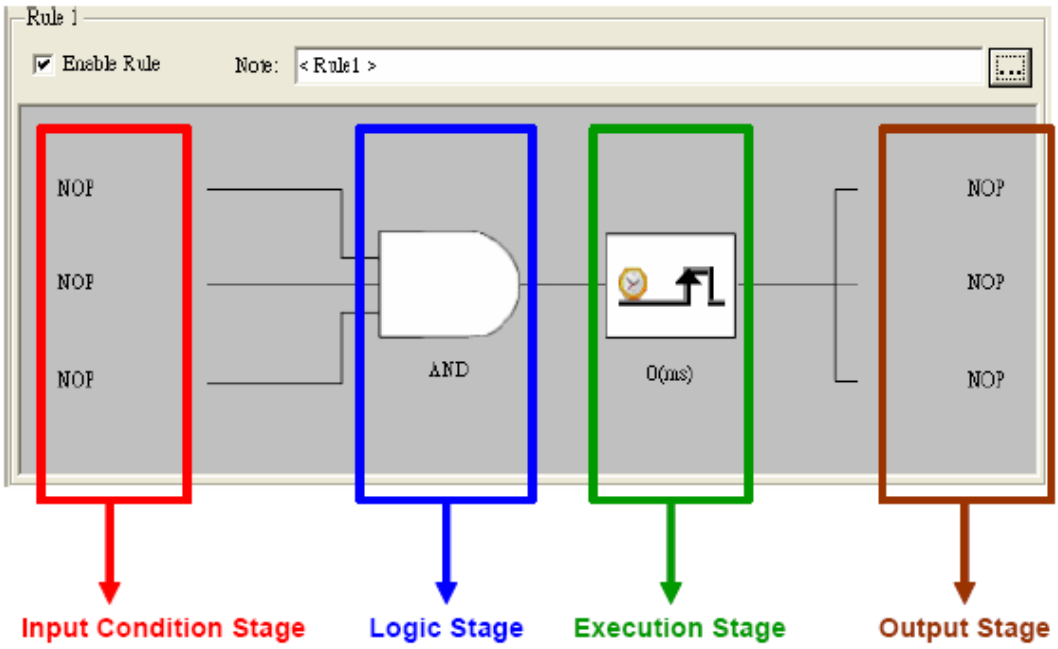
Icon	Function	Description
	Current Status	This icon shows current GCL status. The status represented in the Icon cell is the **Disable**, **Programming** and **Running** mode (From top to button) Note: You cannot enable Peer-to-Peer/Data Stream function and GCL function at the same time. So if you want to enable GCL, Peer-to-Peer and Data Stream function will be disabled automatically.
	Run GCL	Select the **Running** mode. If this mode is chosen, the LED below the button is lit.
	Program GCL	Select the **Programming** mode. If this mode is chosen, the LED below the button is lit.
	Disable GCL	Select the **Disable** mode. If this mode is chosen, the LED below the button is lit.
	IP Table Configuration	Click this button to configure IP table. IP table can used to define the destination for output. Only available in the **Programming** mode.
	Monitoring	Enable Online Monitoring. Only available in the **Running** mode.
	Upload Project	Upload GCL configurations from ADAM-6200 module to computer. Only available in the **Programming** mode.
	Download Project	Download current GCL configurations to the ADAM-6200 module. Only available in the **Programming** mode.
	Project Content	Click this button to show current GCL configurations. You can also save current configurations into a file, or load previous configuration from a specific file.

Logic rule configuration:

There are 16 logic rules available on one ADAM-6200 module, so you can see 16 logic rule icons here. Simply click the logic rule icon to configure that rule. For example, if you want to configure rule 12, just click the logic rule icon with text "Rule 12" below. The text background color of the selected logic rule icon will become green.

After the rule number selected, you need to check the **Enable Rule** checkbox to enable that logic rule and the color of that logic rule icon will become white. Also, you can write some description for that logic rule by clicking the button next to the **Note** text box.

There are four stages for one logic rule: **Input Condition**, **Logic**, **Execution** and **Output**. You can simply click the graphical icon to configure each stage and one related configuration window will pop up.



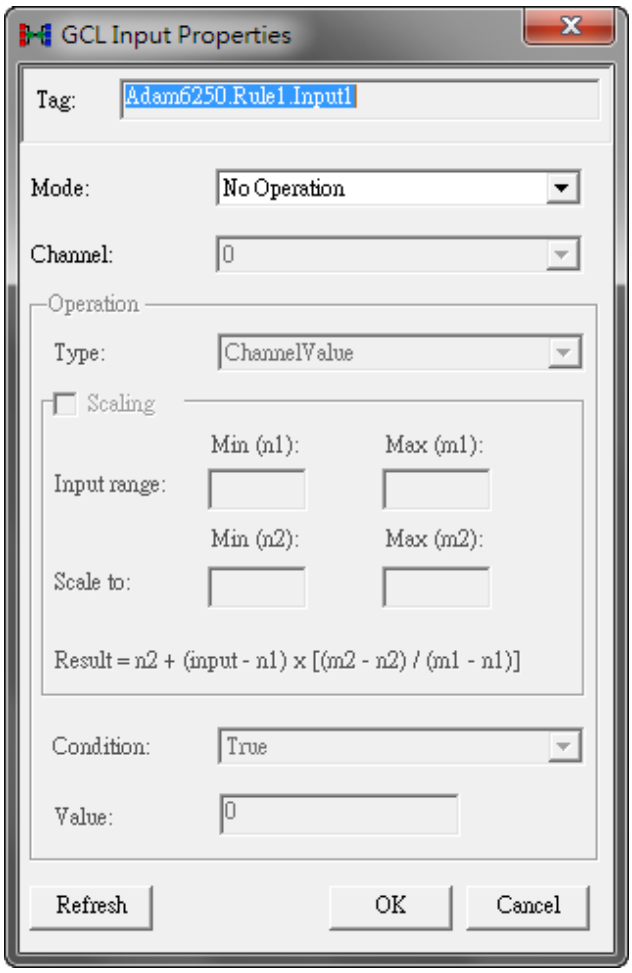
4.3.5.3 Basic Logic rule programming

■ Input Condition stage

The Input Condition stage is a logic condition decision for the input data.

The decision result will be logic True or False, sending to the Logic stage for logic operation. Take analog input mode as example, you can define the condition as if the analog input value is greater than a specific value (the limit). So when the input value becomes greater than the limit, the input stage will transfer True to the Logic stage. Otherwise, it will transfer False to the Logic stage.

When you click the Input Condition stage icon, you should see a dialog window similar as below. You can choose the input mode by the Mode combo box. The default mode is NoOperation, meaning there is no input condition. You can choose local analog input channel (AI), local digital input channel (DI), local counter input channel (DI_Counter), local frequency input channel (DI_Frequency), internal timer (Timer), internal flag (AuxFlag) , local digital output channel (DO) and internal counter (Counter) as the input mode. After you choose the appropriate input mode and complete all related setting, click the OK button. That Input Condition stage icon will change its pattern to present the current condition. We will describe each mode in more detail below.



Mode	Description	Condition
No Operation	No operation	N/A
AI	Local AI channel value	>, =, <
DI	Local DI channel value	True, False
DI_Counter	Local counter input channel value	>, =, <
DI_Frequency	Local frequency input channel value	>, =, <
Timer	Local internal Timer value	>, =, <
AuxFlag	Local internal Flag value	True, False
DO	Local DO channel value	True, False
Counter	Local internal counter value	>, =, <

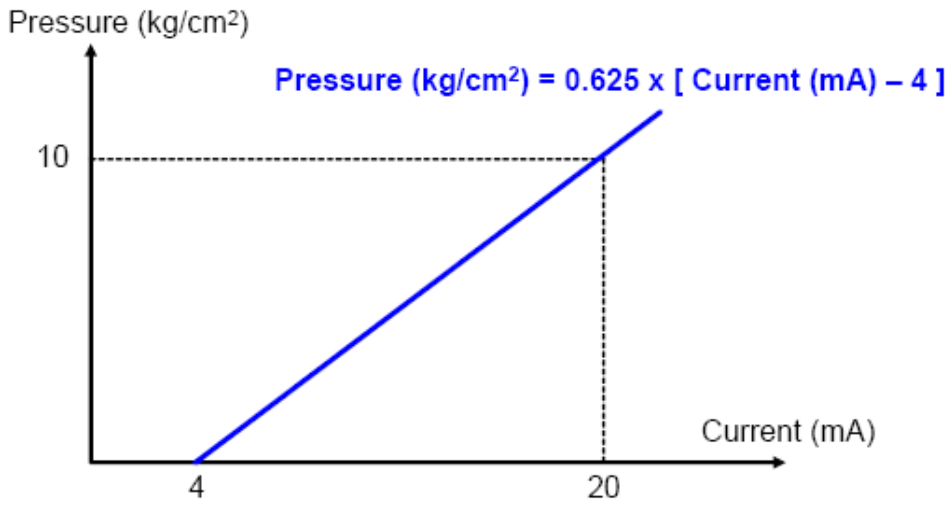
Local Analog Input Channel (AI)

1. After you choose AI as input mode, select the channel by the Channel combo box.
2. In the Operation area, you can define the input condition operation. Select the analog input type by the Type combo box. There are two input types you can choose for analog input: If you select Channel Value, the current value of the selected analog input channel is used as input for condition. If you select Deviation, the deviation (Dividing difference between present sample value and previous sample value by the total range value) of the selected analog input channel is used as input for condition.
3. Select the appropriate condition for that input channel by the Condition combo box and the Value text box. Refer to the examples in the table below to see how to define analog input condition:

Channel	Type	Condition	Value	Description
0	Channel value	>=	5	If the value of analog channel 0 is more than or equal to 5, the condition result is logic **True**. Otherwise, the condition result is logic **False**.
2	Channel value	=	3.2	If the value of analog channel 2 equals to 3.2, the condition result is logic **True**. Otherwise, the condition result is logic **False**.
3	Channel value	<=	1.7	If the value of analog channel 3 is less than or equal to 1.7, the condition result is logic **True**. Otherwise, the condition result is logic **False**.
5	Deviation	N/A	20	If the deviation of analog channel 5 is greater than 20%, the condition result is logic **True**. Otherwise, the condition result is logic **False**.

The analog input will read voltage (or current) from the channel we specified.

Usually, the voltage (or current) value can represent the real-world physical unit value (we call it engineer unit value) and there is linear relationship between the voltage (or current) value and the engineer unit value. For example, the current and the engineer unit value have linear relationship as shown below:



ADAM-6200 analog input module features Scaling function to convert the voltage (or current) value to the engineer unit value. For example, that's say the condition is if the pressure value is more than or equal to 2.5 kg/cm2. Without scaling function, you need to convert the pressure value (2.5 kg/cm2) to the current value (8 mA). Then you enter the current value 8 mA in the Value text box of the Operation area to define the condition.

Actually, you don't need to transfer the pressure value to current value by yourself. You can enable the scaling function by clicking the Scaling checkbox in the Operation area. Then enter the minimum and maximum value for the engineer unit in the Min and Max text box of the Scale to item, to build relationship between the voltage (or current) value and the engineer unit value. For example here, you should enter 0 and 10 as the minimum and maximum pressure value. Since ADAM-6200 module can automatically transfer the pressure value to the current value, you just need to enter the pressure value, 2.5 kg/cm2, into the Value text box to define the condition. This functionality can help you to configure the condition more intuitively.

Local Digital Input Channel (DI)

After you choose DI as input mode, select the channel by the Channel combo box. The value of the selected digital input channel will directly be the input of condition. If the condition is set as True, when value of the selected DI channel is logic high, the condition result is logic True. Otherwise, if the condition is set as False, when the value of DI channel is logic low, the condition result is logic True.

Local Counter Input Channel (DI_Counter)

After you choose DI_Counter as input mode, select the channel by the Channel combo box. The count value of the selected counter input channel will directly be the input of condition. Like the Analog Input condition, select the appropriate condition for that input channel by the Condition combo box and the Value text box (unit: 0.1Hz). The condition will compare the value read from the counter input channel with the value set by the Value text box. If condition is satisfied, the condition result is logic

True. Otherwise, the condition result is logic False.

Local Frequency Input Channel (DI_Frequency)

After you choose DI_Frequency as input mode, select the channel by the Channel combo box. The frequency value of the frequency input channel will directly be the input of condition. Like the Counter Input condition, select the appropriate condition for that input channel by the Condition combo box and the Value text box. The condition will compare the frequency value read from the frequency input channel and the value set by the Value text box. If condition is satisfied, the condition result is logic True. Otherwise, the condition result is logic False.

Internal Timer (Timer)

There are 16 local timers on one ADAM-6200 module. After the timer is started, its value represents how long the time has passed. Here, you can read the timer value and use it as input condition. After you choose Timer as input mode, select appropriate timer by the Index combo box.(From timer 0 to timer 15) Then you can define the condition by the Condition combo box and the Value text box (unit: 0.01 second). Only when the condition is met, the condition result is logic High. For example, if you choose >= in the Condition combo box and type 500 in the Value text box, it means the condition result will remain logic Low until the timer value is greater 5 second (meaning 5 second is passed).

After 5 seconds is passed, the condition result will become logic High.

Internal Flag (AuxFlag)

There are 16 internal flags on one ADAM-6200 module. The data type of internal flag is digital, meaning its value is either logic True or logic False. You can read the internal flag value and use it as input condition. After you choose AuxFlag as input mode, select appropriate internal flag by the Index combo box. (From flag 0 to flag 15) Then you can define the condition by the Condition combo box. If you choose True in the Condition combo box, it means only when the internal flag value equals to logic True, the condition result is logic True. If you choose False in the Condition combo box, only when the internal flag value equals to logic False, the condition result is logic True. You can use internal flag to implement logic cascade or logic feedback. Refer to Section 7.4 for more detail about how to achieve this.

Local Digital Output Channel (DO)

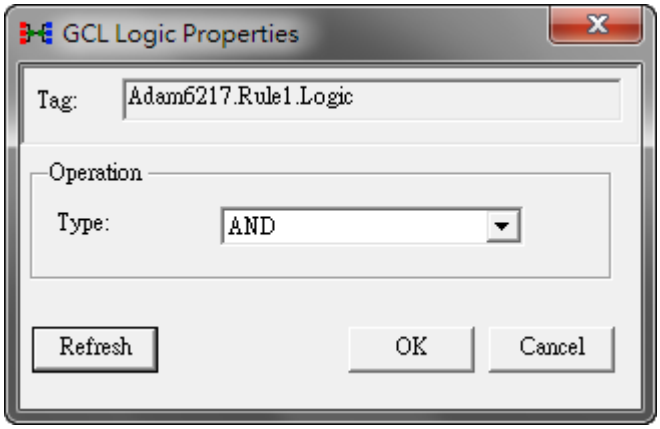
After you choose DO as input mode, select the channel by the Channel combo box. The value of the selected digital output channel will directly be the input of condition. If you choose True in the Condition combo box, it means only when the value of the selected DO channel equals to logic True, the condition result is logic True. If you choose False in the Condition combo box, only when the value of the selected DO channel equals to logic False, the condition result is logic True.

Internal Counter (Counter)

There are 8 internal counters on ADAM-6200 module, you can read its value as input condition. After you choose Counter as input mode, select the counter index by the Channel combo box. (From counter 0 to counter 7). The count value of the selected internal counter will directly be the input of condition. Like the Frequency Input condition, select the appropriate condition for that counter by the Condition combo box and the Value text box. The condition will compare the value read from the internal counter and the value set by the Value text box. If condition is satisfied, the condition result is logic True. Otherwise, the condition result is logic False.

Logic Stage

When you click the Logic stage icon, you should see a dialog window as below.



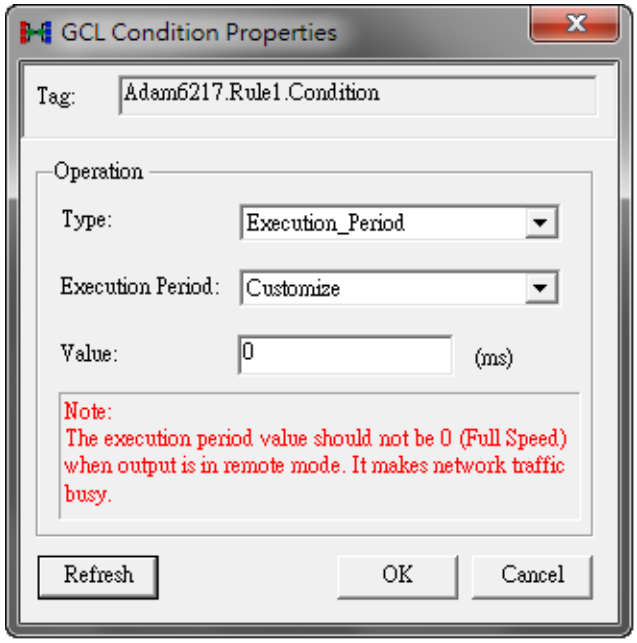
For each logic rule, there will be at most three input conditions passing logic True or False values to the Logic stage here. You can choose four logic operations by the Type combo box: AND, OR, NAND, NOR. The logic operation will process the input logic values, and generate a logic result value to the next Execution stage. After you have selected the appropriate logic operation, click the OK button. The Logic stage icon will change its pattern to present the current logic operation.

In order to present how the four logic operations work, we use the truth table to describe. Here we take two input conditions as example. The letter "T" means logic True, while the letter "F" means logic False.

Logic	Input condition 1	Input condition 1	Outcome
AND	F	F	F
	F	T	F
	T	F	F
	T	T	T
OR	F	F	F
	F	T	T
	T	F	T
	T	T	T
NAND	F	F	T
	F	T	T
	T	F	T
	T	T	F
NOR	F	F	T
	F	T	F
	T	F	F
	T	T	F

Execution Stage

When you click the Execution stage icon, you should see a dialog window as below. There are two possible execution setting you can choose by the Type combo box in the Operation area: Execution Period (Execution_Period) and Send to Next Rule (SendToNextRule). After you choose the appropriate execution setting, click the OK button. The Execution stage icon will change its pattern to present current execution setting condition. We will describe each type in more detail.



Execution Period (Execution_Period)

As we mentioned before, the Logic stage will transfer logic result value, logic True or logic False, to the Execution stage here. The Execution stage will pass this value to the Output stage after a specific period. Below are the steps to configure this period:

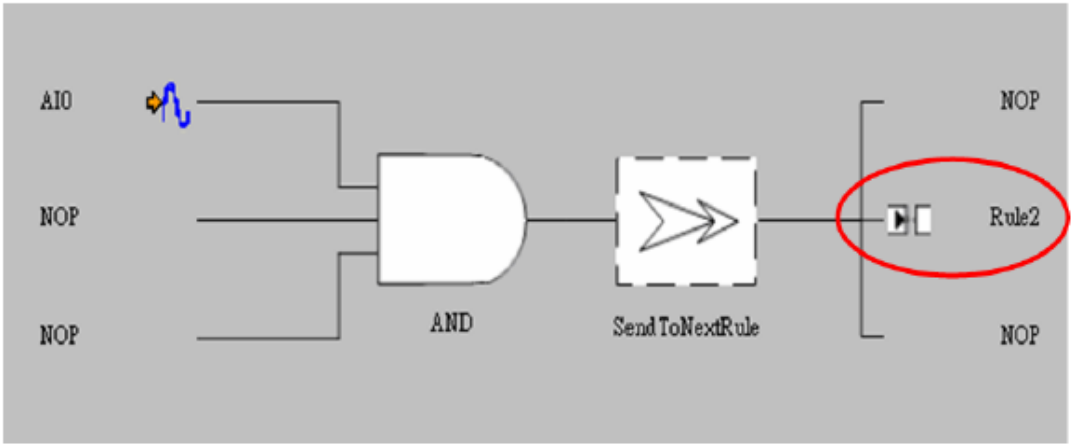
1. Select Execution_Period in the Type combo box.
2. Choose the appropriate period by the Execution Period combo box. You can select some pre-defined period from 1 ms to 60000 ms. You can also select Customize to define the period by yourself, then enter the period value by the Value text box (unit is ms).
3. Click the OK button to complete the configuration.

Note!

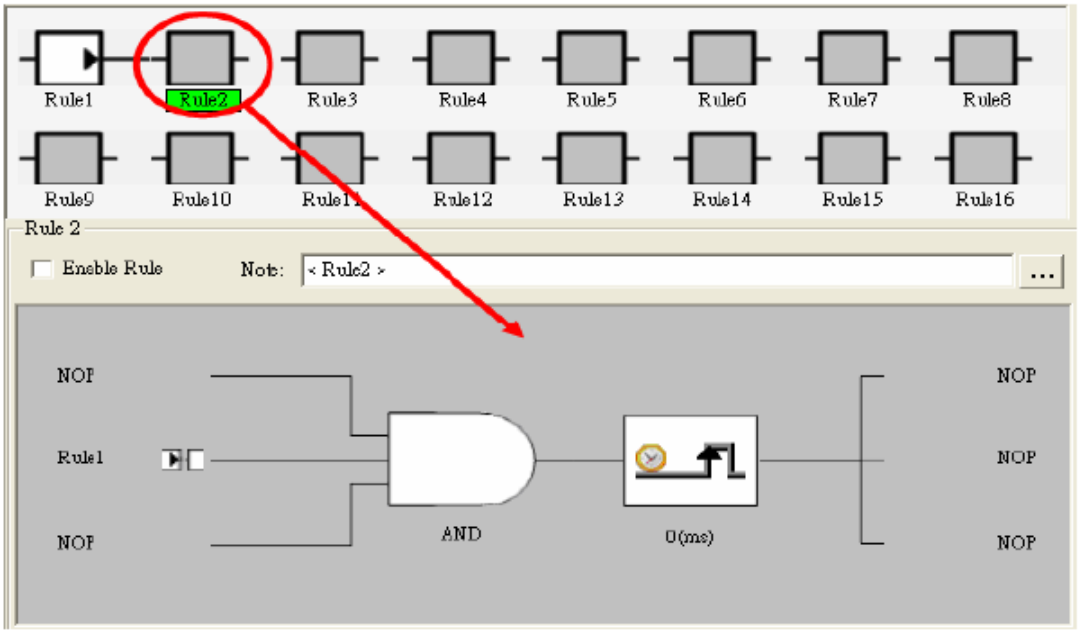
1. *If you choose Full speed in the Execution Period combo box, the execution speed will be as fast as possible. There might be network communication traffic problem when the output is on another module, since the execution speed is too fast that too many network packets are transferred on the Ethernet.*
2. *When you want to use ADAM.NET Utility to configure one ADAM-6200 module which is already running its GCL rules, remember to stop the GCL logic rules first.*

Send to Next Rule (SendToNextRule)

You can combine different logic rules into one single rule, which can help building more complex logic architecture. There are two methods to combine different logic rules: one way is using Send to Next Rule function here, another way is using Internal Flag. When you use Send to Next Rule function, you can set output of one logic rule being input of the next logic rule. Please note it can only combine two logic rules which are next to each other on the same module. If you want to combine different logic rules which are not next to each other, or even on different modules, you need to use internal flag for logic rule cascade. After you select SendToNextRule in the Type combo box, one of the output icons will become the next rule.

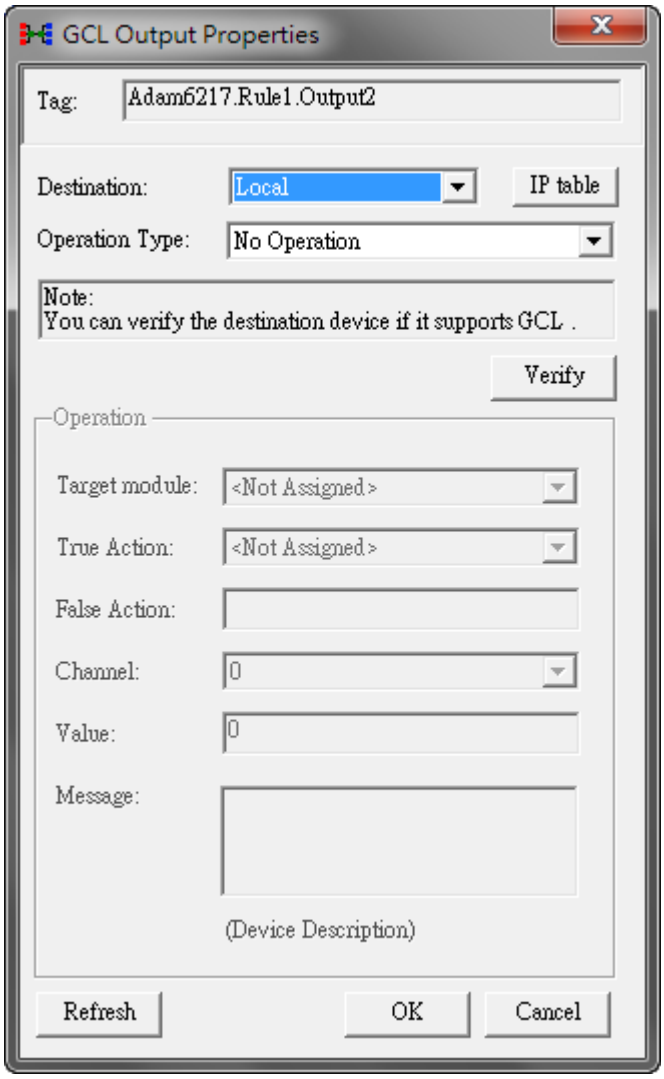


If you click the next logic rule icon, you will find one of the input condition become previous logic rule. Refer as below. Therefore, the logic result value from the previous logic rule (in this example, logic rule 1) will be one of logic input value of current logic rule (in this example, logic rule 2). This makes the two neighbor logic rules combined together. We call it Logic Cascade. Using this method for Logic Cascade, only the two neighbor logic rules can be combined together. If you want to combine two logic rules that are not next to each other, you need to use internal flag.



Output Stage

When you click the Output stage icon, you will see a pop-up window as below. There are three outputs for one logic rule. The logic result value from the Execution stage will be passed to the three outputs. And the three outputs will have different action depend on the logic result value.



You need to decide the target device for the output by the Destination combo box. You can choose Local, meaning the output is on the same module, or another remote module by its IP address. Select the appropriate IP address listed in the combo box. The IP addresses are defined in the IP table and you can click the IP Table button to configure. (The action of clicking the IP Table button here is just the same as click the IP Table Configuration button in the GCL Menu area.)

Note! *When your output destination is not Local, meaning there will be communication between the specific ADAM-6200 module to its target device, remember to use Ethernet switch to connect the ADAM-6200 module with its target device. Do not use an Ethernet hub. This can prevent data packet collision.*

After you decide the target device, then you can choose the output action by the Operation Type combo box. The default setting is No Operation, meaning there is no output action. You can choose analog output (AO), digital output (DO), counter channel setting (DI_counter), pulse output (DO_Pulse), local timer (Timer), local or remote internal flag (AuxFlag), remote message output (RemoteMessage) and local internal counter setting (Counter) as the output action.

After you have chosen the target device and output action, you can click the Verify button to check if the target device exists and supports GCL feature to execute the output action. (If you choose No Operation as output action, it will not check.) After you choose the appropriate output action and complete all related setting, click the OK button. That Output stage icon will change its pattern to present the current condition. We will describe each output action in more detail below.

Analog Output (AO)

Below are the steps to configure analog output:

1. Choose correct model name by the Target module combo box in the Operation area. If the Destination is Local, you don't need to choose the model name.
2. Choose the appropriate output range by the TargetRange combo.
3. Define which channel is responsible to generate output signal on the target device by the Channel combo box.
4. Define what value is generated by the Value text box. (The unit of the value depends on the range in the TargetRange combo box.)
5. Click the OK button to complete the configuration.

Note! *You can see the action description by the Action text box. When the logic result value passed from Execution stage is logic True, the selected analog output channel will generate the new value you defined. When the logic result value passed from Execution stage is logic False, the selected analog output channel will not change its output value.*

Digital Output (DO)

Below are the steps to configure digital output:

1. Choose correct model name by the Target module combo box in the Operation area. If the Destination is Local, you don't need to choose the model name.
2. Define to generate True or False digital output signal for the true action (When the logic result value passed from Execution stage is logic True) by the True Action combo box.
3. The false action (When the logic result value passed from Execution stage is logic False) is displayed from the False Action text box and will automatically be set according to the true action. The false action will be opposite to the true action. For example, when you choose False in the True Action combo box, the False Action text box will automatically display True.
4. Define which channel is responsible to generate output signal on the target device by the Channel combo box.
5. Click the OK button to complete the configuration.

Counter Channel Setting (DI_Counter)

Below are the steps to configure counter channel setting:

1. Choose correct model name by the Target module combo box in the Operation area. If the Destination is Local, you don't need to choose the model name.
2. Define what action (start the counter by Start, or stop the counter by Stop, or reset the counter by Reset) will be taken for the true action (When the logic result value passed from Execution stage is logic True) by the True Action combo box.
3. The false action (When the logic result value passed from Execution stage is logic False) is displayed by the False Action text box and will automatically be set according to the true action. Define which counter channel is responsible to take the defined action by the Channel combo box.
4. Click the OK button to complete the configuration.

Pulse Output (DO_Pulse)

Below are the steps to configure pulse output:

1. Choose correct model name by the Target module combo box in the Operation area.
2. Define what action (continuous generate pulse train by Continue, stop pulse generation by Stop, or only generate finite number of pulse by Num of pulse) will be taken for the true action (When the logic result value passed from Execution stage is logic True) from the True Action combo box.
3. The false action (When the logic result value passed from Execution stage is logic False) is displayed by the False Action text box and will always be Keep current status, meaning there is no action change for the selected digital output channel.
4. Define which digital output channel is responsible to take the defined action (start or stop pulse generation) by the Channel combo box. 5. If you choose Num of pulse in the True Action combo box, type the number of pulse you want to generate in the Value text box.
5. Click the OK button to complete the configuration.

Local Timer (Timer)

There are 16 local timers on ADAM-6200 module. Here, you can define the timer action depending on the logic result value from the Execution stage. After you have chosen Timer in the Operation type combo box, select the interested timer by the Index combo box in the Operation area. (From timer 0 to timer 15) Then you can define the timer action by the Type combo box in the Operation area.

When you choose ON-Delay, the timer will start to count the time passed when the logic result value passed from the Execution stage is logic True, while the timer will stop counting and reset its value to zero when the logic result value is logic False. If you choose OFF-Delay, the true action and

false action will be opposite: the timer will start to count the time passed when the logic result value is logic False, while it will stop counting and rest its value to zero when the logic result value is logic True. The action description is displayed by the True/False act text box. After you have completed the setting, click the OK button.

Local or Remote Internal Flag (AuxFlag)

You can assign the logic result value from the Execution stage, to local or remote internal flag. Select the appropriate internal flag by the Index combo box. Define what value you want to assign to the internal flag for the true action (When the logic result value passed from Execution stage is logic True) by the True Action combo box. The false action (When the logic result value is logic False) is displayed by the False Action text box and will automatically be opposite to the value in the True Action. After you have completed the setting, click the OK button.

Remote Message Output (RemoteMessage)

We can send the Device Description (you can edit it by the Message text box in the Operation area) as message to the target device. When you have several logic rules which all will send message, we need to know the message received by the target device is sent by which logic rule. So, you can give the message a index, defined by the Value text box in the Operation area. Click OK to complete the configuration. So when logic result value passed from the Execution stage is true, the message will be sent to the target device. If the logic result is False, the message won't be sent out.

Local Internal Counter Setting (Counter)

Below are the steps to configure internal counter setting:

1. Define what action (increase one count to internal counter value by Positive edge trigger (F->T), or reset internal counter by Reset) will be taken for the true action (When the logic result value passed from Execution stage is logic True) by the True Action combo box.

   Note! *When you choose Positive edge trigger (F->T) as action, the counter will only add one count for the first time that the logic result value from Execution Stage is logic high. After the first time, the counter value will not change even the logic result value from Execution Stage is still logic high. So it is the reason why it is called positive edge trigger.*

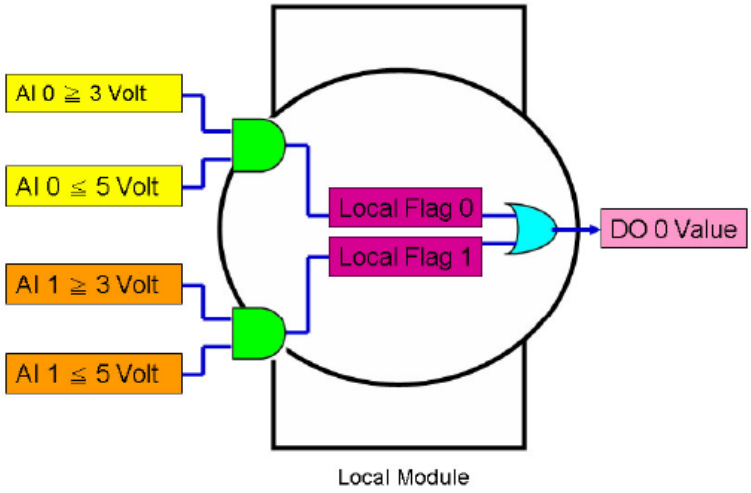
2. The false action (When the logic result value passed from Execution stage is logic False) is displayed by the False Action text box and will automatically be set according to the true action. The false action will be opposite to the true action. Refer to the table below to see the relationship between true action and false action.
3. Define which counter channel is responsible to take the defined action by the Channel combo box.
4. Click the OK button to complete the configuration.

The following table will show the logic true and false of output actions.

Output Action	True action (the logic result value from the Execution stage is logic **True**)	False action (the logic result value from the Execution stage is logic **False**)
AO	Change the analog output value	Keep current status
DO	Output True value	Output False value
	Output False value	Output True value
Counter Channel (DI_Counter)	Start counter counting	Stop counter counting
	Stop counter counting	Start counter counting
	Reset counter	Do nothing
Pulse Output	Generate continuous pulse train	Keep current status
	Generate finite pulses	
	Stop pulse generation	
Timer	Start counting time	Stop counting time and reset timer value to zero
	Stop counting time and reset timer value to zero	Start counting time
Internal Flag	Assign True value to flag	Assign False value to flag
	Assign False value to flag	Assign True value to flag
Remote Message	Send message to target device	Do nothing
Internal Counter Setting (Counter)	Increase 1 count to counter	Do nothing
	Reset counter	

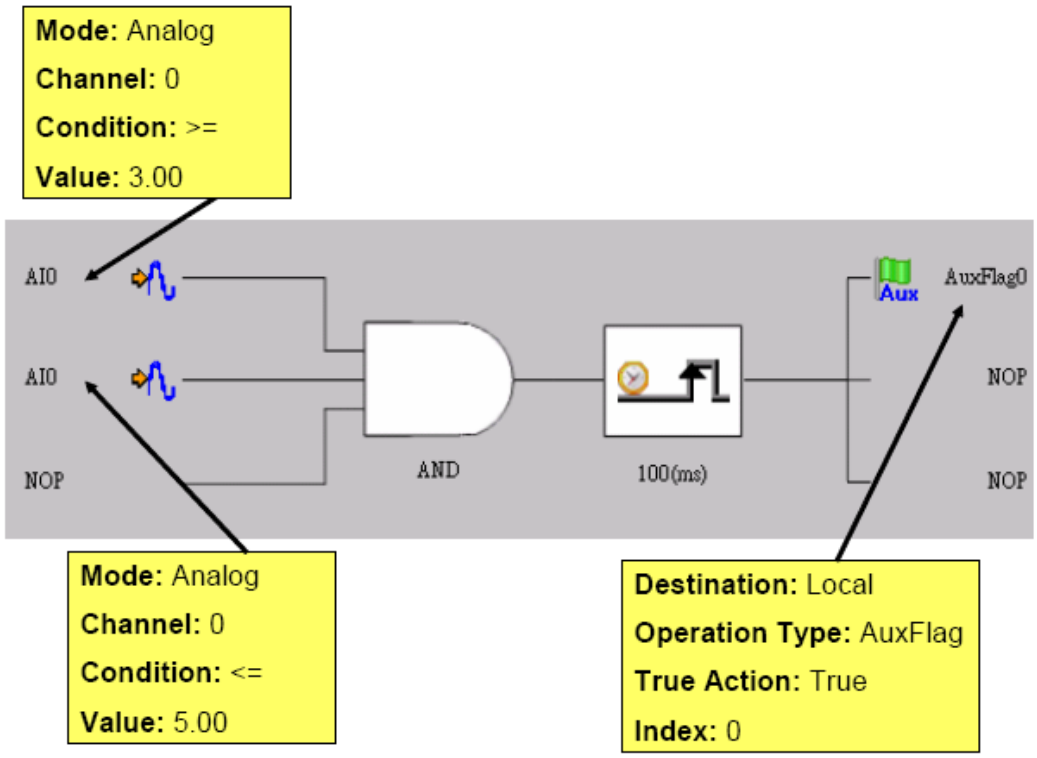
Internal Flag for Logic Cascade and Feedback

- Logic Cascade
  Using internal flag as interface, you can combine different logic together to form a new single logic rule which can play more complex logic architecture. You can combine logic rules on the same module, or even on different modules. Please refer to example below to understand how the internal flag works.
- Local Logic Cascade
  Here, we take one simple example to describe the logic cascade. We use two analog input channels (channel 0 and channel 1) of ADAM-6217 to measure signal from sensors. As long as either of the two input channel read voltage between 3 ~ 5 Volt, digital channel 0 should generate logic high value. Otherwise, the digital channel 0 should generate logic low value. The logic architecture will look like as below.

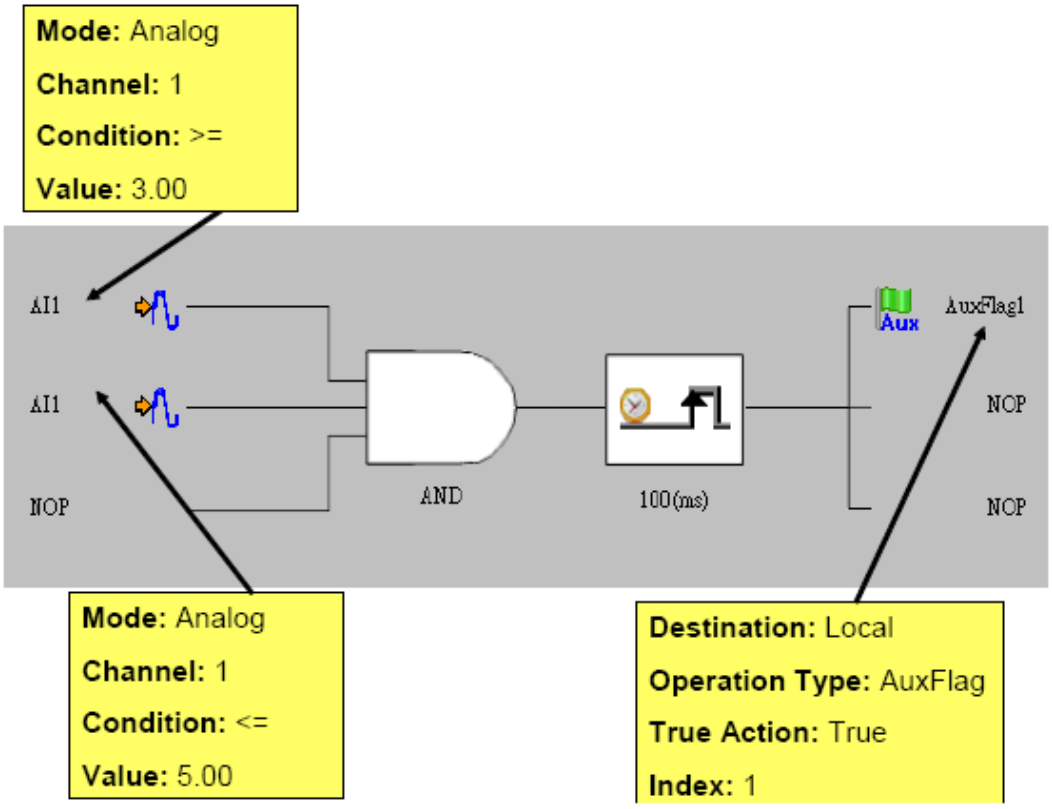


In order to implement this logic architecture, we need to use three logic rule and two internal flag to achieve this. Refer to the following diagrams to see how to configure the three logic rules.

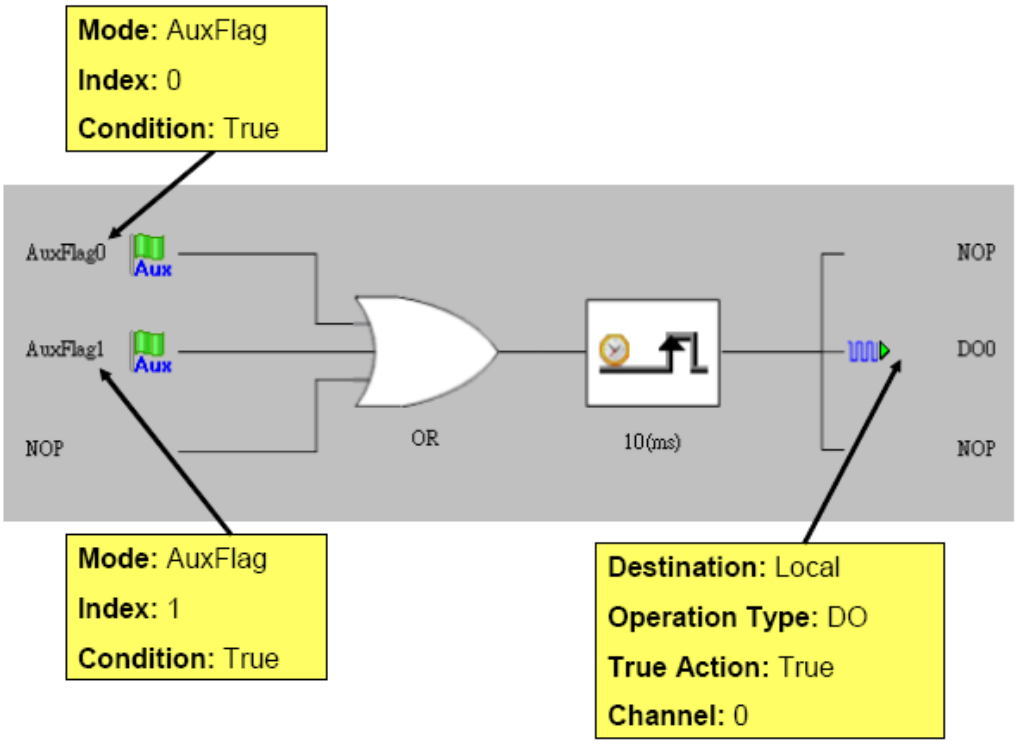
Rule 1.


Mode: Analog
Channel: 0
Condition: >=
Value: 3.00
AI0
AI0
NOP
AND
100(ms)
Aux
AuxFlag0
NOP
NOP
Mode: Analog
Channel: 0
Condition: <=
Value: 5.00
Destination: Local
Operation Type: AuxFlag
True Action: True
Index: 0

Rule 2.


Mode: Analog
Channel: 1
Condition: >=
Value: 3.00
AI1
AI1
NOP
AND
100(ms)
Aux
AuxFlag1
NOP
NOP
Mode: Analog
Channel: 1
Condition: <=
Value: 5.00
Destination: Local
Operation Type: AuxFlag
True Action: True
Index: 1

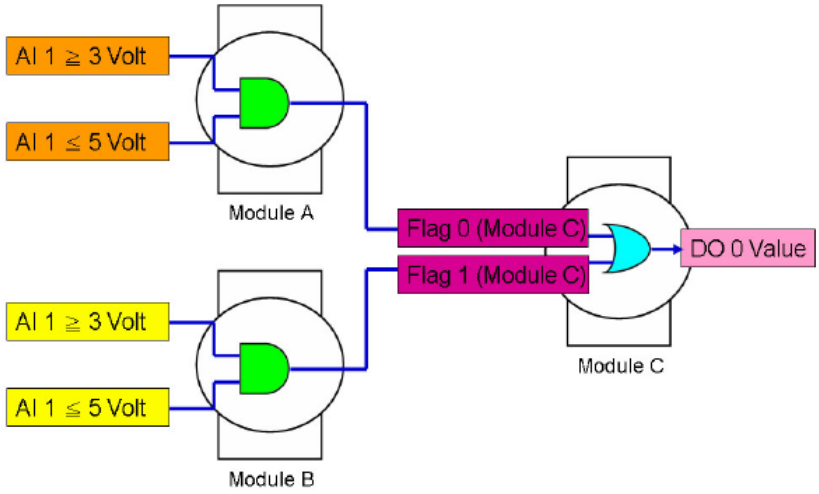
Rule 3.



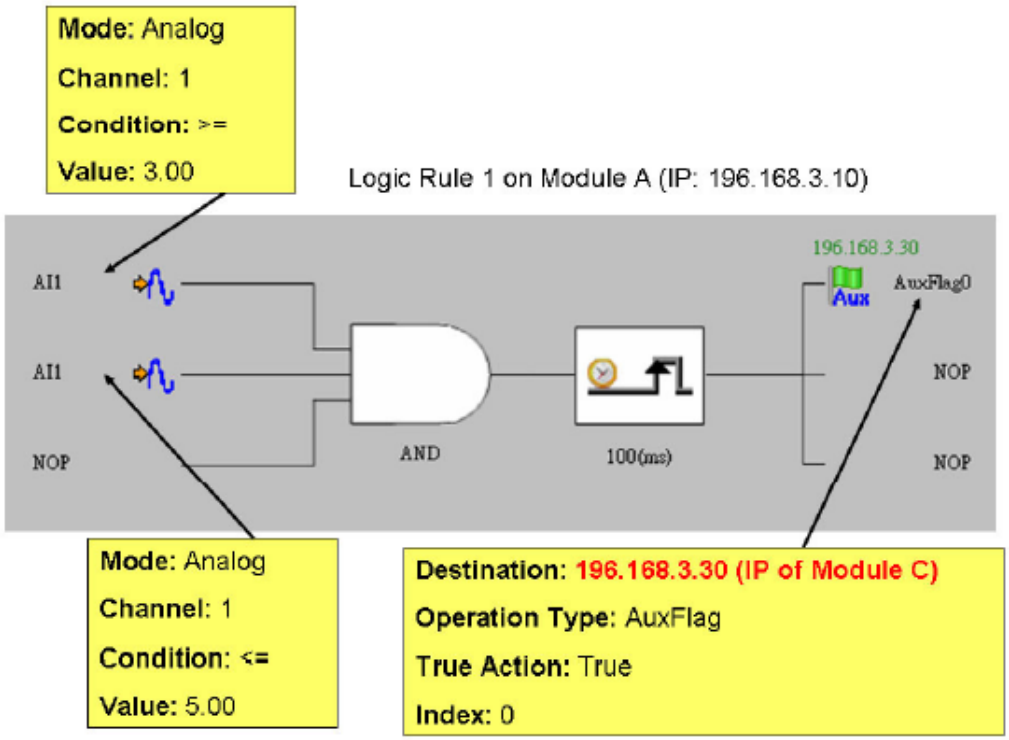
We use the logic rule 1 to check if AI channel 0 value of the ADAM-6217 is within 3 ~ 5 Volt. Logic rule 2 is used to check if AI channel 1 value is within 3 ~ 5 Volt. The comparison result of logic rule 1 and 2 is assigned to internal flag 0 and 1. The logic rule 3 read the value of these two internal flags and use the OR logic operation to define the output of digital output channel 0. You can find that we have built the logic architecture as shown by the internal flag.

Distributed Logic Cascade

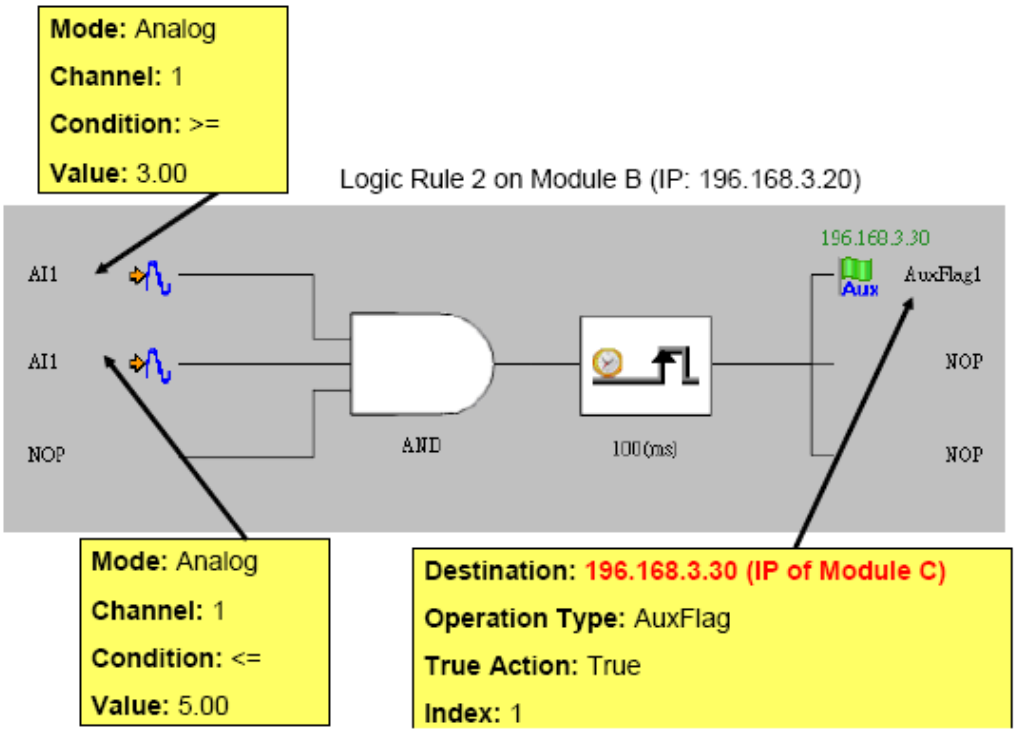
Logic Cascade function is not limited on one single module. Since you can define the internal flag on another module, the logic cascade structure can be across different modules. Take the previous application as example, now you can define the logic rule 1, 2, 3 are running on module A, B and C. Then the logic structure becomes across three ADAM-6200 modules, and we call it Distributed Logic Cascade as refer to below logic architecture and the configuration for the three logic rules.



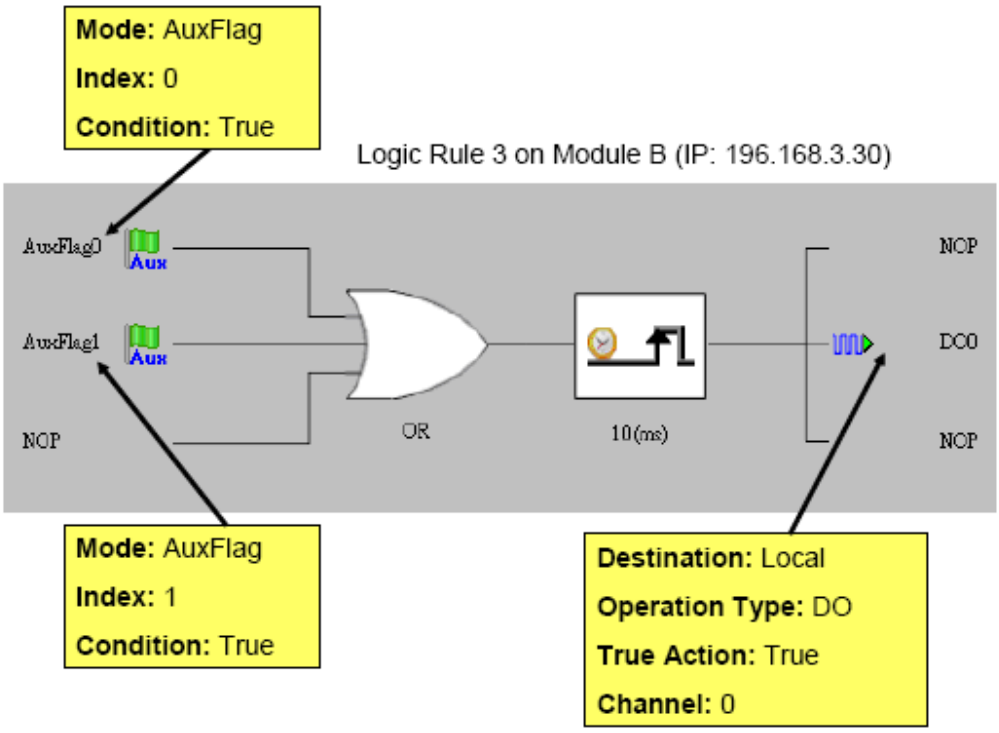
Rule 1.



Rule 2.



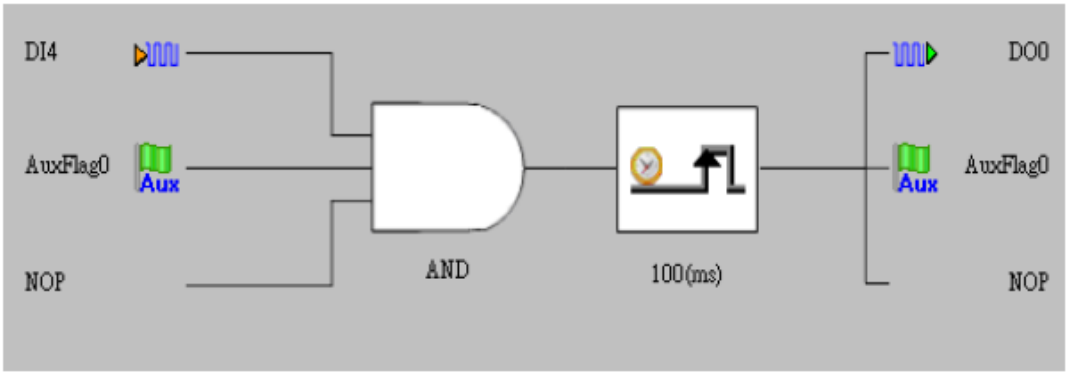
Rule 3.



Using Local or Distributed Logic Cascade architecture, there will be no limitation for input numbers of logic rules. And you can build any logic architecture to meet your application requirement.

Feedback

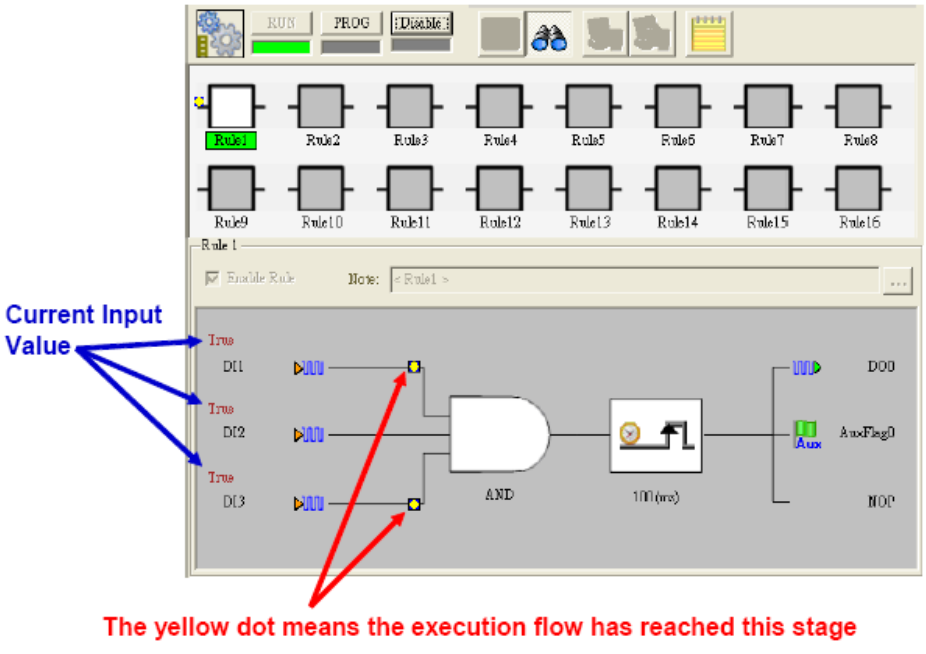
When you choose the same internal flag for the input condition and output of one single logic rule, the logic rule has logic feedback ability. See as below. In this example, one input condition and one output are dedicated to the same internal flag 0 (AuxFlag 0). So the output value in current execution will become the input of the next execution. This gives this logic rule feedback ability.



Export logic file and online monitoring

After you have completed all the configurations for GCL logic rule, click the Download Project button in the GCL Menu area. All the configuration will be downloaded to the target device. Then you can click the Run GCL button in the GCL Menu area to execute the project on the target module. You can see the Current Status icon become the Running mode.

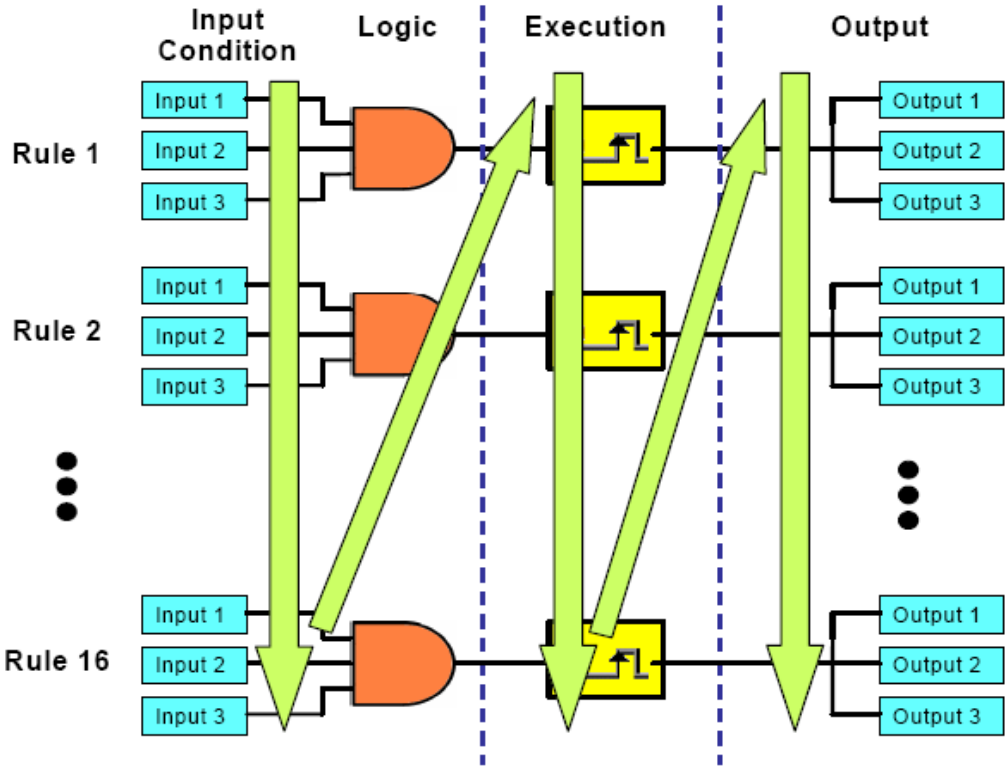
ADAM-6200 Module features special Online Monitoring function. In the Running mode, you can click the Monitoring button in the GCL Menu area to enable this function. Then you can see the execution situation on the Individual Logic Rule Configuration area. The yellow dot appears when the execution flow proceed to that stage. Besides, you can see the current input value besides the Input Condition icon. In this example, you can see the input condition D1, D2 and D3 has been satisfied, so the yellow dot appears next to the two Input Condition icons. And you can see the current input value showing at the top of the three Input Stage icons.



Note! *When you use Internal Flags (AuxFlag) as the inputs of GCL logic rules, you can dynamically change the flag values in the online monitoring window of ADAM.NET Utility. Simply double click the input icons represented the internal flag, and you can see the flag values change from True to False, or from False to True.*

GCL Rule Execution Sequence

There are 16 logic rules on one ADAM-6200 module. As below, you can see the execution flow for one cycle and there are 3 groups for one cycle: Input Condition + Logic, Execution, and Output .All the Input Condition + Logic stages of rules which are enabled will execute sequentially first. Then, all the Execution stages of rules which are enabled will be executed in sequence. In the end, all Output stages of rules which are enabled will be executed in order. For some advanced applications, you may combine different rules together by Logic Cascade architecture. For example, the output of rule 1 is connected to the input of rule 2, by assigning to the same internal flag. Based on the execution flow mentioned above, the Input Condition + Logic, Execution and Output stages of rule 1 will be executed sequentially. Therefore, the output of rule 1 will be updated at the last stage (Output stage) in the first cycle, and the input of rule 2 can detect the change of output of rule 1 in the next cycle.



GCL Execution and Data Transfer Performance

1. Local Output
   Condition: Running 1 logic rule on one ADAM-6250 module
   Processing time: < 1 milliseconds
   (Processing time includes hardware input delay time, 1 logic rule execution time and hardware output delay time)
   If multiple logic rules are used, the processing time can be estimated by equation below:
   - logic rule number n ≤16

   Approximate Processing Time (for one cycle) = 600 + n*370 (μs)
2. Remote Output
   Condition: Running 1 logic rule on one ADAM-6250, output is on another ADAM-6250 module through one Ethernet Switch Processing + Communication time: < 3 milliseconds

Typical Applications with GCL

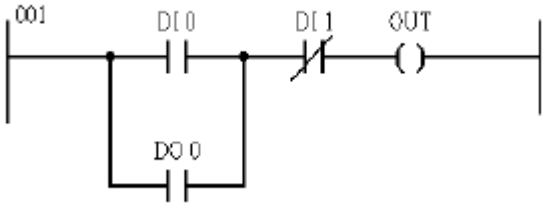
In order to shorten GCL configuration time, Advantech has provided several example project files for some typical applications. You can find these example project files on the CD with the ADAM module. Simply load these example project files by clicking the Project Content button of GCL Menu bar. You can modify an example project based on your application requirements. Then you can download the modified project to your module and execute it. We will introduce each example project file in more detail below:

1. Empty Project

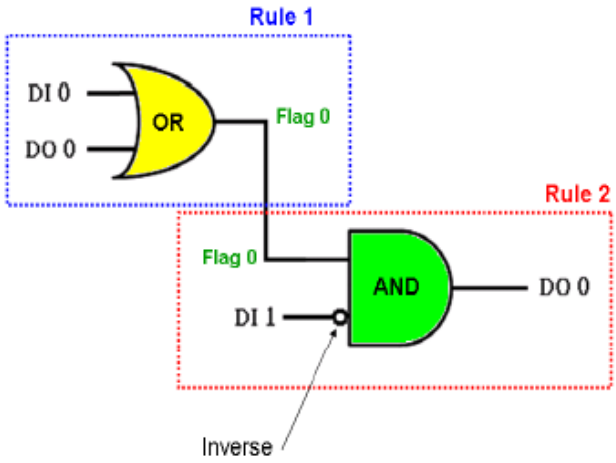
When you want to clear all configurations for GCL, it is simple to load this example project. Then you don't need to clear all the configurations manually.

2. On/Off Control (Two buttons to control On and Off Separately)

In some automation applications, two digital inputs (DI 0 and DI 1) are used to control one digital output status (DO 0). The DO status will become logic high when DI 0 is logic high, and the DO status will return to logic low when DI 1 is logic high. For example, motor operation is controlled by two buttons. When the first button is pressed, the motor is started. If the second button is pressed, the motor will be stopped immediately. PLCs are typically used for this kind of industrial automation application, and the ladder diagram will look like below figure:



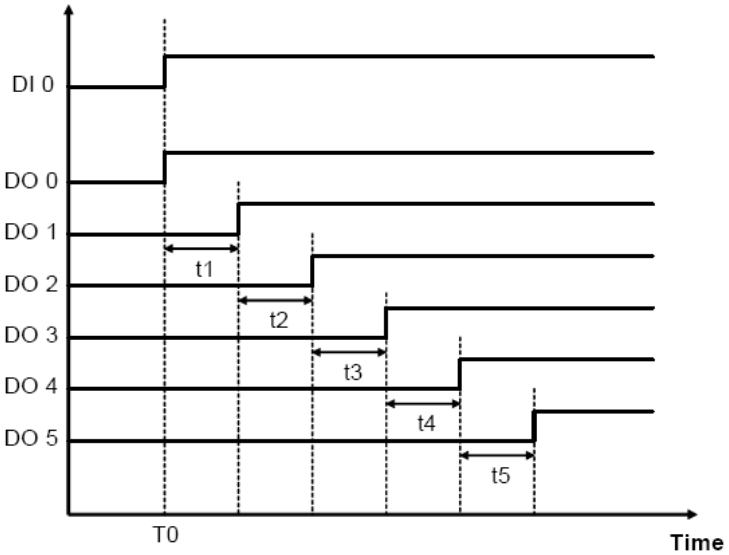
Now, we can use GCL logic to achieve the same control operation. Two logic rules are used. The complete logic architecture is shown as below.



After you load the example project file, you can find that it uses rule 1 and rule 2. One output of rule 1 and one input of rule 2 are assigned to the same internal flag: Flag 0. This can combine 2 or more logic rules together, that we call it Logic Cascade. We have configured the condition for DI 1 input as False, so the DI input value is inverted before entering the AND operator of rule 2. The GCL logic architecture is similar to the PLC ladder diagram.

3. Sequential Control (Turn On in Sequence and Remain On)

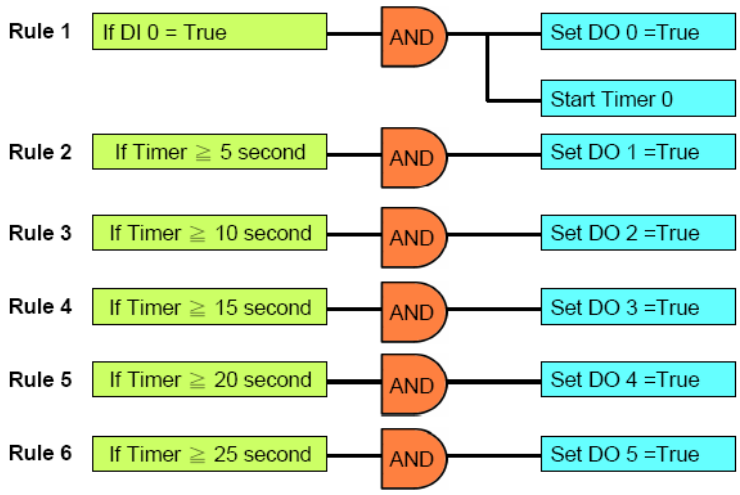
In this kind of application, several digital outputs will be activated in sequence and latch their values. In the example project we provide, DO 0 ~ DO 5 will sequentially be controlled to change their status. The time chart for this application can be shown as below:



(Turns On in Sequence and Remains On)

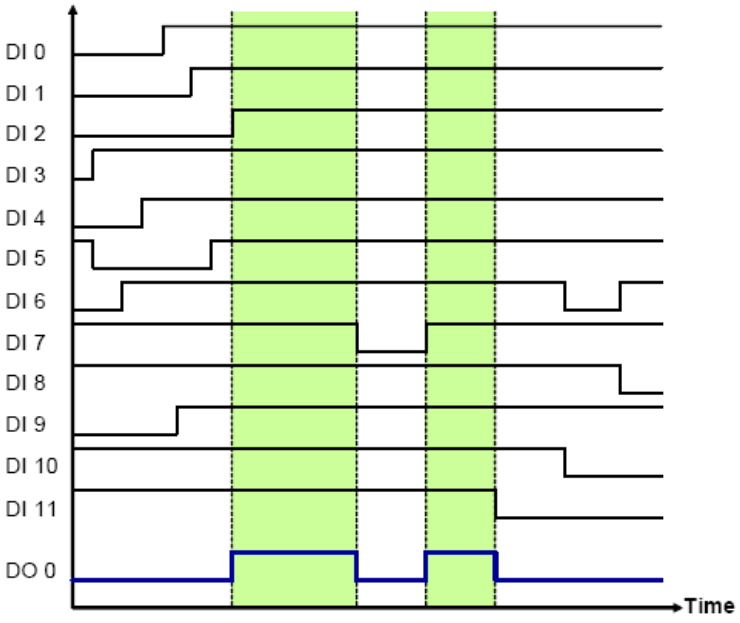
In the example project, DI 0 is used as a trigger to start the sequential control action. Therefore, when DI 0 turn to logic high (at the moment T0), DO 0 will also change to logic high immediately. Then, DO1 ~ DO5 will sequentially be activated to logic high after a specific time interval. You can decide the time interval t1 ~ t5 (They can be different values). In this example project, t1 ~ t5 are all 5 seconds.

We can use 6 logic rules and 1 internal timer for this GCL application. In the first logic rule, DI 0 is used to trigger Timer 0 and DO 0. Since the timer has been triggered, it will start counting time and DO 1 ~ DO 5 will be activated after a specific amount of time has elapsed.. The GCL architecture can be shown by as below:



4. Multiple DI to control one DO (12 DI to 1 DO)

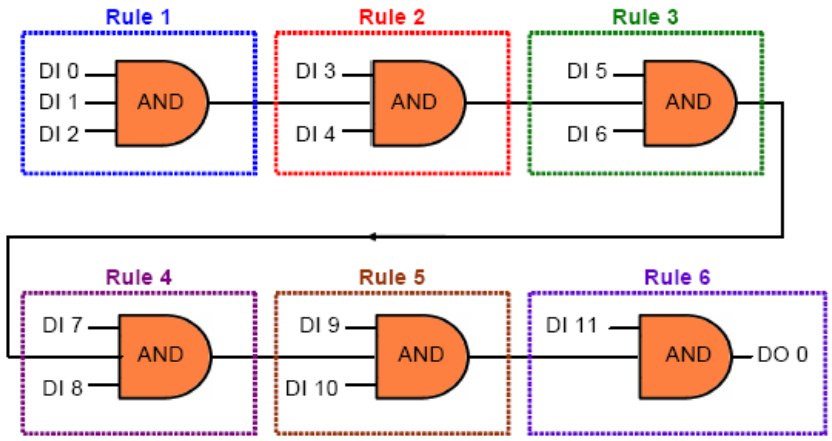
In many applications, only when multiple digital inputs are logic high (related conditions are all satisfied), digital output status will become logic high. In the example project we provide, only when DI 0 ~ DI 11 are all logic high at the same time, the DO 0 will become logic high. The time chart of this application can be shown as below. The green band area indicates the moment that all 12 DI are logic high, thus DO 0 will be logic high. For other time interval, there is at least one DI channel whose value is not logic high, so DO 0 value is logic low.



You can simply implement one AND logic operator to achieve this control system. However, since one logic rule only has three inputs, we need to use Logic Cascade function to have 12 inputs. There are two ways to achieve Logic Cascade:

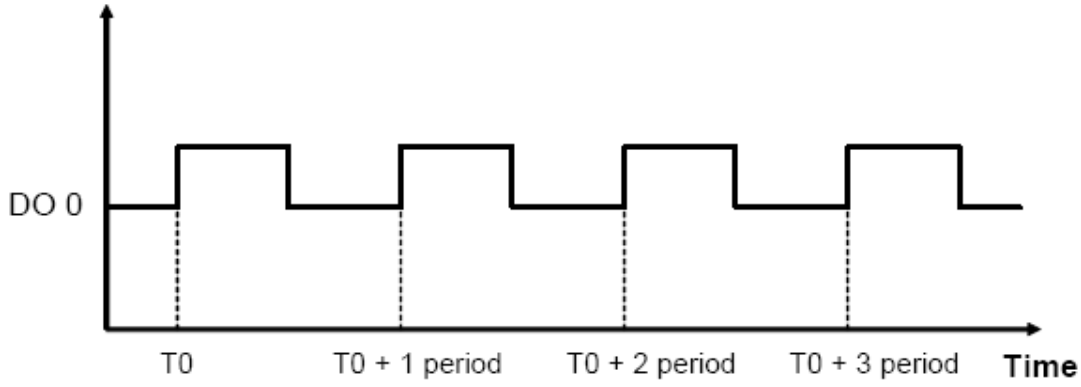
a. Select SendtoNextRule in Execution Stage of one logic rule. It will combine this logic rule to the next logic rule.
b. Assign output of one logic rule and input of another logic rule to the same internal flag, combining the two logic rules together.

With the first method, the two logic rules must be next to each other. For example, rule 1 and rule 2 can be combined together. But you cannot combine rule 1 and rule 3. There is no such limitation if you use the second method to combine two different logic rules. Using the second method, you can even combine two rules on different modules together. Here, this example project adapts the first method for Logic Cascade. So the GCL logic architecture can be shown as below:



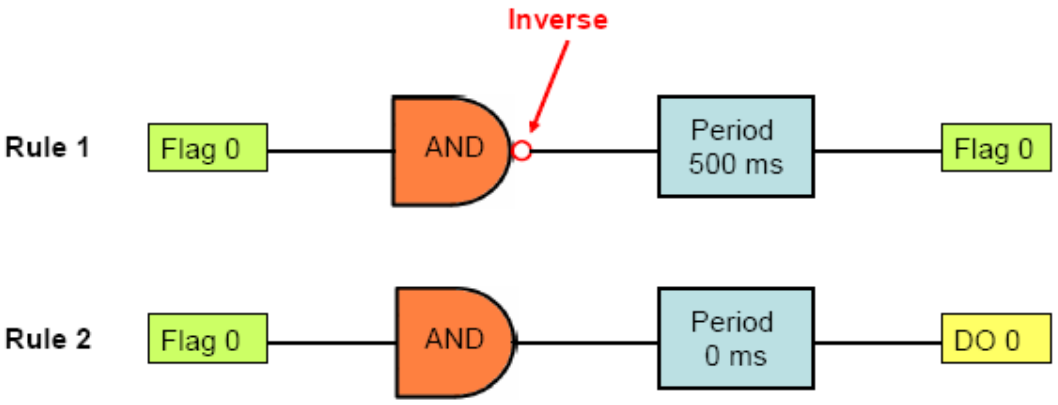
5. Flicker

Flicker is commonly used in automation control application. Typical example is we want to make the alarm flashing, controlled by digital output. This application requires a continuous pulse train by a digital output channel and we can decide the period of the pulse train. The time chart for flicker application can be shown as below:



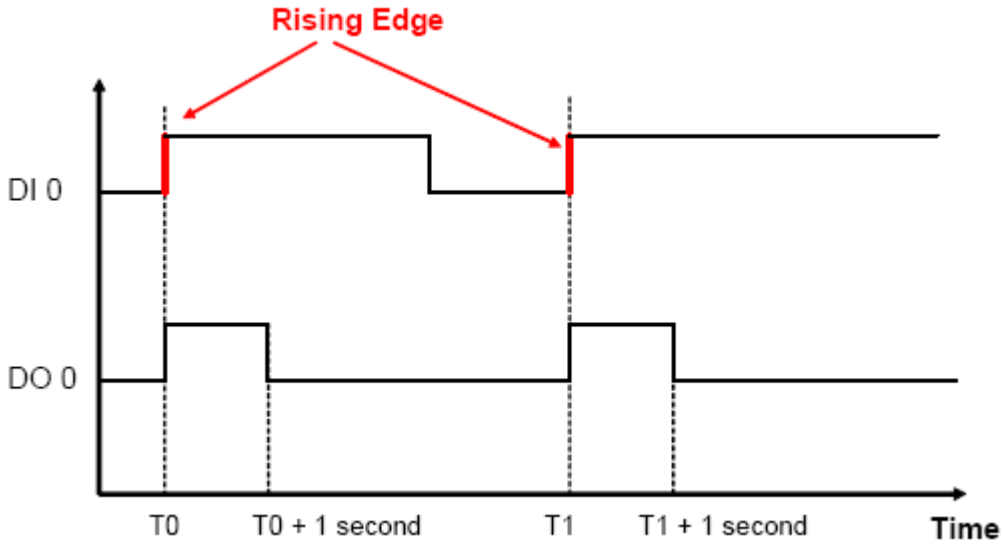
We need to use 1 Internal Flag (Flag 0) and 2 logic rules for the Flicker application described above. In logic rule 1, the value of Flag 0 is inverted (By choosing NAND in the Logic stage) periodically. (Here, it is 0.5 second) The period is defined by the Execution_Period in the Execution Stage. The status of DO 0 is controlled by Flag 0

in logic rule 2, so DO 0 will change every 0.5 second. The GCL logic rule architecture is shown as below:

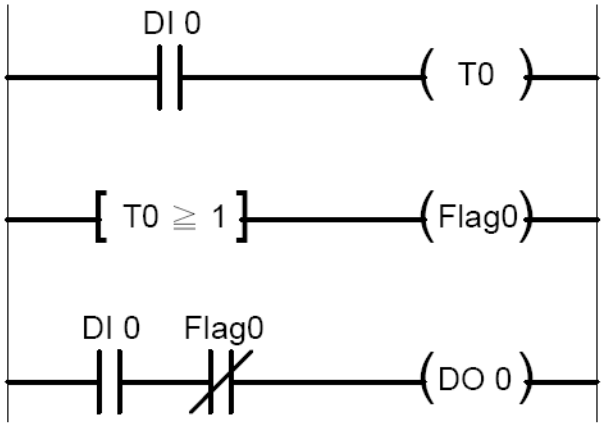


6. Rising Edge

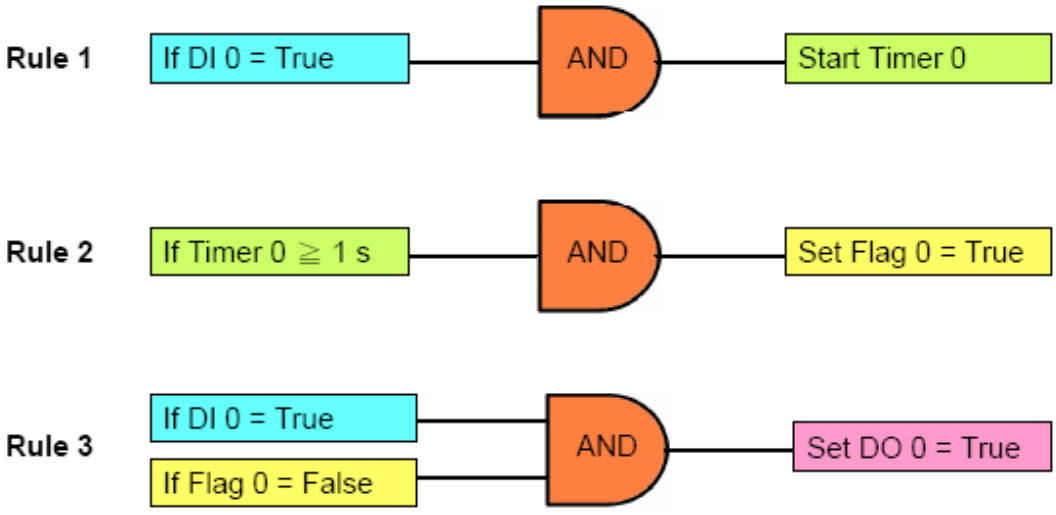
For Rising Edge application, the DO status will be activated to logic high, when DI value is changed from logic low to logic high (it is so-called rising edge). But the DO value won't continuously remain logic high. Instead, after a specific time interval (in the example, it is 1 second), the DO value will return to logic low. Refer below for its time chart:



You can see that DO 0 will only be triggered when rising edge of DI 0 occurs. In the example project we provide, the DO status will remain logic high for 1 second. Then it will back to logic low. When PLC is used for this kind of application, the ladder diagram will be likely as below:

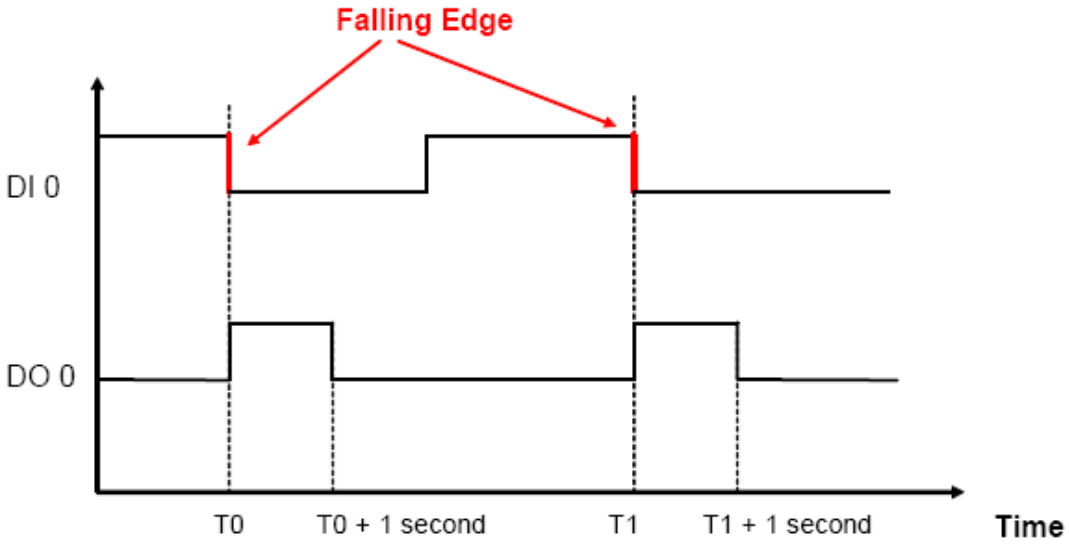


When you use GCL to achieve rising edge application, 3 logic rules, 1 Internal Timer (Timer 0) and 1 Internal Flag (Flag 0) are needed. For example, with logic rule 3, DO 0 value is controlled by DI 0 and Flag 0. Flag 0 is initially set as False. When rising edge occurs (DI value changes from logic low to logic high), DO will be activated (logic rule 3 are satisfied), and Timer 0 starts to count time (logic rule 1 are satisfied). After Timer 0 counts up to the specific time interval (1 second), Flag 0 will become logic True by logic rule 2, making DO 0 value logic low (logic rule 3 are not satisfied). The GCL architecture is similar to the ladder diagram.

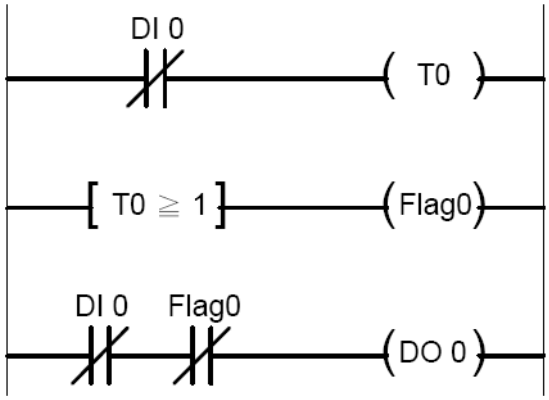


7. Falling Edge

For Falling Edge application, the DO value will be activated to logic high, when DI value is changing from logic high to logic low (it is so-called falling edge). But the DO value won't continuously remain logic high. Instead, after a specific period (in the example project, it is 1 second), the DO value will back to logic low. The time chart is shown as below:

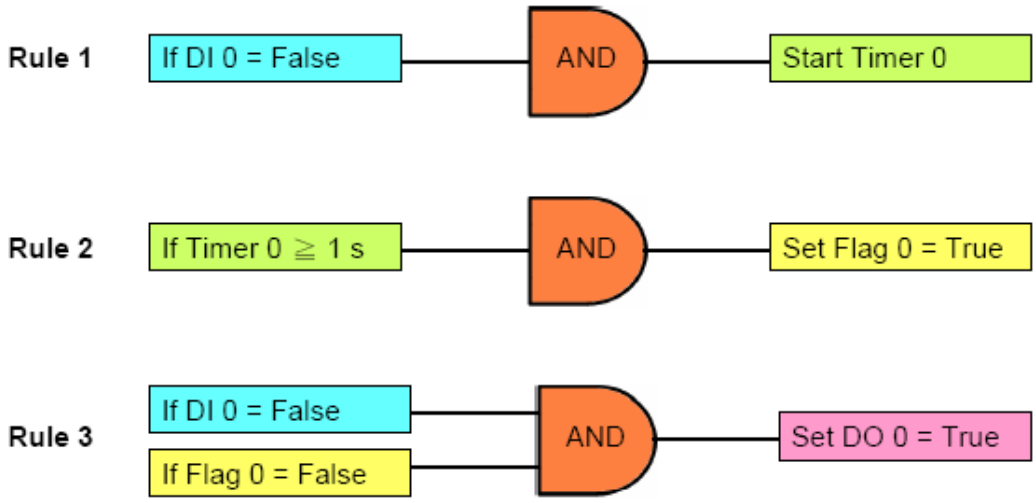


You can see the DO 0 will only be triggered when falling edge of DI 0 occurs. In the example project we provide, the DO status will remain logic high for 1 second. Then it will back to logic low. When PLC is used for this kind of application, the ladder diagram will be likely as below:



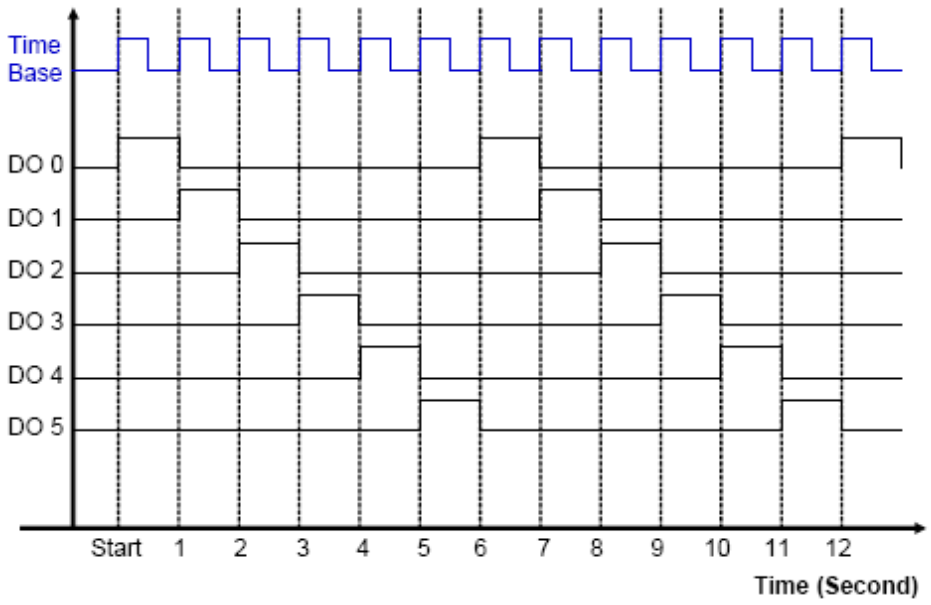
When you use GCL to achieve falling edge application, 3 logic rules, 1 Internal Timer (Timer 0) and 1 Internal Flag (Flag 0) are needed. For example, with logic rule 3, DO 0 value is controlled by DI 0 and Flag 0. Flag 0 value is logic False at beginning.

When falling edge occurs (DI value changes from logic high to logic low), DO will be activated (logic rule 3 are satisfied), and Timer 0 starts to count time (logic rule 1 are satisfied). After Timer 0 counts up to the specific time interval (1 second), Flag 0 will become logic True by logic rule 2, making DO 0 value logic low (logic rule 3 are not satisfied). The GCL architecture is similar to the ladder diagram.



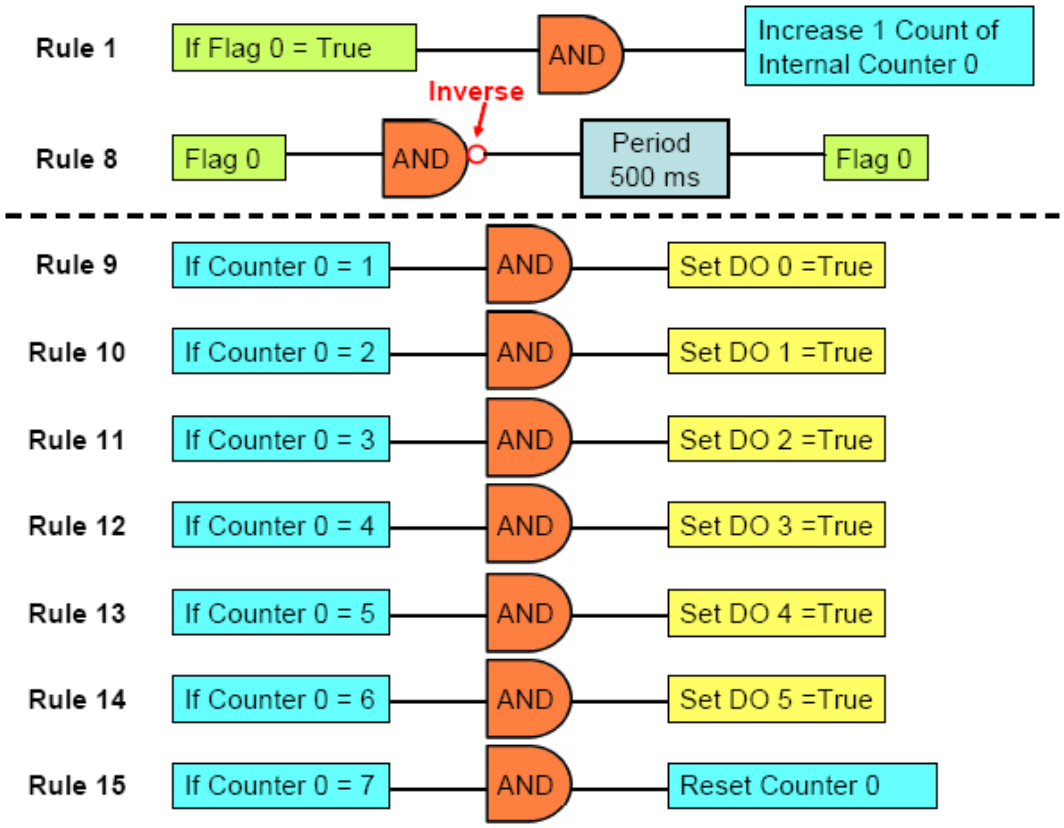
8. Sequential Control (Turn On and Off in Sequence Continuously)

This type of automation application is similar to the 3rd application we have introduced. They are both sequential control applications. For example 3, DO channel will keep its value after it is turned on. In example 8 here, after DO channel is turned on, it will be turned off after a specific time. You can see the time chart for this application as below. There is one time base needed to control the digital output sequential action. In this example, the period of the time base to turn off one DO and turn on the next-door DO is 1 second.



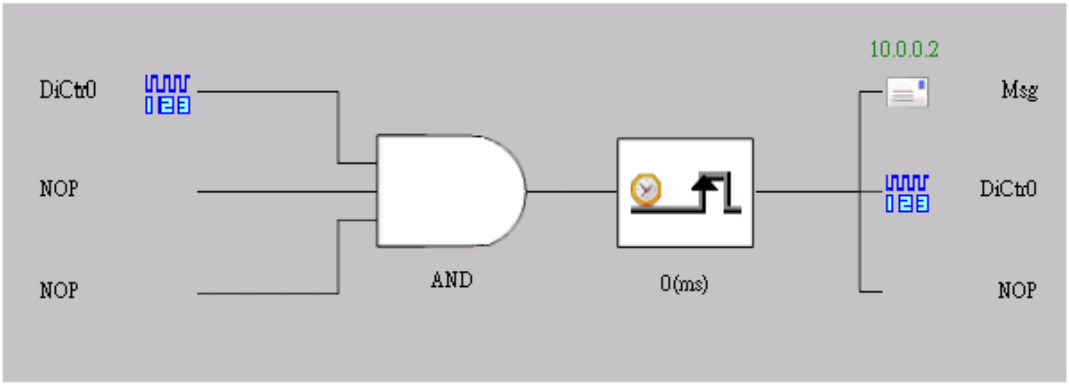
In order to implement this kind of application, 9 logic rules, 1 Internal Counter (Counter 0) and 1 Internal Flag (Flag 0) are used. In the example project we provide, logic rule 1 and 8 are used to create the time base. By logic rule 8, Flag 0 value will change every 0.5 second. In logic rule 1, once the Flag 0 value is logic high, the Counter 0 will increase 1 unit. So every 1 second, Counter 0 will increase 1 unit, making Counter 0 the time base.

Logic rules 9 ~ 14 are used to control DO 0 ~ 5. Which logic rule should be executed is based on Counter 0 value. Since Counter 0 value will continuously add 1 unit every 1 second, logic rules 9 ~ 14 will be executed in sequence every 1 second. Therefore, DO 0 ~ DO 5 will be activated sequentially in 1 second. When logic rule 15 is executed, Counter 0 will reset and its value will back to zero. So it makes the logic rules execution become a continuous loop. You can refer its GCL architecture as below.

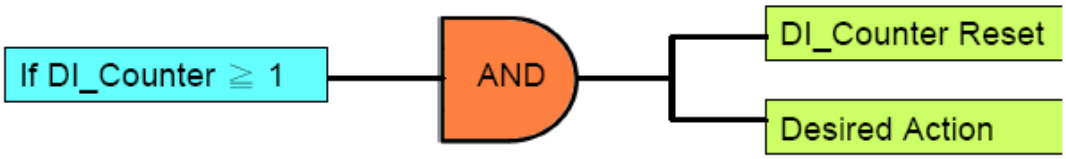


9. DI Event Trigger (Only Occurs Once)

We can simply use GCL to perform Event trigger. For this kind of application, a DI channel is used to trigger some action. So, the input condition of GCL logic rule will be if the DI value is logic True, and output of the rule can be some desired action, such as sending remote message. When the DI value becomes logic True, the input condition is satisfied. The GCL logic rule will send message continuously until the DI value backs to logic False. However, it is not what we plan. (We don't want the message is sent out continuously. Instead, we want the message will only be sent once at the first moment that the condition is satisfied.) This kind of application can be simply achieved by using one counter input channel. You can refer to below figure. Select local counter input channel (DI_Counter) in the Input condition for one logic rule. There are two output used for the same logic rule, one is reset the counter input channel, and another is the desired action you want. Then, when the counter input channel detect DI signal, the condition is satisfied and the desired action will be done. At the same time, the GCL rule reset the counter input channel. So the desired action will only be executed once.



The true image of configuration of GCL logic rule in ADAM.NET Utility can be shown as below. (Here, the desired action is to send remote message.)



4.3.6 Remotely Monitoring and Control via Web Server

As mentioned in feature highlight section, this new feature will bring obvious benefit to user in maintenance anywhere over the Ethernet in the local field. Now the following paragraph will show how to build up this circumstance and use it. This feature not only works on PC or laptop, it can also work on portable devices, like smart phone or pad.

Preset conditions:

1. Finish the installation and network setting of ADAM-6200 module
2. ADAM-6200 is connected to your local Ethernet network

Operating steps: (Take smart phone as example)

1. Make you smart phone connect to your local Ethernet network via wireless and then open the browser of your smart phone.

Note!



ADAM-6200 module is developed by public HTML 5 base, but for detailed indication and data transmission mode may be different on Web page of the operating system.

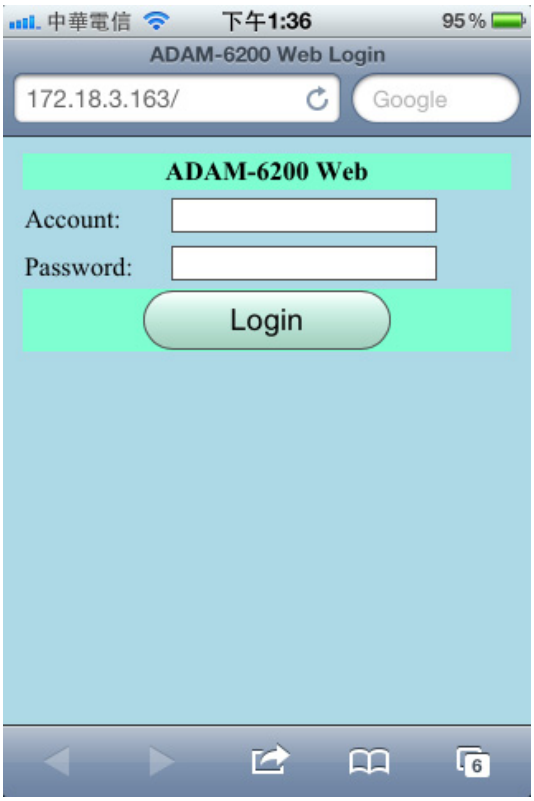
For smart phone system, the minimum requirement of Web browsers as below:

- *Safari 5 in Apple iOS*
- *Web Browser in Google Android 4.0 (Ice Cream Sandwich)*
- *Chrome in Google Android 4.0 (Ice Cream Sandwich)*

Otherwise, if where you operate is PC platform, you can refer to the minimum requirement of Web browsers as below:

- *Internet Explorer (version 9)*
- *Google Chrome (version 8)*
- *Mozilla Firefox (version 10)*

2. Enter IP address of ADAM-6200 module
   Please enter the IP address in URL field of the browser, for example: 172.18.3.163.
   You will see the login page (shown as below) on the screen.

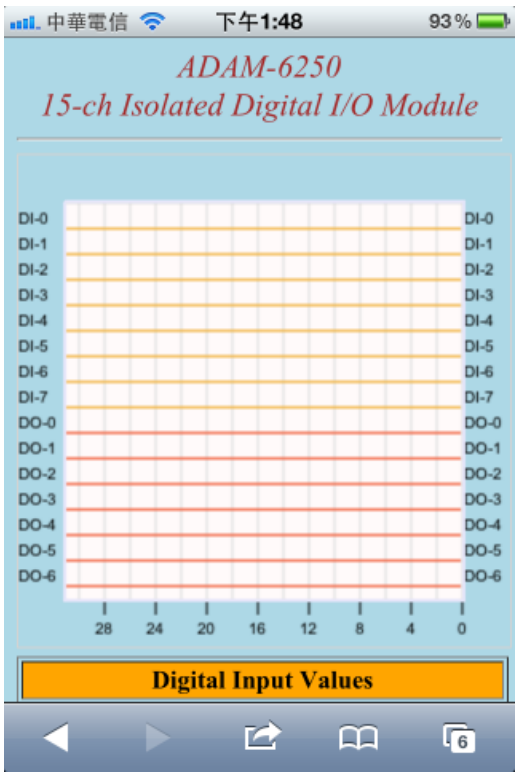


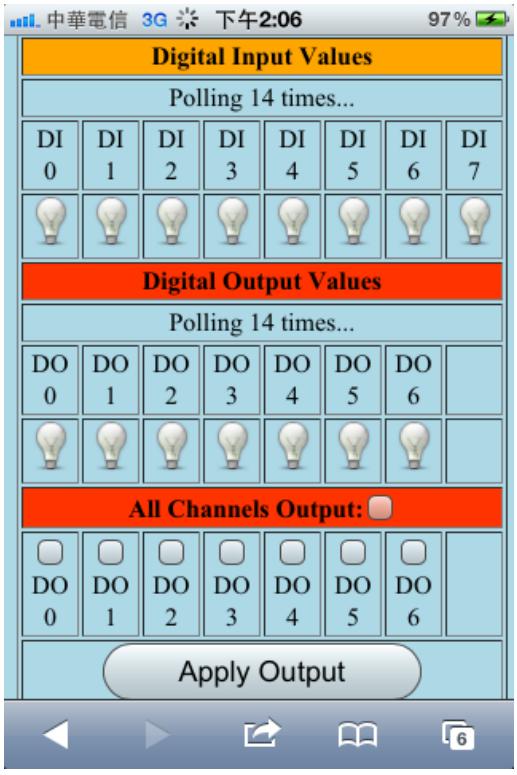
3. Enter the account and password.

Note! *The default setting of account is "root", and password is "00000000"*

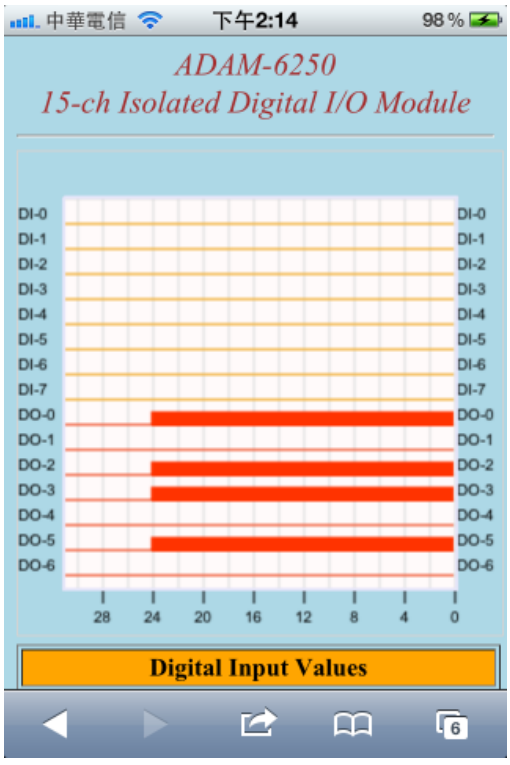


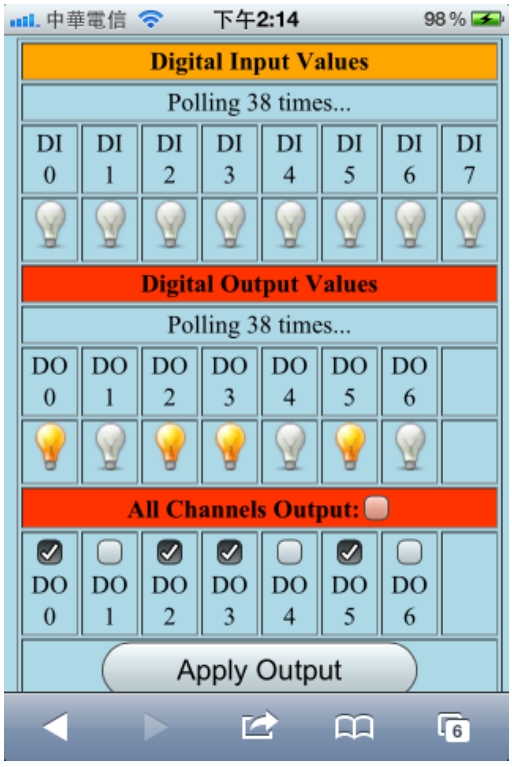
4. Click Login button, you will see the operation page. In this page, you will not only to monitor I/O status (trend log), but also simply enable output setting below.





5. For example, if you check on DO 0, 2, 3, and 5, then click “Apply Output”, you will see the bulb is lit on and trend log is also changed.





Appendix A

I/O Modbus Mapping Table

A.1 I/O Modbus Mapping Table

A.1.1 Modbus Function Code Introduction

To full-fill the programming requirement, there is a series of function code standard for user's reference.

Code (Hex)	Name	Usage
01	Read Coil Status	Read Discrete Output Bit
02	Read Input Status	Read Discrete Input Bit
03	Read Holding Registers	Read 16-bit register. Used to read integer or floating point process data.
04	Read Input Registers	
05	Force Single Coil	Write data to force coil ON/OFF
06	Preset Single Register	Write data in 16-bit integer format
08	Loopback Diagnosis	Diagnostic testing of the communication port
0F	Force Multiple Coils	Write multiple data to force coil ON/OFF
10	Preset Multiple Registers	Write multiple data in 16-bit integer format

A.1.2 ADAM-6217 8-ch Isolated Analog Input Module

Address (0X):

Address (0X)	Channel	Description	Attribute
00101	0	Reset Historical Max. Value to present value	Write
00102	1		Write
00103	2		Write
00104	3		Write
00105	4		Write
00106	5		Write
00107	6		Write
00108	7		Write
00109	Average Ch0 - 7		Write

00111	0	Reset Historical Min. Value to present value	Write
00112	1		Write
00113	2		Write
00114	3		Write
00115	4		Write
00116	5		Write
00117	6		Write
00118	7		Write
00119	Average Ch0 - 7		Write

00121	0	Open-Circuit Flag (Burnout)	Read
00122	1		Read
00123	2		Read
00124	3		Read
00125	4		Read
00126	5		Read
00127	6		Read
00128	7		Read

00131	0	High Alarm Flag	Read
00132	1		Read
00133	2		Read
00134	3		Read
00135	4		Read
00136	5		Read
00137	6		Read
00138	7		Read
00139	Average Ch0 - 7		Read

00141	0	Low Alarm Flag	Read
00142	1		Read
00143	2		Read
00144	3		Read
00145	4		Read
00146	5		Read
00147	6		Read
00148	7		Read
00149	Average Ch0 - 7		Read

Address (4X):

Address (4X)	Channel	Description	Attribute
40001	0	AI Value	Read
40002	1		Read
40003	2		Read
40004	3		Read
40005	4		Read
40006	5		Read
40007	6		Read
40008	7		Read
40009	Average Ch0 - 7		Read

40011	0	Historical Max. AI Value	Read
40012	1		Read
40013	2		Read
40014	3		Read
40015	4		Read
40016	5		Read
40017	6		Read
40018	7		Read
40019	Average Ch0 - 7		Read

40021	0	Historical Min. AI Value	Read
40022	1		Read
40023	2		Read
40024	3		Read
40025	4		Read
40026	5		Read
40027	6		Read
40028	7		Read
40029	Average Ch0 - 7		Read

40101 - 40102	0	AI Status	Read
40103 - 40104	1		Read
40105 - 40106	2		Read
40107 - 40108	3		Read
40109 - 40110	4		Read
40111 - 40112	5		Read
40113 - 40114	6		Read
40115 - 40116	7		Read

40201	0	Type Code	R/W
40202	1		R/W
40203	2		R/W
40204	3		R/W
40205	4		R/W
40206	5		R/W
40207	6		R/W
40208	7		R/W
40209	Average Ch0 - 7		Read

40221	All	AI Channel Enable	R/W
40305	0 - 15	GCL Internal Flag Value	R/W

Note:

1. Users can configure the High alarm value in the ADAM.NET utility. When AI value is higher than the High alarm, this bit will be 1.
2. Users can configure the Low alarm value in the ADAM.NET utility. When AI value is lower than the Low alarm, this bit will be 1.
3. The value definition of AI Status

Bit	Description
0	Failed to provide AI value (UART timeout)
1	Over Range
2	Under Range
3	Open Circuit (Burnout)
7	AD Converter failed
9	Zero/Span Calibration Error

4. Type code table

Input Range	Type Code (HEX)
+/- 10V	0x0143
+/- 5 V	0x0142
+/- 1 V	0x0140
+/- 500 mV	0x0104
+/- 150 mV	0x0103
+/- 20 mA	0x0181
0 ~ 20 mA	0x0182
4 ~ 20 mA	0x0180

A.1.3 ADAM-6224 4-ch Isolated Analog Output Module

Address (0X):

Address (0X)	Channel	Description	Attribute
00001	0	DI Value	Read
00002	1		Read
00003	2		Read
00004	3		Read

Address (4X):

Address (4X)	Channel	Description	Attribute
40001	0	AO Value	R/W
40002	1		R/W
40003	2		R/W
40004	3		R/W
40101 – 40102	0	AO Status	Read
40103 – 40104	1		Read
40105 – 40106	2		Read
40107 – 40108	3		Read

40111	0	DI Event Status	Read
40112	1		Read
40113	2		Read
40114	3		Read

40201	0	Type Code	R/W
40202	1		R/W
40203	2		R/W
40204	3		R/W

40301	All	DI Value	Read
40305	0 – 15	GCL Internal Flag Value	R/W

40411	0	Safety Value	R/W
40412	1		R/W
40413	2		R/W
40414	3		R/W

40401	0	Startup Value	R/W
40402	1		R/W
40403	2		R/W
40404	3		R/W

Note:

1. The value definition of DI Event Status

Bit	Description
0	Unreliable DI value (UART Timeout)
1	Safety Value triggered
2	Startup Value triggered

2. The value definition of AO Status

First register

Bit	Description
0	Fail to provide AO Value
3	No Output Current)
9	Zero/Span Calibration Error

Second register

Bit	Description
0	DI triggered to Safety Value
1	DI triggered to Startup Value
2	AO triggered to Fail Safety Value

3. Type Code Table

Input Range	Type Code (HEX)
0 ~ 20 mA	0x0182
4 ~ 20 mA	0x0180
0 ~ 10 V	0x0148
0 ~ 5 V	0x0147
+/- 10 V	0x0143
+/- 5 V	0x0142

A.1.4 ADAM-6250 15-ch Digital Input/Output Module

Address (0X):

Address (0X)	Channel	Description	Attribute
00001	0	DI Value	Read
00002	1		Read
00003	2		Read
00004	3		Read
00005	4		Read
00006	5		Read
00007	6		Read
00008	7		Read
00017	0	DO Value	R/W
00018	1		R/W
00019	2		R/W
00020	3		R/W
00021	4		R/W
00022	5		R/W
00023	6		R/W
00033	0	Counter Start (1) / Stop (0)	R/W
00034	1		R/W
00035	2		R/W
00036	3		R/W
00037	4		R/W
00038	5		R/W
00039	6		R/W
00040	7		R/W
00041	0	Clear Counter (1)	Write
00042	1		Write
00043	2		Write
00044	3		Write
00045	4		Write
00046	5		Write
00047	6		Write
00048	7		Write

00049	0	Clear Overflow	R/W
00050	1		R/W
00051	2		R/W
00052	3		R/W
00053	4		R/W
00054	5		R/W
00055	6		R/W
00056	7		R/W

00057	0	DI Latch Status (1)	R/W
00058	1		R/W
00059	2		R/W
00060	3		R/W
00061	4		R/W
00062	5		R/W
00063	6		R/W
00064	7		R/W

Address (4X):

Address (4X)	Channel	Description	Attribute
40001 – 40002	0	Counter/Frequency Value	Read
40003 – 40004	1		Read
40005 – 40006	2		Read
40007 – 40008	3		Read
40009 – 40010	4		Read
40011 – 40012	5		Read
40013 – 40014	6		Read
40015 – 40016	7		Read

40017 – 40018	0	Pulse Output Low Level Width	R/W
40019 – 40020	1		R/W
40021 – 40022	2		R/W
40023 – 40024	3		R/W
40025 – 40026	4		R/W
40027 – 40028	5		R/W
40029 – 40030	6		R/W

40031 – 40032	0	Pulse Output High Level Width	R/W
40033 – 40034	1		R/W
40035 – 40036	2		R/W
40037 – 40038	3		R/W
40039 – 40040	4		R/W
40041 – 40042	5		R/W
40043 – 40044	6		R/W

40045 – 40046	0	Set Absolute Pulse	R/W
40047 – 40048	1		R/W
40049 – 40050	2		R/W
40051 – 40052	3		R/W
40053 – 40054	4		R/W
40055 – 40056	5		R/W
40057 – 40058	6		R/W

40059 – 40060	0	Set Incremental Pulse	R/W
40061 – 40062	1		R/W
40063 – 40064	2		R/W
40065 – 40066	3		R/W
40067 – 40068	4		R/W
40069 – 40070	5		R/W
40071 – 40072	6		R/W

40301	All	DI Value	Read
40303	All	DO Value	R/W
40305	0 - 15	GCL Internal Flag Value	R/W

A.1.5 ADAM-6251 16-ch Digital Input Module

Address (0X):

Address (0X)	Channel	Description	Attribute
00001	0	DI Value	Read
00002	1		Read
00003	2		Read
00004	3		Read
00005	4		Read
00006	5		Read
00007	6		Read
00008	7		Read
00009	8		Read
00010	9		Read
00011	10		Read
00012	11		Read
00013	12		Read
00014	13		Read
00015	14		Read
00016	15		Read

00033	0	Counter Start (1) / Stop (0)	R/W
00034	1		R/W
00035	2		R/W
00036	3		R/W
00037	4		R/W
00038	5		R/W
00039	6		R/W
00040	7		R/W
00041	8		R/W
00042	9		R/W
00043	10		R/W
00044	11		R/W
00045	12		R/W
00046	13		R/W
00047	14		R/W
00048	15		R/W

00049	0	Counter Start (1) / Stop (0)	Write
00050	1		Write
00051	2		Write
00052	3		Write
00053	4		Write
00054	5		Write
00055	6		Write
00056	7		Write
00057	8		Write
00058	9		Write
00059	10		Write
00060	11		Write
00061	12		Write
00062	13		Write
00063	14		Write
00064	15		Write

00065	0	Clear Overflow	R/W
00066	1		R/W
00067	2		R/W
00068	3		R/W
00069	4		R/W
00070	5		R/W
00071	6		R/W
00072	7		R/W
00073	8		R/W
00074	9		R/W
00075	10		R/W
00076	11		R/W
00077	12		R/W
00078	13		R/W
00079	14		R/W
00080	15		R/W

00081	0	DI Latch Status	R/W
00082	1		R/W
00083	2		R/W
00084	3		R/W
00085	4		R/W
00086	5		R/W
00087	6		R/W
00088	7		R/W
00089	8		R/W
00090	9		R/W
00091	10		R/W
00092	11		R/W
00093	12		R/W
00094	13		R/W
00095	14		R/W
00096	15		R/W

Address (4X):

Address (4X)	Channel	Description	Attribute
40001 – 40002	0	Counter/Frequency Value	Read
40003 – 40004	1		Read
40005 – 40006	2		Read
40007 – 40008	3		Read
40009 – 40010	4		Read
40011 – 40012	5		Read
40013 – 40014	6		Read
40015 – 40016	7		Read
40017 – 40018	8		Read
40019 – 40020	9		Read
40021 – 40022	10		Read
40023 – 40024	11		Read
40025 – 40026	12		Read
40027 – 40028	13		Read
40029 – 40030	14		Read
40031 – 40032	15		Read

40301	All	DI Value	Read
40305	0 - 15	GCL Internal Flag Value	R/W

A.1.6 ADAM-6256 16-ch Digital Output Module

Address (0X):

Address (0X)	Channel	Description	Attribute
00017	0	DO Value	R/W
00018	1		R/W
00019	2		R/W
00020	3		R/W
00021	4		R/W
00022	5		R/W
00023	6		R/W
00024	7		R/W
00025	8		R/W
00026	9		R/W
00027	10		R/W
00028	11		R/W
00029	12		R/W
00030	13		R/W
00031	14		R/W
00032	15		R/W

Address (4X):

Address (4X)	Channel	Description	Attribute
40001 – 40002	0	Pulse Output Low Level Width	R/W
40003 – 40004	1		R/W
40005 – 40006	2		R/W
40007 – 40008	3		R/W
40009 – 40010	4		R/W
40011 – 40012	5		R/W
40013 – 40014	6		R/W
40015 – 40016	7		R/W
40017 – 40018	8		R/W
40019 – 40020	9		R/W
40021 – 40022	10		R/W
40023 – 40024	11		R/W
40025 – 40026	12		R/W
40027 – 40028	13		R/W
40029 – 40030	14		R/W
40031 – 40032	15		R/W

40033 – 40034	0	Pulse Output High Level Width	R/W
40035 – 40036	1		R/W
40037 – 40038	2		R/W
40039 – 40040	3		R/W
40041 – 40042	4		R/W
40043 – 40044	5		R/W
40045 – 40046	6		R/W
40047 – 40048	7		R/W
40049 – 40050	8		R/W
40051 – 40052	9		R/W
40053 – 40054	10		R/W
40055 – 40056	11		R/W
40057 – 40058	12		R/W
40059 – 40060	13		R/W
40061 – 40062	14		R/W
40063 – 40064	15		R/W

40065 – 40066	0	Set Absolute Pulse	R/W
40067 – 40068	1		R/W
40069 – 40070	2		R/W
40071 – 40072	3		R/W
40073 – 40074	4		R/W
40075 – 40076	5		R/W
40077 – 40078	6		R/W
40079 – 40080	7		R/W
40081 – 40082	8		R/W
40083 – 40084	9		R/W
40085 – 40086	10		R/W
40087 – 40088	11		R/W
40089 – 40090	12		R/W
40091 – 40091	13		R/W
40093 – 40092	14		R/W
40095 – 40096	15		R/W

40097 – 40098	0	Set Incremental Pulse	R/W
40099 – 40100	1		R/W
40101 – 40102	2		R/W
40103 – 40104	3		R/W
40105 – 40106	4		R/W
40107 – 40108	5		R/W
40109 – 40110	6		R/W
40111 – 40112	7		R/W
40113 – 40114	8		R/W
40115 – 40116	9		R/W
40117 – 40118	10		R/W
40119 – 40120	11		R/W
40121 – 40122	12		R/W
40123 – 40124	13		R/W
40125 – 40126	14		R/W
40127 – 40128	15		R/W

40303	All	DO Value	R/W
40305	0 – 15	GCL Internal Flag Value	R/W

A.1.7 ADAM-6260 6-ch Relay Output Module

Address (0X):

Address (0X)	Channel	Description	Attribute
00017	0	DO Value	R/W
00018	1		R/W
00019	2		R/W
00020	3		R/W
00021	4		R/W
00022	5		R/W

Address (4X):

Address (4X)	Channel	Description	Attribute
40001 – 40002	0	Pulse Output Low Level Width	R/W
40003 – 40004	1		R/W
40005 – 40006	2		R/W
40007 – 40008	3		R/W
40009 – 40010	4		R/W
40011 – 40012	5		R/W
40013 – 40014	0	Pulse Output High Level Width	R/W
40015 – 40016	1		R/W
40017 – 40018	2		R/W
40019 – 40020	3		R/W
40021 – 40022	4		R/W
40023 – 40024	5		R/W
40025 – 40026	0	Set Absolute Pulse	R/W
40027 – 40028	1		R/W
40029 – 40030	2		R/W
40031 – 40032	3		R/W
40033 – 40034	4		R/W
40035 – 40036	5		R/W
40037 – 40038	0	Set Incremental Pulse	R/W
40039 – 40040	1		R/W
40041 – 40042	2		R/W
40043 – 40044	3		R/W
40045 – 40046	4		R/W
40047 – 40048	5		R/W
40303	All	DO Value	R/W
40305	0 - 15	GCL Internal Flag Value	R/W

A.1.8 ADAM-6266 4-ch Relay Output Module with 4-ch DI

Address (0X):

Address (0X)	Channel	Description	Attribute
00001	0	DI Value	Read
00002	1		Read
00003	2		Read
00004	3		Read
00017	0	DO Value	R/W
00018	1		R/W
00019	2		R/W
00020	3		R/W
00033	0	Counter Start (1) / Stop (0)	R/W
00034	1		R/W
00035	2		R/W
00036	3		R/W
00037	0	Clear Counter (1)	Write
00038	1		Write
00039	2		Write
00040	3		Write
00041	0	Clear Overflow	R/W
00042	1		R/W
00043	2		R/W
00044	3		R/W
00045	0	DI Latch Status	R/W
00046	1		R/W
00047	2		R/W
00048	3		R/W

Address (4X):

Address (4X)	Channel	Description	Attribute
40001 – 40002	0	Counter/Frequency Value	Read
40003 – 40004	1		Read
40005 – 40006	2		Read
40007 – 40008	3		Read
40009 – 40010	0	Pulse Output Low Level Width	R/W
40011 – 40012	1		R/W
40013 – 40014	2		R/W
40015 – 40016	3		R/W
40017 – 40018	0	Pulse Output High Level Width	R/W
40019 – 40020	1		R/W
40021 – 40022	2		R/W
40023 – 40024	3		R/W
40025 – 40026	0	Set Absolute Pulse	R/W
40027 – 40028	1		R/W
40029 – 40030	2		R/W
40031 – 40032	3		R/W
40033 – 40034	0	Set Absolute Pulse	R/W
40035 – 40036	1		R/W
40037 – 40038	2		R/W
40039 – 40040	3		R/W
40301	All	DI Value	Read
40303	All	DO Value	R/W
40305	0 - 15	GCL Internal Flag Value	R/W

Appendix B

ADAM-6200 ASCII Commands

B.1 ASCII Commands for ADAM-6200 modules

For users do not familiar to Modbus protocol, Advantech offers a function library as a protocol translator, integrating ASCII command into Modbus/TCP structure. Therefore, users familiar to ASCII command can access ADAM-6200 module easily. Before explaining the structure of ASCII command packed with Modbus/TCP format. Let's see how to use an ASCII command and how many are available for your program.

The command set is divided into the following categories:

- System Command Set
- Analog Input Command Set
- Analog Input Alarm Command Set
- Digital I/O Command Set

Every command set category starts with a command summary of the particular type of module, followed by datasheets that give detailed information about individual commands. Although commands in different subsections sometime share the same format, the effect they have on a certain module can be completely different than that of another. Therefore, the full command sets for each type of modules are listed along with a description of the effect the command has on the given module.

B.1.1 System Command Set

Function	Description	Command	Example
Read module name	Return the module name from a specified module	Cmd: $aaM(cr) Ret: !aannnn(cr)	Send: $01M(cr) Receive: !016250(cr)
Read Firmware Version	Return the firmware version from a specified module	Cmd: $aaF(cr) Ret: !aa nn.nn Bnn(cr)	Send: $01F(cr) Receive: !01 A1.00 B01(cr)
Write GCL Internal Flags	Write value(s) to GCL internal flag(s) on a specific ADAM-6200 Module	Cmd:#aaVd-bbbbddddddd(cr) Ret: >aa(cr)	Send: #01Vd000000000000(cr) Receive: >01(cr)
Read GCL Internal Flags	Read all GCL internal flags' values from a specific ADAM-6200 module	Cmd: #aaVd(cr) Ret: >aaddddddddd(cr)	Send: $01Vd(cr) Receive: >0100000000(cr)

Note!

1. Command $aaM and $aaF support all ADAM-6200 I/O modules.
2. Command #aaVdbbbbddddddddd supports all ADAM-6200 I/O modules.
3. Command #aaVd supports all ADAM-6200 I/O modules.

$aaM

Name	Read Module Name
Description	Returns the module name from a specified module.
Syntax	$aaM(cr) $ is a delimiter character. aa (range 00-FF) means the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) M is the Module Name command. (cr) is the terminating character, carriage return (0Dh).
Response	!aa60bb(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error, communication error or if the address does not exist. ! delimiter indicating a valid command was received. ? delimiter indicating the command was in-valid. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. bb (range 00-FF) represents the 2-character model number of an ADAM-6200 module. (cr) is the terminating character, carriage return (0Dh).
Example	The command requests the system at address 01h to send its module name. The system at address 01h responds with module name 6250 indicating that there is an ADAM-6250 at address 01h. Command: $01M(cr) Response: !016250(cr)

$aaF

Name	Read Firmware Version
Description	Returns the firmware version from a specified module.
Syntax	$aaF(cr)
	$ is a delimiter character.
	aa (range 00-FF) means the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01)
	F is the Firmware Version command.
	(cr) is the terminating character, carriage return (0Dh).
Response	!aa(version)(cr) if the command is valid.
	?aa(cr) if an invalid operation was entered.
	There is no response if the module detects a syntax error, communication error or if the address does not exist.
	! delimiter indicating a valid command was received.
	? delimiter indicating the command was invalid.
	aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module.
	(version) represents the firmware version of the module.
	(cr) is the terminating character, carriage return (0Dh).
Example	The command requests the system at address 01h to send its firmware version.
	The system responds with firmware version 1.01.
ASCII	Command: $01F(cr)
	Response: !01 A1.00B01(cr)

#aaVdbbbbdddddddd

Name	Write Value(s) to GCL Internal Flags (Auxiliary Flags)
Description	This command sets a single or all GCL internal flag(s) on the specific ADAM-6200 module.
Syntax	#aaVdbbbbdddddddd(cr)
	# is a delimiter character.
	aa (range 00-FF) represents the 2-character hexadecimal slave network address of the ADAM-6200 module. (Always 01)
	Vd is the GCL Internal Flag command.
	bbbb is used to indicate which GCL internal flag(s) to set.
	Writing to all GCL internal flags: 0000
	Writing to a single GCL internal flag: First character is 1, and 2 ~4 characters indicate the GCL internal flag number which can range from 0h to Fh.
	dddddddd is the hexadecimal representation of the GCL internal flag value(s).
	Each character represents 4 GCL internal flags' values.
	(cr) is the terminating character, carriage return (0Dh)
Response	>aa(cr) if the command was valid.
	?aa(cr) if an invalid command has been issued.
	There is no response if the module detects a syntax error or communication error or if the address does not exist.
	> delimiter indicating a valid command was received.
	? delimiter indicating the command was invalid.
	aa (range 00-FF) represents the 2-character hexadecimal slave address of a module that is responding.
	(cr) is the terminating character, carriage return (0Dh)
Example 1	This command sets all GCL internal flags values.
	Values of All GCL internal flags (Flag 0 ~ 15) are logic low.
ASCII	Command: #01Vd000000000000(cr)
	Response: >01(cr)
Example 2	This command sets all GCL internal flags values.
	Values of All GCL internal flags (Flag 0 ~ 15) are logic high.
ASCII	Command: #01Vd00000000FFFF(cr)
	Response: >01(cr)
Example 3	This command sets one specific GCL internal flag value.
	GCL internal flag (Flag 3) is logic high.
ASCII	Command: #01Vd100300000001(cr)
	Response: >01(cr)
Example 4	This command sets one specific GCL internal flag value.
	GCL internal flag (Flag 14) is logic low.
ASCII	Command: #01Vd100E00000000(cr)
	Response: >01(cr)

$aaVd

Name	Read GCL Internal Flags' (Auxiliary Flags) Values
Description	This command reads all GCL internal flags' values from the specific ADAM-6200 module.
Syntax	$aaVd(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave network address of the ADAM-6200 module. (Always 01) Vd is the GCL Internal Flag command. (cr) is the terminating character, carriage return (0Dh)
Response	>aadddddddd(cr) if the command was valid. ?aa(cr) if an invalid command has been issued. There is no response if the module detects a syntax error or communication error or if the address does not exist. > delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of a module that is responding. ddddddddd is the hexadecimal representation of the GCL internal flag value(s). Each character represents 4 GCL internal flags' values. (cr) is the terminating character, carriage return (0Dh)
Example	This command reads all GCL internal flags values. The characters "B" mean "1011" and character "2" means "0010",so GCL internal flag 0, 1, 3, 5 are logic high while other GCL internal flags are all logic low.
ASCII	Command: $01Vd (cr) Response: >010000002B(cr)

B.1.2 Analog Input Command Set

Function	Description	Command	Example
Read analog input from Channel N	Return the input value from the specified analog input channel	Cmd: #aan(cr) Ret:>+nnnn.nnn(cr) or >–nnnn.nnn(cr)	Send: #010(cr) Receive: >+0000.000(cr)
Read analog input from all channels	Returns the input data from all analog input channels in a specified module.	Cmd: #aa(cr) Ret:>(data) (data) (data) (data) (data) (data)(cr) Where (data) is +nnnn.nnn or –nnnn.nnn	Send: #010(cr) Receive: >+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000(cr)
Auto Calibration	Calibrate the analog input range of module	Cmd: $aa1(cr) Ret:!aa(cr)	Send: $011(cr) Receive: !01(cr)
Read channel Enable/Disable Status	Asks a specified module to return the Enable/Disable status of all analog input channels	Cmd: $aa6(cr) Ret:!aamm(cr)	Send: $016(cr) Receive: !01FF(cr)
Set channel Enable/Disable Status	Set Enable/Disable status for analog input channels	Cmd: $aa5mm(cr) Ret:!aa(cr)	Send: $015FF(cr) Receive: !01(cr)
Read all Max. Data	Read the maximum data from all analog input channels	Cmd: #aaMH(cr) Ret:> (data) (data) (data) (data) (data) (data)(cr) Where (data) is +nnnn.nnn or –nnnn.nnn	Send: #01MH(cr) Receive: >+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000(cr)
Read single Max. Data	Read the maximum date from a specified analog input channel	Cmd: #aaMHn(cr) Ret:> (data) (cr) Where (data) is +nnnn.nnn or –nnnn.nnn	Send: #01MH1(cr) Receive: >+0010.000(cr)
Read all Min. Data	Read the minimum data from all analog input channels	Cmd: #aaML(cr) Ret:> (data) (data) (data) (data) (data) (data)(cr) Where (data) is +nnnn.nnn or –nnnn.nnn	Send: #01ML(cr) Receive: >+0000.000+0000.000+0000.000+0000.000+0000.000+0000.000+0000.000+0000.000(cr)
Read single Min. Data	Read the minimum data from a specified analog input channel	Cmd: #aaMLn(cr) Ret:> (data) (cr)	Send: #01ML3(cr) Receive: >+0000.000(cr)
Read Analog input range code	Return the input range code from the specific analog input channel	Cmd: $aaBnn(cr)(cr) Ret:> aa(Code) (cr)	Send: $01B01(cr) Receive: >010182(cr)

#aan

Name	Read Analog Input from Channel N
Description	Returns the input data from a specified analog input channel in a specified module.
Syntax	#aan(cr) # is a delimiter character. aa (range 00-FF) means the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) n (range 0-8) represents the specific channel you want to read the input data. (cr) is the terminating character, carriage return (0Dh).
Response	>(data)(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. > delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. (cr) is the terminating character, carriage return (0Dh).
Example	Channel 2 of the ADAM-6200 analog module at address 01h responds with an input value +10.000.
ASCII	Command: #012(cr) Response: >+0010.000(cr)

#aa

Name	Read Analog Input from All Channels
Description	Returns the input data from all analog input channels in a specified module.
Syntax	#aa(cr) # is a delimiter character. aa (range 00-FF) menans the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) (cr) is the terminating character, carriage return (0Dh).
Response	>(data)(data)(data)(data)(data)(data)(data)(data)(data)(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. > delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. (cr) is the terminating character, carriage return (0Dh).
Example	Command: #01(cr) Response:>+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000(cr)

Note! *The latest data returned is the Average value of the preset channels in this module.*

$aa1

Name	Auto Calibration
Description	Calibrate the analog input range of module
Syntax	$aa1(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal Modbus address of the ADAM-6200 module which is to be calibrated. (Always 01) 1 represents the zero calibration command. (cr) is the terminating character, carriage return (0Dh)
Response	!aa(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (cr) is the terminating character, carriage return (0Dh)

$aa6

Name	Read Channel Enable/Disable Status
Description	Asks a specified module to return the Enable/Disable status of all analog input channels
Syntax	$aa6(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) 6 is the read channels status command. (cr) is the terminating character, carriage return (0Dh)
Response	!aamm(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. mm are two hexadecimal values. Each value is interpreted as 4 bits. The first 4-bit value represents the status of channels 7-4, the second 4 bits represents the status of channels 3-0. A value of 0 means the channel is disabled, while a value of 1 means the channel is enabled. (cr) is the terminating character, carriage return (0Dh)
Example	The command asks the specific module at address 01h to send Enable/Disable status of all analog input channels. The analog input module responds that all its channels are enabled (FF equals 1111 and 1111).
ASCII	Command: $016(cr) Response: !01FF(cr)

$aa5mm

Name	Set Channel Enable/Disable Status
Description	Set Enable/Disable status for all analog input channels
Syntax	$aa5mm(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module. (Always 01) 5 identifies the enable/disable channels command. mm (range 00-FF) are two hexadecimal characters. Each character is interpreted as 4 bits. The first 4-bit value represents the status of channels 7-4. The second 4-bit value represents the status of channels 3-0. A value of 0 means the channel is disabled, while a value of 1 means the channel is enabled. (cr) is the terminating character, carriage return (0Dh)
Response	!aa(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (cr) is the terminating character, carriage return (0Dh)
Example	The command enables/disables channels of the analog input module at address 01h. Hexadecimal 8 equals binary 1000, which enables channel 7 and disables channels 4, 5 and 6. Hexadecimal 1 equals binary 0001, which enables channel 0 and disables channels 1, 2 and 3.
ASCII	Command: $01581(cr) Response: !01(cr)

#aaMH

Name	Read Maximum Value
Description	Read the maximum values from all analog input channels in a specified analog module
Syntax	#aaMH(cr) # is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module to be read.(Always 01) MH represents the read maximum value command. (cr) is the terminating character, carriage return (0Dh)
Response	>(data)(data)(data)(data)(data)(data)(data)(data)(data)(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. > delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (cr) is the terminating character, carriage return (0Dh)
Example	The command asks the specific module at address 01h to send historic maximum value from analog input channels.
ASCII	Command: #01MH(cr) Response: >+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000+0010.000 +0010.000(cr)

Note! *The latest data returned is the Average value of the preset channels in this module.*

#aaMHn

Name	Read Maximum Value from channel N
Description	Read the maximum value from a specific channel in a specified module
Syntax	#aaMHn(cr) # is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module to be read. (Always 01) MH represents the read maximum value command. n (range 0-8) represents the specific channel you want to read the input data. (cr) is the terminating character, carriage return (0Dh)
Response	>(data)(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. > delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (cr) is the terminating character, carriage return (0Dh)
Example	The command asks the specific module at address 01h to send historic maximum value from analog input channel 2.
ASCII	Command: #01MH2(cr) Response: >+0010.000(cr)

#aaML

Name	Read Minimum Value
Description	Read the minimum values from all analog input channels in a specified module
Syntax	#aaML(cr) # is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module to be read.(Always 01) ML represents the read minimum value command. (cr) is the terminating character, carriage return (0Dh)
Response	>(data)(data)(data)(data)(data)(data)(data)(data)(data)(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. > delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (cr) is the terminating character, carriage return (0Dh)
Example	The command asks the specific module at address 01h to send historic minimum value from all AI channels.
ASCII	Command: #01ML(cr) Response: >+0000.000+0000.000+0000.000+0000.000+0000.000+0000.000+0000.000+0000.000(cr)

Note! *The latest data returned is the Average value of the preset channels in this module.*

#aaMLn

Name	Read Minimum Value from channel N
Description	Read the minimum value from a specific analog input channel in a specified module
Syntax	#aaMLn(cr) # is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module to be read. (Always 01) ML stands for the read minimum value command. n (range 0-8) means the specific channel you want to read the input data. (cr) is the terminating character, carriage return (0Dh)
Response	>(data)(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. > delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (cr) is the terminating character, carriage return (0Dh)
Example	The command asks the specific module at address 01h to send historic minimum value from analog input channel 3.
ASCII	Command: #01ML3(cr) Response: >+000.000(cr)

$aaBnn

Name	Read Analog Input Range Code from Channel N
Description	Returns the range code from a specified analog input channel in a specified module.
Syntax	$aaBnn(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) B is the Analog Input Range Code command. nn (range 00-07) represents the specific channel you want to read the range code. (cr) is the terminating character, carriage return (0Dh).
Response	!aa(code) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. (cr) is the terminating character, carriage return (0Dh). (code) is the range code read. Refer to the tables below to see the meaning of range code for different modules.

B.1.3 Analog Input Range code

Range Code	HEX
+/- 10V	0x0143
+/- 5 V	0x0142
+/- 1 V	0x0140
+/- 500 mV	0x0104
+/- 150 mV	0x0103
+/- 20 mA	0x0181
0 ~ 20 mA	0x0182
4 ~ 20 mA	0x0180

Example The command asks the specific module at address 01h to send analog input range code value from analog input channel 3.

ASCII Command: #01B03(cr)

Response: >010182(cr)

B.1.4 Analog Input Alarm Command Set

Command Syntax	Command Name	Description
$aaCjAhs	Set Alarm Mode	Set the High/Low alarm in either Momentary or Latch-ing mode
$aaCjAh	Read Alarm Mode	Returns the alarm mode for the specified chan-nels
$aaCjAhEs	Enable/Dis-able Alarm	Enables/Disables the high/low alarm of the specified channels
$aaCjCh	Clear Latch Alarm	Resets a latched alarm
$aaCjAhCCn	Set Alarm Connection	Connects the High/Low alarm of a specified input channel to interlock with a specified out-put channel
$aaCjRhC	Read Alarm Connection	Returns the alarm configuration of a specified input channel
$aaCjAhU	Set Alarm Limit	Sets the High/Low alarm limit value to a speci-fied channel
$aaCjRhU	Read Alarm Limit	Returns the High/Low alarm limit value of the speci-fied channel
$aaCjS	Read Alarm Status	Reads whether an alarm occurred in the speci-fied channel

$aaCjAhs

Name	Set Alarm Mode
Description	Sets the High/Low alarm of the specified input channel in the addressed ADAM-6200 module to either Latching or Momentary mode.
Syntax	$aaCjAhs(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave network address of an ADAM-6200 module. (Always 01) Cj identifies the desired channel j (j : 0 to 7). A is the Set Alarm Mode command. h indicates alarm types (H = High alarm, L = Low alarm) s indicates alarm modes (M = Momentary mode, L = Latching mode) (cr) represents terminating character, carriage return (0Dh)
Response	!aa(cr) if the command was valid ?aa(cr) if an invalid operation was entered. There is no response if the system detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. aa represents the 2-character hexadecimal slave address of the corresponding ADAM-6200 module. (cr) represents terminating character, carriage return (0Dh)
Example	Channel 1 of the ADAM-6200 module at address 01h is instructed to set its High alarm in Latching mode. The module confirms that the command has been received.
ASCII	Command: $01C1AHL(cr) Response: !01(cr)

$aaCjAh

Name	Read Alarm Mode
Description	Returns the alarm mode for the specified channel in the specified ADAM-6200 module.
Syntax	$aaCjAh(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (Always 01) Cj identifies the desired channel j (j : 0 to 7). A is the Read Alarm Mode command. h indicates the alarm types (H = High alarm, L = Low alarm) (cr) represents terminating character, carriage return (0Dh)
Response	!aas(cr) if the command was valid ?aa(cr) if an invalid operation was entered. There is no response if the system detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. aa presents the 2-character hexadecimal slave address of the corresponding ADAM-6200 module. s indicates alarm modes (M = Momentary mode, L = Latching mode) (cr) represents terminating character, carriage return (0Dh)
Example	Channel 1 of the ADAM-6200 module at address 01h is instructed to return its Low alarm mode. The system responds that it is in Momentary mode.
ASCII	Command: $01C1AL(cr) Response: !01M(cr)

$aaCjAhEs

Name	Enable/Disable Alarm
Description	Enables/Disables the High/Low alarm of the specified input channel in the addressed ADAM-6200 module
Syntax	$aaCjAhEs(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (Always 01) Cj identifies the desired channel j (j : 0 to 7). AhEs is the Set Alarm Mode command. h indicates alarm type (H = High alarm, L = Low alarm) s indicates alarm enable/disable (E = Enable, D = Disable) (cr) represents terminating character, carriage return (0Dh)
Response	!aa(cr) if the command was valid ?aa(cr) if an invalid operation was entered. There is no response if the system detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. aa represents the 2-character hexadecimal slave address of the corresponding ADAM-6200 module. (cr) represents terminating character, carriage return (0Dh)
Example	Channel 1 of the ADAM-6200 module at address 01h is instructed to enable its Low alarm function. The module confirms that its Low alarm function has been enabled.
ASCII	Command: $01C1ALEE(cr) Response: !01(cr)

Note! *An analog input module requires a maximum of 2 seconds after it receives an Enable/Disable Alarm command to let the setting take effect.*

$aaCjCh

Name	Clear Latch Alarm
Description	Sets the High/Low alarm to OFF (no alarm) for the specified input channel in the addressed ADAM-6200 module
Syntax	$aaCjCh(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave network address of an ADAM-6200 module. (Always 01) Cj identifies the desired channel j (j : 0 to 7). Ch is the Clear Latch Alarm command. h indicates alarm type (H = High alarm, L = Low alarm) (cr) represents terminating character, carriage return (0Dh)
Response	!aa(cr) if the command was valid ?aa(cr) if an invalid operation was entered. There is no response if the system detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. aa represents the 2-character hexadecimal slave address of the corresponding ADAM-6200 module. (cr) represents terminating character, carriage return (0Dh)
Example	Command: $01C1CL(cr) Response: !01(cr) Channel 1 of the ADAM-6200 module at address 01h is instructed to set its Low alarm state to OFF. The system confirms it has done so accordingly.

$aaCjAhCCn

Name	Set Alarm Connection
Description	Connects the High/Low alarm of the specified input channel to interlock the specified digital output in the addressed ADAM-6200 module
Syntax	$aaCjAhCCn(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module.(Always 01) Cj identifies the desired analog input channel j (j : 0 to 7). AhC is the Set Alarm Connection command. h indicates alarm type (H = High alarm, L = Low alarm) Cn identifies the desired digital output channel n (n : 0 to 1). To disconnect the digital output, n should be set as ?*?. (cr) represents terminating character, carriage return (0Dh)
Response	!aa(cr) if the command was valid ?aa(cr) if an invalid operation was entered. There is no response if the system detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. aa represents the 2-character hexadecimal slave address of the corresponding ADAM-6200 module. (cr) represents terminating character, carriage return (0Dh)
Example	Channel 1 of the ADAM-6200 module at address 01h is instructed to connect its Low alarm to the digital output of channel 0 in the specific module. The system confirms it has done so accordingly.
ASCII	Command: $01C1ALCC0(cr) Response: !01(cr)

$aaCjRhC

Name	Read Alarm Connection
Description	Returns the High/Low alarm limit output connection of a specified input channel in the addressed module
Syntax	$aaCjRhC(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (Always 01) Cj identifies the desired analog input channel j (j : 0 to 7). RhC is the Read Alarm Connection command. h indicates alarm type (H = High alarm, L = Low alarm) (cr) represents terminating character, carriage return (0Dh)
Response	!aaCn(cr) if the command was valid ?aa(cr) if an invalid operation was entered. There is no response if the system detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. aa represents the 2-character hexadecimal slave address of the corresponding ADAM-6200 module. Cn identifies the desired digital output channel n (n : 0 to 1) whether interlock with the alarm of the specific analog input channel. If the values of n are "*", the analog input has no connection with a digital output point. (cr) represents terminating character, carriage return (0Dh)
Example	Channel 1 of the ADAM-6200 module at address 01h is instructed to read its Low alarm output connection. The system responds that the Low alarm output connects to the digital output at channel 0 in the specific module.
ASCII	Command: $01C1RLC(cr) Response:!01C0(cr)

$aaCjAhU

Name	Set Alarm Limit
Description	Sets the High/Low alarm limit value for the specified input channel of a specified ADAM-6200 module.
Syntax	$aaCjAhU(data)(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module.(Always 01) Cj identifies the desired analog input channel j (j : 0 to 7). AhU is the Set Alarm Limit command. h indicates alarm type (H = High alarm, L = Low alarm) (data) represents the desired alarm limit setting. The format is always in engineering units. (cr) represents terminating character, carriage return (0Dh)
Response	!aa(cr) if the command was valid ?aa(cr) if an invalid operation was entered. There is no response if the system detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. aa represents the 2-character hexadecimal slave address of the corresponding ADAM-6200 module. (cr) represents terminating character, carriage return (0Dh)
Example	The high alarm limit of the channel 1 in the specific module at address 01h is been set +80. The system confirms the command has been received.
ASCII	Command: $01C1AHU+080.00(cr) Response:!01(cr)

Note! *An analog input module requires a maximum of 2 seconds after it receives a Set Alarm Limit command to let the settings take effect.*

$aaCjRhU

Name	Read Alarm Limit
Description	Returns the High/Low alarm limit value for the specified input channel in the addressed ADAM-6200 module
Syntax	$aaCjRhU(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module.(Always 01) Cj identifies the desired analog input channel j (j : 0 to 7). RhU is the Read Alarm Limit command. h indicates alarm type (H = High alarm, L = Low alarm) (cr) represents terminating character, carriage return (0Dh)
Response	!aa(data)(cr) if the command was valid ?aa(cr) if an invalid operation was entered. There is no response if the system detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. aa represents the 2-character hexadecimal slave address of the corresponding ADAM-6200 module. (data) represents the desired alarm limit setting. The format is always in engineering units. (cr) represents terminating character, carriage return (0Dh)
Example	Channel 1 of the ADAM-6200 module at address 01h is configured to accept 5V input. The command instructs the system to return the High alarm limit value for that channel. The system responds that the High alarm limit value in the desired channel is 2.0500 V.
ASCII	Command: $01C1RHU(cr) Response: !01+2.0500(cr)

$aaCjS

Name	Read Alarm Status
Description	Reads whether an alarm occurred to the specified input channel in the specified ADAM-6200 module
Syntax	$aaCjS(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of an ADAM-6200 module. (Always 01) Cj identifies the desired analog input channel j (j : 0 to 7). S is the Read Alarm Status command. (cr) represents terminating character, carriage return (0Dh)
Response	!aahl(cr) if the command was valid ?aa(cr) if an invalid operation was entered. There is no response if the system detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. aa represents the 2-character hexadecimal slave address of the corresponding ADAM-6200 module. h represents the status of High alarm. "1" means the High alarm occurred, '0" means it did not occur. l represents the status of Low alarm. ?1" means the Low alarm occurred,?0" means it did not occur. (cr) represents terminating character, carriage return (0Dh)
Example	The command asks the module at address 01h to return its alarm status for channel 1. The system responds that a High alarm has not occurred, but the Low alarm has occurred.
ASCII	Command: $01C1S(cr) Response: !0101(cr)

B.1.5 Analog Output Command Set

Function	Description	Command	Example
Read analog output startup value	Read startup output value from a specific analog output channel	Cmd: $aaDScc(cr) Ret:!aannnn (cr) nnnn range: 0x0000 ~ 0x0FFF	Send: $01DS01(cr) Receive: !010800(cr)
Set analog output startup value	Set startup output value to a specific analog output channel	Cmd: #aaDSccnnnn(cr) Ret: !aa(cr)	Send: #01DS010800(cr) Receive: !01(cr)
Write AO values	Write value to the specific analog output channel	Cmd: #aaBCccnnnn(cr) Ret: !aa(cr)	Send: #01BC010800(cr) Receive: !01(cr)
Read AO range code	Return the channel range code form specific analog output channel	Cmd: $aaBEcc(cr) Ret: !aannnn(cr)	Send: $01BE00(cr) Receive: !010147(cr)

$aaDScc

Name	Read AO Startup Value from One Channel
Description	Returns the startup value from a specified analog output channel in a specified module.
Syntax	$aaDScc(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) DS represents the analog output channel startup command. cc (range 00-01) represents the specific channel you want to read the startup value. (cr) is the terminating character, carriage return (0Dh).
Response	!aannnn(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) nnnn (range: 0000-0FFF) represents the 4-character hexadecimal startup value of the specific AO channel. (cr) is the terminating character, carriage return (0Dh).
Example	Startup value for analog output channel 1 of the ADAM-6224 module at address 01h responds with a value +10.000. (The AO range of channel 1 is 0~10V)
ASCII	Command: $01DS01(cr)

Response: !010FFF(cr)

#aaDSccnnnn

Name	Set AO Startup Value to One Channel
Description	Set the startup value to a specified analog output channel in a specified module.
Syntax	$aaDSccnnnn(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) DS represents the analog output channel startup command. cc (range 00-01) represents the specific channel you want to set the startup value. nnnn (range 0000-0FFF) represents the 4-character hexadecimal startup value of the specific AO channel. (cr) is the terminating character, carriage return (0Dh).
Response	!aa(cr) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) (cr) is the terminating character, carriage return (0Dh).
Example	Startup value for analog output channel 1 of the ADAM-6224 module at address 01h is set with a value +10.000. (The AO range of channel 1 is 0~10V)
ASCII	Command: $01DS01FFF(cr) Response: !01(cr)

#aaBCccnnnn

Name	Write AO Value to One Channel
Description	Write output value to a specified analog output channel in a specified module.
Syntax	#aaBCccnnnn(cr)
	# is a delimiter character.
	aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01)
	cc (range 00-01) represents the specific channel you want to write output value.
	nnnn (range:0000 - 0FFF) represents the analog output value of the specific analog output channel.
	(cr) is the terminating character, carriage return (0Dh).
Response	!aa(cr) if the command is valid.
	?aa(cr) if an invalid operation was entered.
	There is no response if the module detects a syntax error or communication error or if the address does not exist.
	! delimiter indicating a valid command was received.
	? delimiter indicating the command was invalid.
	aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01)
	BC is Write AO Value to One Channel command
	(cr) is the terminating character, carriage return (0Dh).
Example	Value for analog output channel 1 of the ADAM-6224 module at address 01h is set with a value 0x800.
ASCII	Command: #01DS010800 (cr)
	Response: !01(cr)

$aaBEcc

Name	Read Analog Output Range Code from Channel N
Description	Returns the range code from a specified analog output channel in a specified module.
Syntax	$aaBEcc(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) cc (range 00-03) represents the specific channel you want to read the range code. (cr) is the terminating character, carriage return (0Dh).
Response	!aa(code) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. ! delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) nnnn stands for Analog output range code (cr) is the terminating character, carriage return (0Dh).

B.1.6 Analog Output Range code

Range Code	HEX
+/- 10V	0x0143
+/- 5 V	0x0142
0 ~ 10V	0x0148
0 ~ 5V	0x0147
0 ~ 20 mA:	0x0182
4 ~ 20 mA:	0x0180

Example The command asks the specific module at address 01h to send analog output range code value from analog input channel 3.

ASCII Command: #01BE03(cr)

Response: !010147(cr)

B.1.7 Digital Input/Output Command Set

Function	Description	Command	Example
Read Channels	Asks a specified input module to return the status of all channels	Cmd: $aa6(cr) Ret:!aannnnnn (cr)	Send: $016(cr) Receive: !01000124(cr)
Write all DO channels	Writes specified values to all channels simultaneously	Cmd: #aa00nnnn(cr) Ret:>aa(cr)	Send: #0100007F(cr) Receive: >01(cr)
Write a single DO channel	Writes specified values to a single channel	Cmd: #aa1c01(cr) or #aa1c00(cr) Ret:>aa(cr)	Set channel 2 to logic 1 Send: #011201(cr) Receive: >01(cr)
Read DI Channel Counter Value	Returns the counter value from specified DI channels in a specified module	Cmd: $aaJCFFFFssmm(cr) Ret:>aa(data)… (cr)	Send: $01JCFFFF0102(cr) Receive: >0100000002000000 6(cr)

$aa6

Name Read Channel Status

Description This command requests that the specified ADAM-6200 module return the status of its digital input channels

Syntax $aa6(cr)

$ is a delimiter character.

aa (range 00-FF) represents the 2-character hexadecimal slave network address of the ADAM-6200 module. (Always 01)

6 is the Digital Data In command.

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

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

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

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

! delimiter indicating a valid command was received.

? delimiter indicating the command was invalid.

aa (range 00-FF) represents the 2-character hexadecimal slave network address of an ADAM-6200 module.

(data) a 2-character hexadecimal value representing the values of the digital I/O module.

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

Example Command: $016(cr)

Response: !01000124(cr)

Response	!01	0	0	0	1	2	4
Transfer to 3 Bytes		0x00		0x01		0x24	
Data can be presented as 24 Bit		0x000124 (Bit 0 ~ 23, Bit 2, 5, and 8 is 1, but others is 0)					
ADAM-6250		Bit 0~7: DI[0] ~ DI[7] Bit 8~14: DO[0] ~ DO[6]					
ADAM-6251		Bit 0~15: DI[0] ~ DI[15]					
ADAM-6256		Bit 0~15: DO[0] ~ DO[15]					
ADAM-6260		Bit 0~5: DO[0] ~ DO[5]					
ADAM-6266		Bit 0~3: DI[0] ~ DI[3] Bit 4~7: DO[0] ~ DO[3]					

#aa00(data)

Name	Write all digital output channels
Description	This command sets all digital output channels to the specific ADAM- 6200 module.
Syntax	#aa00(data)(cr)
	# is a delimiter character.
	aa (range 00-FF) represents the 2-character hexadecimal slave network address of the ADAM-6200 module. (Always 01)
	00 means to set all channel(s).
	(data) is the hexadecimal representation of the DO value(s).
Response	>(cr) if the command was valid.
	?aa(cr) if an invalid command has been issued.
	There is no response if the module detects a syntax error or communication error or if the address does not exist.
	> delimiter indicating a valid command was received.
	? delimiter indicating the command was invalid.
	aa (range 00-FF) represents the 2-character hexadecimal slave network address of a module that is responding.
	(cr) is the terminating character, carriage return (0Dh)
Example	All Digital Output channels of ADAM-6250 will be logic 1.
ASCII	Command: #0100007F(cr)
	Response: >01<(cr)
Example	All Digital Output channels of ADAM-6256 will be logic 1.
ASCII	Command: #0100FFFF(cr)
	Response: >01<(cr)

#aa1c(data)

Name	Write a single digital output channel
Description	This command sets a single DO channels to the specific ADAM-6200 module.
Syntax	#aa1c(data)(cr) # is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave network address of the ADAM-6200 module. (Always 01) c is used to indicate which channel(s) you want to set. (data) is the hexadecimal representation of the DO value(s). When writing to a single channel (bit), the first character is always 0. The value of the second character is either 0 or 1.
Response	>(cr) if the command was valid. ?aa(cr) if an invalid command has been issued. There is no response if the module detects a syntax error or communication error or if the address does not exist. > delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. aa (range 00-FF) represents the 2-character hexadecimal slave network address of a module that is responding. (cr) is the terminating character, carriage return (0Dh)
Example	Set channel 2 to logic 1
ASCII	Command: #011201(cr) Response: >01(cr)

$aaJCFFFFssmm

Name	Read DI Channel Counter Value
Description	Returns the counter value from specified DI channels in a specified module.
Syntax	$aaJCFFFFssmm(cr) $ is a delimiter character. aa (range 00-FF) represents the 2-character hexadecimal slave address of the ADAM-6200 module you want to request. (Always 01) JCFFFF is the Digital Input Channel Counter Value command. ss (range 00-07) means the specific start channel you want to read the counter value. mm (range 00-07) means the total channel numbers you want to read the counter value. (cr) is the terminating character, carriage return (0Dh).
Response	>aa(data) if the command is valid. ?aa(cr) if an invalid operation was entered. There is no response if the module detects a syntax error or communication error or if the address does not exist. > delimiter indicating a valid command was received. ? delimiter indicating the command was invalid. (data) is the counter value read. (cr) is the terminating character, carriage return (0Dh).
Example	The command requests the module at address 01h to return count value from channel 0. (the first read channel is 0, represent by "00", and only one channel is read) That module return the count value 0000000A(h) from channel 0.
ASCII	Command: $01JCFFFF0001(cr) Response: >010000000A(cr)
Example	The command requests the module at address 01h to return count value from channel 12 and 13. (The first read channel is 12, represent by "0C", and two channels are read) That module return the count value 0000000A(h) from channel 12 and 00000001(h) from channel 13.
ASCII	Command: $01JCFFFF0C02(cr) Response: >010000000A00000001(cr)

Appendix C

REST for ADAM-6200

C.1 Introduction

REpresentational State Transfer (REST) is a design style of software architecture for Web application behaves and services including image indication, resource request and response and message delivery. It can be developed compatible with popular protocols or standards like HTTP, URI, XML, HTML. With the advantage of scalability, simplicity and performance, it's already adopted in Web service by Amazon, Yahoo. The Web service of ADAM-6200 is developed based on HTML5 language, if user need to integrate this into other Web services, the following information/command list should be referred for implementation.

C.2 REST Resources for ADAM

C.2.1 Analog Input

C.2.1.1 GET /analoginput/(all|{id})/value

Request	The content-type will be 'application/x-www-form-urlencoded'. {id} : is the AI channel ID starting from 0 **Examples:** Use the following URI to get the AI-0 value. http://10.0.0.1/analoginput/0/value Use the following URI to get the all AI values. http://10.0.0.1/analoginput/all/value
Response	The content-type will be 'text/xml' If result is OK, the content will look like below <?xml version="1.0" ?> <ADAM-6217 status="OK"> <AI> <ID>0</ID> < VALUE>FFFF</VALUE> </AI> </ADAM-6217> If result is failed , the content will look like below <?xml version="1.0" ?> <ADAM-6217 status="{error}"> </ADAM-6217> {error} : The error message.
Remarks	If the {id} is out of range, the response will return HTTP status code 501 (Not implemented)

C.2.1.2 GET /analoginput/(all|{id})/range

Request	The content-type will be 'application/x-www-form-urlencoded'. {id} : is the AI channel ID starting from 0 **Examples:** Use the following URI to get the AI-0 range information. http://10.0.0.1/analoginput/0/range Use the following URI to get the all AI range information. http://10.0.0.1/analoginput/all/range
Response	The content-type will be 'text/xml' If result is OK, the content will look like below <?xml version="1.0" ?> <ADAM-6217 status="OK"> <AI> <ID>0</ID> <RANGE>7</RANGE> <NAME>4~20 mA</NAME> <MAX>20</MAX> <MIN>4</MIN> <UNIT>mA</UNIT> </AI> </ADAM-6217> If result is failed , the content will look like below <?xml version="1.0" ?> <ADAM-6217 status="{error}"> </ADAM-6217> {error} : The error message.
Remarks	If the {id} is out of range, the response will return HTTP status code 501 (Not implemented)

C.2.2 Analog Output

C.2.2.1 GET /analogoutput/(all|{id})/value

Request	The content-type will be 'application/x-www-form-urlencoded'. {id} : is the AO channel ID starting from 0 **Examples:** Use the following URI to get the AO-0 value. http://10.0.0.1/analogoutput/0/value Use the following URI to get the all AO values. http://10.0.0.1/analogoutput/all/value
Response	The content-type will be 'text/xml' If result is OK, the content will look like below <?xml version="1.0" ?> <ADAM-6224 status="OK"> <AO> <ID>0</ID> <VALUE>0FFF</VALUE> </AO> </ADAM-6224> If result is failed , the content will look like below <?xml version="1.0" ?> <ADAM-6224 status="{error}"> </ADAM-6224> {error} : The error message.
Remarks	If the {id} is out of range, the response will return HTTP status code 501 (Not implemented)

C.2.2.2 POST /analogoutput/all/value

Request	The content-type will be 'application/x-www-form-urlencoded'. **Examples:** Use the following URI to set the AO value(s). http://10.0.0.1/analogoutput/all/value The coming data with the request will be {name}={value} pair(s). {name} : The name of the channel, for example AO0. {value} : The value to be set to the indicated channel. For example, if the request is going to set channel 0, 1, 2 at +- 5V output range to 0V, -2V and 5V respectively, then the name-value pairs will look like below: AO0=0800&AO1=04CC&AO2=0FFF
Response	The content-type will be 'text/xml' The content will look like below <?xml version="1.0" ?> <ADAM-6224 status="{status}"> </ADAM-6224> {status} : The result. If succeed, the result will be 'OK'; otherwise, the result will be the error message.
Remarks	

C.2.2.3 GET /analogoutput/(all|{id})/range

Request	The content-type will be 'application/x-www-form-urlencoded'. {id} : is the AO channel ID starting from 0 **Examples:** Use the following URI to get the AO-0 range information. http://10.0.0.1/analogoutput/0/range Use the following URI to get the all AO range information. http://10.0.0.1/analogoutput/all/range
Response	The content-type will be 'text/xml' If result is OK, the content will look like below <?xml version="1.0" ?> <ADAM-6224 status="OK"> <AO> <ID>0</ID> <RANGE>7</RANGE> <NAME>4~20 mA</NAME> <MAX>20</MAX> <MIN>4</MIN> <UNIT>mA</UNIT> </AO> </ADAM-6224> If result is failed , the content will look like below <?xml version="1.0" ?> <ADAM-6224 status="{error}"> </ADAM-6224> {error} : The error message.
Remarks	If the {id} is out of range, the response will return HTTP status code 501 (Not implemented)

C.2.2.4 GET /analogoutput/(all|{id})/range

Request	The content-type will be 'application/x-www-form-urlencoded'. {id} : is the AO channel ID starting from 0 **Examples:** Use the following URI to get the AO-0 range information. http://10.0.0.1/analogoutput/0/range Use the following URI to get the all AO range information. http://10.0.0.1/analogoutput/all/range
Response	The content-type will be 'text/xml' If result is OK, the content will look like below <?xml version="1.0" ?> <ADAM-6224 status="OK"> <AO> <ID>0</ID> <RANGE>7</RANGE> <NAME>4~20 mA</NAME> <MAX>20</MAX> <MIN>4</MIN> <UNIT>mA</UNIT> </AO> </ADAM-6224> If result is failed , the content will look like below <?xml version="1.0" ?> <ADAM-6224 status="{error}"> </ADAM-6224> {error} : The error message.
Remarks	If the {id} is out of range, the response will return HTTP status code 501 (Not implemented)

C.2.3 Digital Input

C.2.3.1 GET /digitalinput/(all|{id})/value

Request	The content-type will be 'application/x-www-form-urlencoded'. {id} : is the DI channel ID starting from 0 **Examples:** Use the following URI to get the DI-0 value. http://10.0.0.1/digitalinput/0/value Use the following URI to get the all DI values. http://10.0.0.1/digitalinput/all/value
Response	The content-type will be 'text/xml' If result is OK, the content will look like below <?xml version="1.0" ?> <ADAM-6250 status="OK"> <DI> <ID>0</ID> <VALUE>0</VALUE> </DI> </ADAM-6250> If result is failed , the content will look like below <?xml version="1.0" ?> <ADAM-6250 status="{error}"> </ADAM-6250> {error} : The error message.
Remarks	If the {id} is out of range, the response will return HTTP status code 501 (Not implemented)

C.2.4 Digital Output

C.2.4.1 GET /digitaloutput/(all|{id})/value

Request	The content-type will be 'application/x-www-form-urlencoded'. {id} : is the DO channel ID starting from 0 **Examples:** Use the following URI to get the DO-0 value. http://10.0.0.1/digitaloutput/0/value Use the following URI to get the all DO values. http://10.0.0.1/digitaloutput/all/value
Response	The content-type will be 'text/xml' If result is OK, the content will look like below <?xml version="1.0" ?> <ADAM-6250 status="OK"> <DO> <ID>0</ID> <VALUE>1</VALUE> </DO> </ADAM-6250> If result is failed , the content will look like below <?xml version="1.0" ?> <ADAM-6250 status="{error}"> </ADAM-6250> {error} : The error message.
Remarks	If the {id} is out of range, the response will return HTTP status code 501 (Not implemented)

C.2.4.2 POST /digitaloutput/all/value

Request	The content-type will be 'application/x-www-form-urlencoded'. **Examples:** Use the following URI to set the DO value(s). http://10.0.0.1/digitaloutput/all/value The coming data with the request will be {name}={value} pair(s). {name} : The name of the channel, for example DO0. {value} : The value to be set to the indicated channel. For example, if the request is going to set channel 0, 1, 2 to value 1, then the name-value pairs will look like below: DO0=1&DO1=1&DO2=1
Response	The content-type will be 'text/xml' The content will look like below <?xml version="1.0" ?> <ADAM-6250 status="{status}"> </ADAM-6250> {status} : The result. If succeed, the result will be 'OK'; otherwise, the result will be the error message.
Remarks	

ADVANTECH

Enabling an Intelligent Planet

www.advantech.com

Please verify specifications before quoting. This guide is intended for reference purposes only.

All product specifications are subject to change without notice.

No part of this publication may be reproduced in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission of the publisher.

All brand and product names are trademarks or registered trademarks of their respective companies.

© Advantech Co., Ltd. 2013