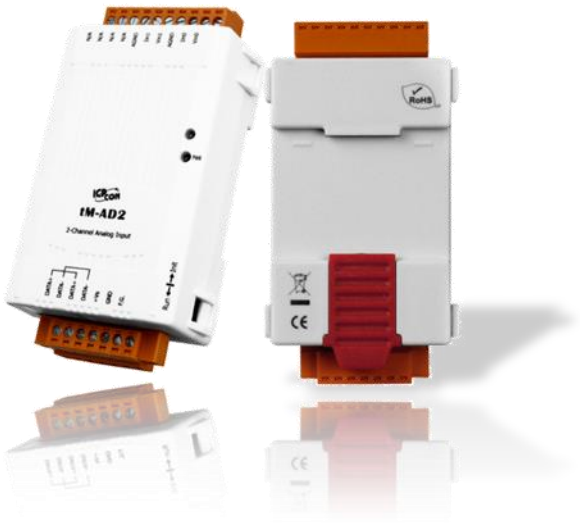
tM Series User Manual

V1.1, 2023/07



Technical support: service@icpdas.com
Author: Wayne Chen, David Chen
Editor: Janice Hong

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, not for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Contact US

If you have any problem, please feel free to contact us.
You can count on us for quick response.
Email: service@icpdas.com

Revision History

This chapter provides revision history information to this document.

The table below shows the revision history.

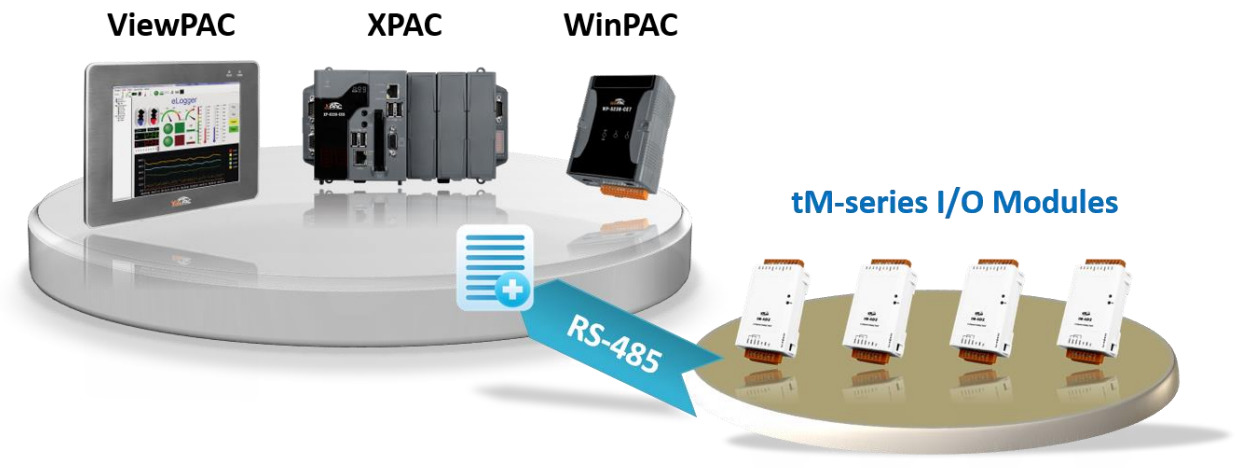
Revision	Date	Description
V1.1	Jul. 2023	Modify the download link in Section 6.4.1
V1.1	Dec. 2020	Add Appendix G – Save or Load I/O Project
V1.0	Jul. 2020	Initial issue

Content of Table

Chapter 1 Introduction

The tM series is a family of network data acquisition and control modules that are equipped with digital or analog input/output functions. The modules can be remotely controlled through an RS-485 serial bus by using DCON and Modbus RTU protocols. The selectable transmission speed of the RS-485 port is up to 115,200 bps.

Modbus has become a de facto standard communications protocol in industry and is now the most commonly available means of connecting industrial electronic devices. This makes the tM series perfect integration with the HMI, SCADA, PLC, and other software systems.

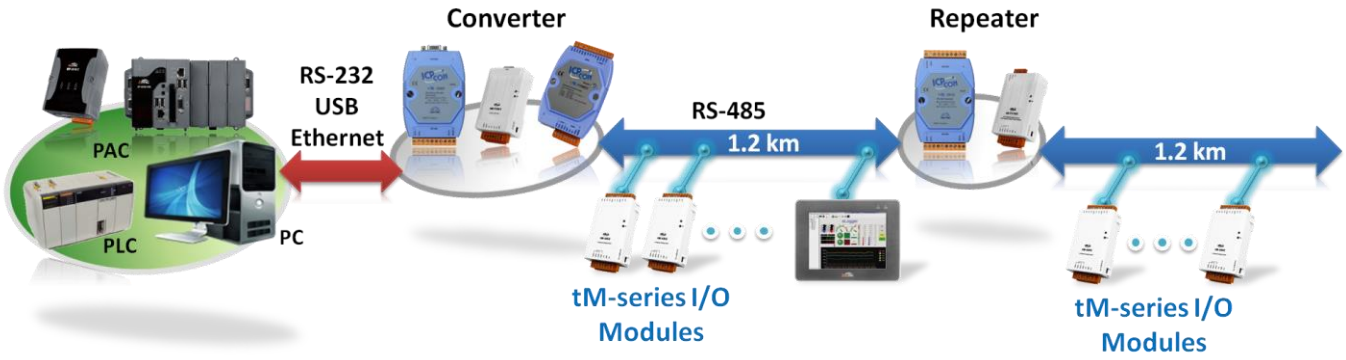


The tM series tiny RS-485 I/O modules support a variety of I/O types, like photo-isolated digital input, relay contact, photoMOS relay, and open-collector output. Compare to M-7000 series, the small number of channels designed for tM series is more cost-effective for distributed I/O applications.

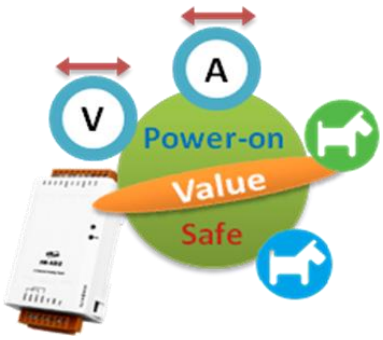
The tM series provides the module watchdog and host watchdog to enhance the reliability and stability of the control system. For maximum space savings, the tM series is offered in an amazing tiny form-factor that makes it easily installed anywhere, even directly embedded into a machine. It is equipped with two removable terminal block connectors for easy wiring.

1.1 Features

- Tiny form-factor and Low Power Consumption
- Protocol supported:
  DCON, Modbus RTU/ASCII
- Supports RS-485 Multi-drop Network
  RS-485 bus supports baud rate up to 115200 bps



- Configurable I/O Types and Ranges
- Supports Dual-Watchdog
- Configurable Power-on Value and Safe Value
- Provides DI Latched Function
- Provides Low Speed Counter
- Isolated Digital Input and Output
- RoHS Compliant with no Halogen
- Easy DIN-Rail Mounting



1.2 Selection Guide

tM Series modules				(Tabulating Date: 06, 2019)
Bus	**RS-485**			
Protocol	**Modbus RTU, Modbus ASCII, DCON**			
Models	**AI**	**AO**	**DI**	**DO**
tM-AD2	**2-ch** Single-ended, Voltage/Current	-	-	-
tM-AD5	**5-ch** Differential, Voltage			
tM-AD5C	**5-ch** Differential, Current			
tM-AD8	**8-ch** Single-ended, Voltage			
tM-AD8C	**8-ch** Single-ended, Current			
tM-TH8	**8-ch** Thermistor			

Models	**AI**	**AO**	**DI**	**DO**
tM-DA1P1R1	-	**1-channel**		
		Voltage/Current	Sink/Source	Form C Relay
tM-AD4P2C2	**4-channel**	-	**2-channel**	
	2-ch Single-ended, Voltage **2-ch** Single-ended, Current		Source	NPN, Sink
tM-P3R3	-		**3-channel**	
			Sink/Source	Form A Relay
tM-PD3R3			Source + Dry	
tM-P3POR3			Sink/Source	PhotoMos Relay
tM-P4A4	-		**4-channel**	
			Sink	PNP, Source
tM-P4C4			Source	NPN, Sink
tM-R5	-		-	**5-channel**
				Form A Relay
tM-P8	-		**8-channel**	-
			Sink/Source	
tM-PDW8			Sink/Source + Dry	
tM-C8			-	**8-channel**
				NPN, Sink

1.3 Specifications

The specifications of all tM series will be described in this section.

1.3.1 Common Specification

Communication	
Interface	RS-485
Format	(N, 8, 1), (N, 8, 2), (O, 8, 1), (E, 8, 1)
Baud Rate	1200 to 115200 bps
Protocol	DCON, Modbus/RTU, Modbus/ASCII
Dual Watchdog	Yes, Module (2.3 seconds), Communication (Programmable)
LED Indicators	
Power	1 LED as Power Indicator
EMS Protection	
ESD (IEC 61000-4-2)	±4 kV Contact for Each Terminal ±8 kV Air for Random Point
	tM-TH8: ±2 kV Contact for Each Terminal ±3 kV Air for Random Point
EFT (IEC 61000-4-4)	±4 kV for Power
Isolation	
Intra-module Isolation, Field-to-Logic	Analog Input: 2500 VDC (Multi-function I/O: tM-AD4P2C2, tM-DA1P1R1) Digital Input: 3750 VDC PhotoMOS Relay Output: 2000 VDC
Mechanical	
Dimensions (W x L x H)	52 mm x 98 mm x 27 mm
Installation	DIN-Rail Mounting
Temperature Range	
Operating	-25 °C to +75 °C
Storage	-30 °C to +75 °C
Humidity	10 to 95% RH, Non-condensing

1.3.2 Specification - AI

Module Name		tM-AD2	tM-AD5	tM-AD5C	tM-AD8	tM-AD8C
Analog Input						
Input Channels		2 single-ended	5 differential	5 differential	8 single-ended	8 single-ended
Type		0 to 500 mV 0 to 1 V 0 to 2.5 V 0 to 5 V 0 to 10 V 0 to 20 mA 4 to 20 mA	+/- 1 V +/- 2.5 V +/- 5 V +/- 10 V	0 to 20 mA 4 to 20 mA +/- 20 mA	0 to 500 mV 0 to 1 V 0 to 2.5 V 0 to 5 V 0 to 10 V	0 to 20 mA 4 to 20 mA
Resolution	Normal Mode	14-bit				
	Fast Mode	12-bit				
Sampling Rate	Normal Mode	10 Hz total				
	Fast Mode	200 Hz total				
Accuracy	Normal Mode	± 0.1%				
	Fast Mode	± 0.5%				
Zero Drift		± 20 µV/°C				
Span Drift		± 25 ppm/°C				
Input Impedance	Voltage	10 MΩ	20 MΩ	-	10 MΩ	-
	Current	125 Ω	-	125 Ω	-	125 Ω
Over Protection	Voltage	120 VDC		-	120 VDC	-
	Current	Yes, 50 mA at 110 VDC	-			
Open Wire Detection for 4 to 20 mA		Yes	-			
Power Requirements						
Reverse Polarity Protection		Yes				
Powered from Terminal Block		Yes, 10 to 30 VDC				
Consumption		0.7 W Max.	1 W Max.		0.7 W Max.	

(Refer to Section 1.3.1 - Common Specification)

Module Name	tM-TH8
Analog Input	
Input Channels	8
Type	Thermistor
Thermistor Type	Precon ST-A3, Fenwell Type U, YSI L 100, YSI L 300, YSI L 1000, YSI B 2252, YSI B 3000, YSI B 5000, YSI B 6000, YSI B 10000, YSI H 10000, YSI H 30000, User-defined
Resolution	16-bit
Sampling Rate	8 Hz total
Accuracy	± 1%
Over-voltage Protection	120 VDC
Open Wire Detection	Yes
Individual Channel Configurable	Yes
Power Requirements	
Reverse Polarity Protection	Yes
Powered from Terminal Block	Yes, 10 to 30 VDC
Consumption	1 W Max.

(Refer to Section 1.3.1 - Common Specification)

1.3.3 Specification - DI/DO

Module Name		tM-P8	tM-PDW8
Digital Input/Counter			
Input Channels		8	
Type		Wet Contact (Sink, Source)	Wet Contact (Sink, Source) Dry Contact (Source)
Wet Contact	ON Voltage Level	+3.5 VDC to 50 VDC	
	OFF Voltage Level	+1 VDC Max.	
Dry Contact	ON Voltage Level	-	Close to GND
	OFF Voltage Level	-	Open
Counters	Max. Count	65535 (16-bit)	
	Max. Input Frequency	100 Hz	
	Min. Pulse Width	5 ms	
Input Impedance		10 KΩ, 0.5 W	
Over-voltage Protection		±70 VDC	
Power Requirements			
Reverse Polarity Protection		Yes	
Powered from Terminal Block		Yes, 10 to 30 VDC	
Consumption		0.2 W Max.	0.43 W Max.

(Refer to Section 1.3.1 - Common Specification)

Module Name		tM-P3POR3	tM-P3R3	tM-PD3R3
Digital Input/Counter				
Input Channels		3		
Type		Wet Contact (Sink, Source)		Dry Contact (Source)
Wet Contact	ON Voltage Level	+3.5 VDC to +50 VDC		-
	OFF Voltage Level	+1 VDC Max.		-
Dry Contact	ON Voltage Level	-		Close to GND
	OFF Voltage Level	-		Open
Counters	Max. Count	65535 (16-bit)		
	Max. Input Frequency	100 Hz		
	Min. Pulse Width	5 ms		
Input Impedance		10 KΩ, 0.5W		-
Over-voltage Protection		70 VDC		-

Module Name		tM-P3POR3	tM-P3R3	tM-PD3R3	tM-R5
Relay Output					
Output Channels		3			5
Relay Type		PhotoMOS Relay	Power Relay, Form A (SPST N.O.)		
Operating Load Voltage Range		80 V (AC peak or DC)	250 VAC or 30 VDC		
Load Current		1 A Max.	5 A Max.		
		3 A, (Peak, 1 ms, 1 shot)			
Output Off State Leakage Current		1 uA	-		
Operating Time		5 ms (Max.)	6 ms		
Release Time		0.5 ms (Max.)	3 ms		
Electrical Life (Resistive load)	VDE	-	5 A @250 VAC 30,000 ops (10 ops/minute) at 75°C		
			5 A @30 VDC 70,000 ops (10 ops/minute) at 75°C		
	UL		5 A @250 VAC/30 VDC 6,000 ops		
			3 A @250 VAC/30 VDC 100,000 ops		
Mechanical Life		-	20,000,000 ops at no load (300 ops/minute)		
Electrical Endurance		No Arcing, No Bounce, and No Switching Noise	-		
Power-on Value		Yes, Programmable			
Safe Value		Yes, Programmable			
Power Requirements					
Reverse Polarity Protection		Yes			
Powered from Terminal Block		Yes, 10 to 30 VDC			
Consumption		0.4 W Max.	0.8 W Max.	0.9 W Max.	1 W Max.

(Refer to Section 1.3.1 - Common Specification)

Module Name		tM-P4A4	tM-P4C4	tM-C8
Digital Input/Counter				
Input Channels		4		No Digital Input
Type		Wet Contact (Sink)	Wet Contact (Source)	
On Voltage Level		+4.0 VDC to +50 VDC	+3.5 VDC to +50 VDC	
Off Voltage Level		+1 VDC Max.		
Input Impedance		10 KΩ, 0.66 W		
Counters	Channels	4		
	Max. Count	65535 (16-bit)		
	Max. Frequency	100 Hz		
	Min. Pulse Width	5 ms		
Over-voltage Protection		70 VDC		
Digital Output				
Output Channels		4		8
Type		Isolated Open Emitter (Source)	Isolated Open Collector (Sink)	
Max. Load Current		650 mA/channel	700 mA/channel	
Load Voltage		+10 VDC to +40 VDC	+3.5 to +50 VDC	
Over-voltage Protection		47 VDC	60 VDC	
Overload Protection		Yes		1.4 A
Short Circuit Protection		Yes		
Power-on Value		Yes, Programmable		
Safe Value		Yes, Programmable		
Isolation				
Intra-module Isolation, Field-to-Logic		3750 VDC		
Power Requirements				
Reverse Polarity Protection		Yes		
Powered from Terminal Block		Yes, 10 to 30 VDC		
Consumption		0.5 W Max.		0.8 W Max.

(Refer to Section 1.3.1 - Common Specification)

1.3.4 Specification - Multifunction I/O

Module Name	AI	AO	DI	DO
tM-DA1P1R1	**-**	**1**	**1**	**Relay: 1**
tM-AD4P2C2	**Voltage: 2 ; Current: 2**	**-**	**2**	**2**

(Refer to Section 1.3.1 - Common Specification)

Module Name		tM-DA1P1R1	tM-AD4P2C2
Analog Input			
Input Channel	Voltage	No Analog Input	2 single-ended
	Current		
Type	Voltage		±1 V, ±2.5 V, ±5 V, ±10 V
	Current		±20 mA, 0 to 20 mA, 4 to 20 mA
Resolution	Normal Mode		14-bit
	Fast Mode		12-bit
Sampling Rate	Normal Mode		10 Hz total
	Fast Mode		200 Hz total
Accuracy	Normal Mode		± 0.1%
	Fast Mode		± 0.5%
Zero Drift			±20 µV/°C
Span Drift			±25 ppm/°C
Input Impedance	Voltage		10 MΩ
	Current		136 Ω
Over Protection	Voltage		120 VDC
	Current		Yes, 50 mA at 110 VDC
Analog Output			
Output Channels		1	No Analog Output
Type		0 to 10 V, 0 to 20 mA, 4 to 20 mA	
Resolution		12-bit	
Accuracy		+/-0.1% of FSR	
DA Output Response Time		10 ms	
Voltage Output Capability		20 mA	
Current Load Resistance		500 Ω	

Module Name		tM-DA1P1R1	tM-AD4P2C2
Digital Input/Counter			
Input Channel		1	2
Type	Dry Contact	Source	-
	Wet Contact	Sink/Source	Source
Wet Contact	ON Voltage Level	+3.5 VDC to +50 VDC	
	OFF Voltage Level	+1 VDC Max.	
Dry Contact	ON Voltage Level	Close to GND	-
	OFF Voltage Level	Open	
Input Impedance		10 KΩ, 0.5 W	
Counters	Channels	1	2
	Max. Count	65535 (16-bit)	
	Max. Input Frequency	100 Hz	50 Hz
	Min. Pulse Width	5 ms	10 ms
Over-voltage Protection		70 VDC	
Digital/Relay Output			
Channels		1	2
Type		Power Relay, Form C	Isolated Open Collector (Sink)
Max. Load Current		NO : 10 A @ 250 VAC NC : 6 A @ 250 VAC	700 mA/Channel
Load Voltage		-	+3.5 VDC to +50 VDC
Over Protection	voltage	-	60 VDC
	Current		1.4 A (with short circuit protection)
Short Circuit Protection			Yes
Operating Load Voltage Range		250 VAC or 30 VDC	-
Operating Time		15 ms Max.	
Release Time		5 ms Max.	
Mechanical Endurance		1 X 10^7 OPS	
Electrical Endurance		5 X 10^4 OPS	
Power On Value		Yes, Programmable	
Safe Value		Yes, Programmable	
Power Requirements			
Reverse Polarity Protection		Yes	
Powered from Terminal Block		Yes, 10 to 30 VDC	
Consumption		1.8 W Max.	1 W Max.

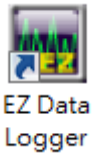
1.4 Supported Software

ICP DAS provides some free software tool and development tools as follows, including:

1.4.1 DCON Utility Pro

DCON Utility Pro is used to search, configure, and test the I/O module through the RS-232/485 port. It supports DCON and Modbus protocols and can run on a Windows PC or a PAC. (Refer to Chapter 3 for more information.)

1.4.2 EZ Data Logger



EZ Data Logger is the software that ICP DAS provides for users to easily build a SCADA system on Windows 2000/XP/Vista. It comes with two versions, "Lite" & "Professional". The Lite version is not only full-functioned but free to all ICP DAS users!
(Refer to Chapter 4 for more information.)

EZ Data Logger is a small data logger software. It can be applied to small scale remote I/O system. With its user-friendly interface, users can quickly and easily build a data logger software without any programming skill.

Functions:

- Support DCON, Modbus RTU, Modbus ASCII, and Modbus TCP protocols
- Support multiple COM Ports and TCP/IP connections
- Support Virtual Channel definition
- Support Control Logic (VB Script)
- Support Alarm Notifier (by sending SMS or E-Mail)
- Flexible workgroup configuration
- Real-time data trend (with zoom in and zoom out)
- Provide a Layout view
- Provide IP Camera Viewer
- Access database supported (can be exported to Excel file or CVS file)
- Provide Reporter to print trend line or data
- Provide High/Low alarm with audio warning
- Can search for DCON and Modbus modules
- Provide value scaling
- Support three-level authority management
- Do not need highly programming skills

1.4.3 OPC Server

NAPOPC_ST DA Server is a free OPC DA Server ("OPC" stands for "OLE for Process Control" and "DA" stands for "Data Access") for ICP DAS products.

Based on Microsoft's OLE COM (component object model) and DCOM (distributed component object model) technologies, NAPOPC_ST DA Server defines a standard set of objects, interfaces, and methods for use in process control and manufacturing automation applications to facilitate the interoperability.

Using NAPOPC_ST DA Server, the system integrates data with SCADA/HMI/Database software on the same computer and others. SCADA/ HMI/Database sends a request and NAPOPC DA Server fulfills the request by gathering the data of ICP DAS modules (License Free) and third-party devices (License Charge) to SCADA/HMI/Database.

For different OS of PAC products, ICP DAS provides several professional DA Servers:

Version	NAPOPC_ST	NAPOPC_XPE	NAPOPC_CE5	NAPOPC_CE6
Platform	Desktop Windows	Windows XP Embedded	Windows CE5	Windows CE6
Price	Free/$	Free	Free	Free

For more information please visit https://opc.icpdas.com

1.4.4 PacSDK

The PacSDK is a software development kit that contains header files, libraries, documents, and tools required to develop an application for XPAC, WinPAC, and ViewPAC series. Refer to section 6.2 for more information.

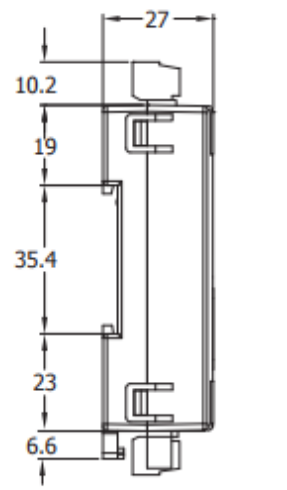
ICP DAS has released a new SDK, named PacSDK, which merged with and replaced the XPACSDK and the WinPACSDK. The applications that users developed can be easily applied to different PACs.

Chapter 2 Hardware Installation

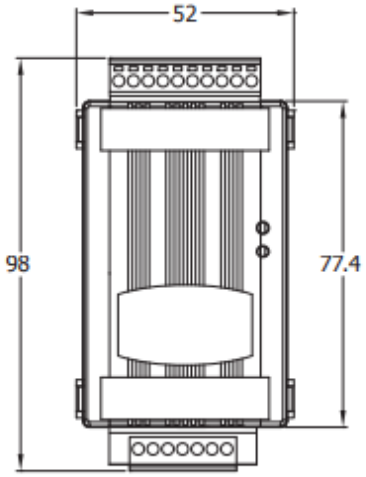
2.1 Mounting

2.2 Dimensions

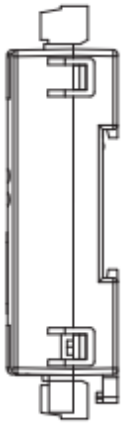
The following diagrams provide the dimensions of the tM series module and can be used as a reference when defining the specifications for any custom enclosures. All dimensions are in millimeters.



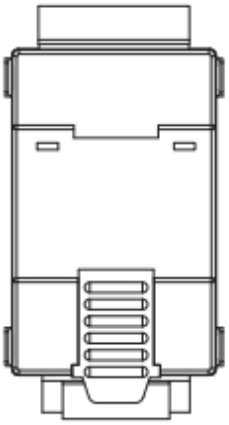
Left Side View



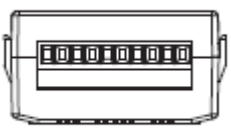
Front View



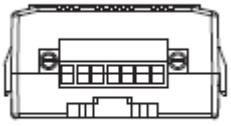
Right Side View



Rear View



Top View



Bottom View

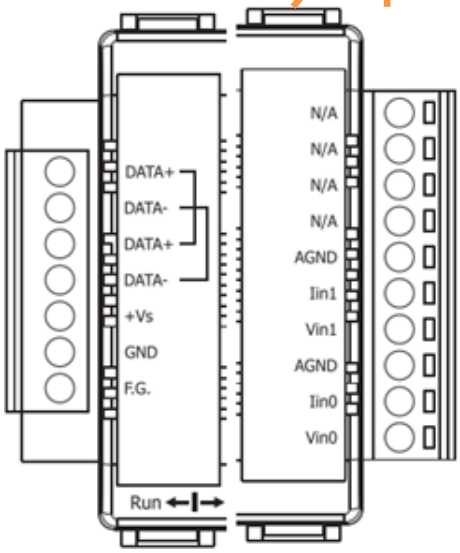
2.3 Wiring Connections

2.3.1 Wiring Recommendations

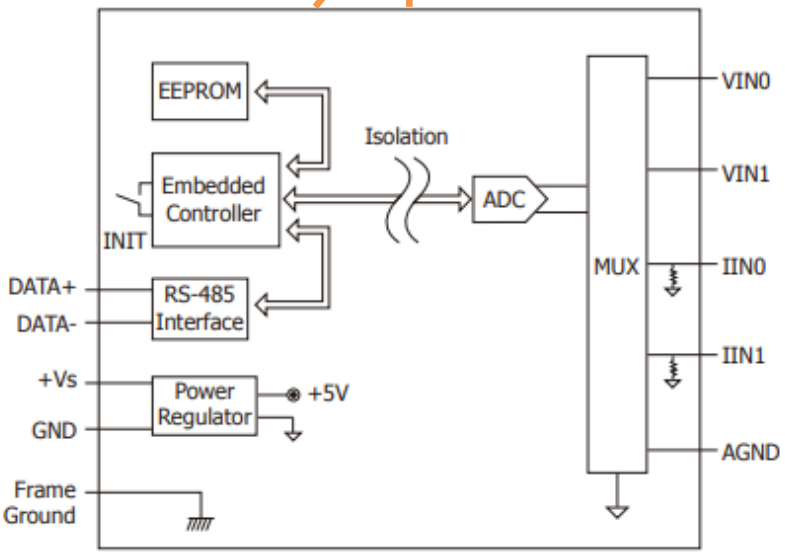
- Use 26-12 AWG wire for signal connections
- Strip the wire to a length of 7±0.5mm
- Use a crimp terminal for wiring
- Avoid high-voltage cables and power equipment as much as possible
- For RS-485 communication, use insulated and twisted pair 24 AWG wire, e.g., Belden 9841

2.3.2 tM-AD2 Wiring

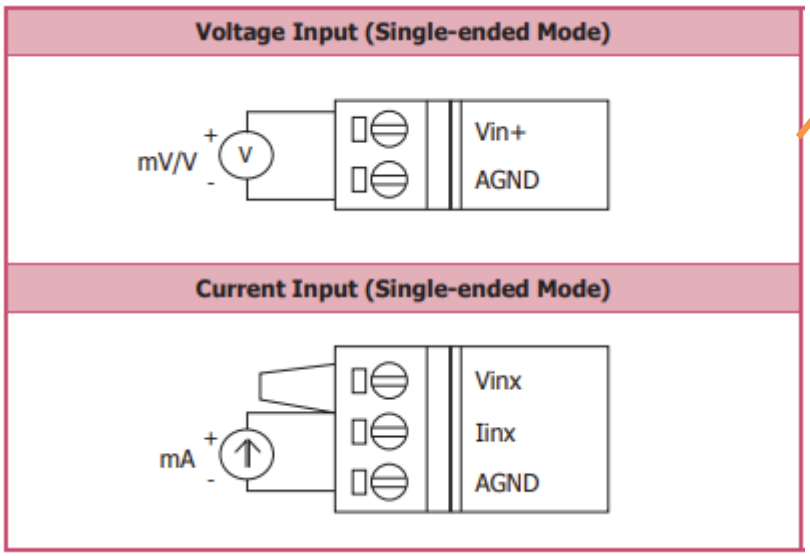
Pin Assignment



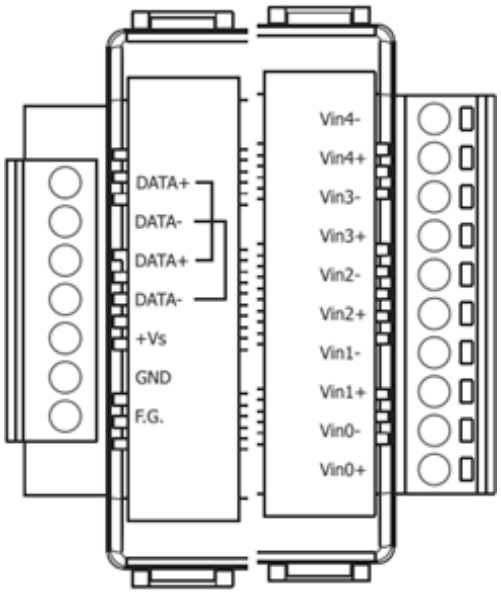
Block Diagram

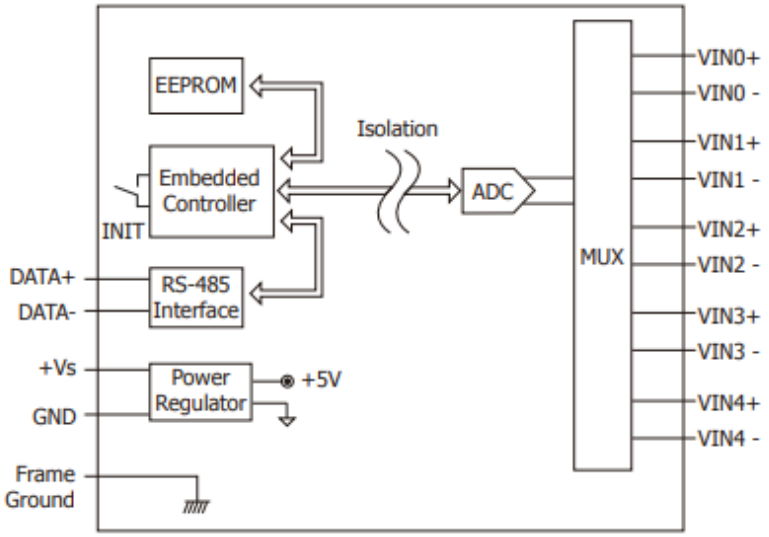


Wiring Diagram

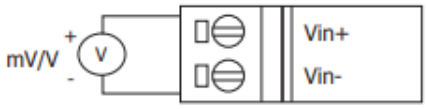


2.3.3 tM-AD5 Wiring

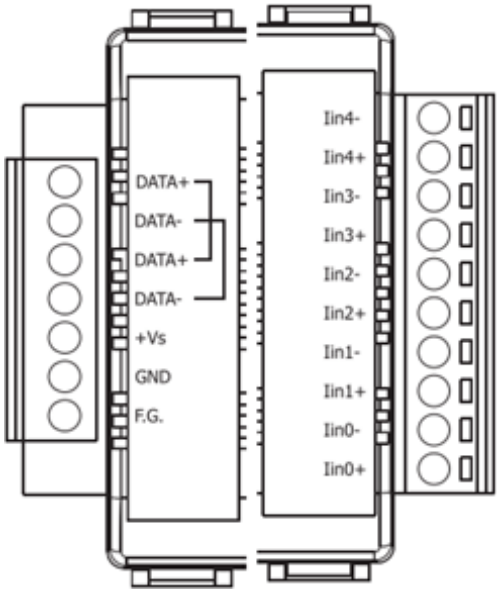


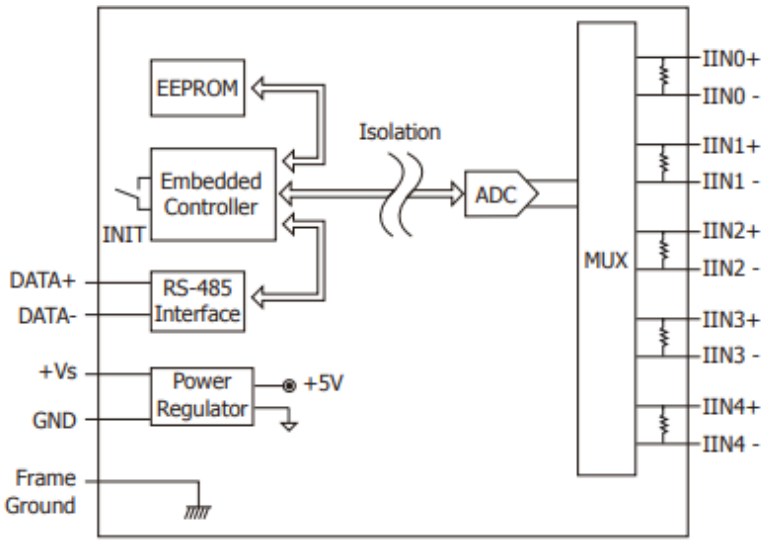


Voltage Input (Differential Mode)

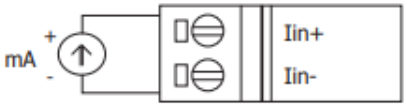


2.3.4 tM-AD5C Wiring

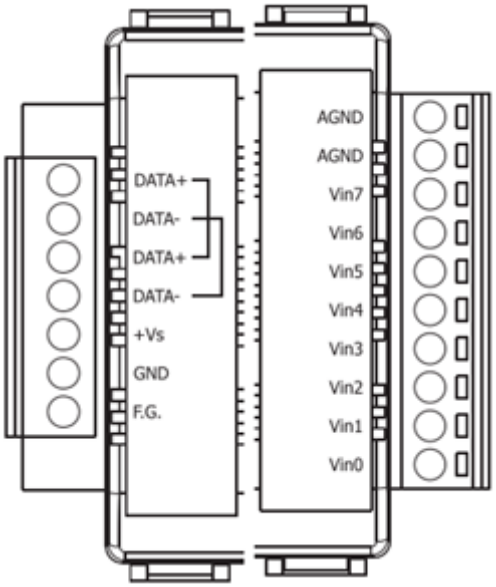


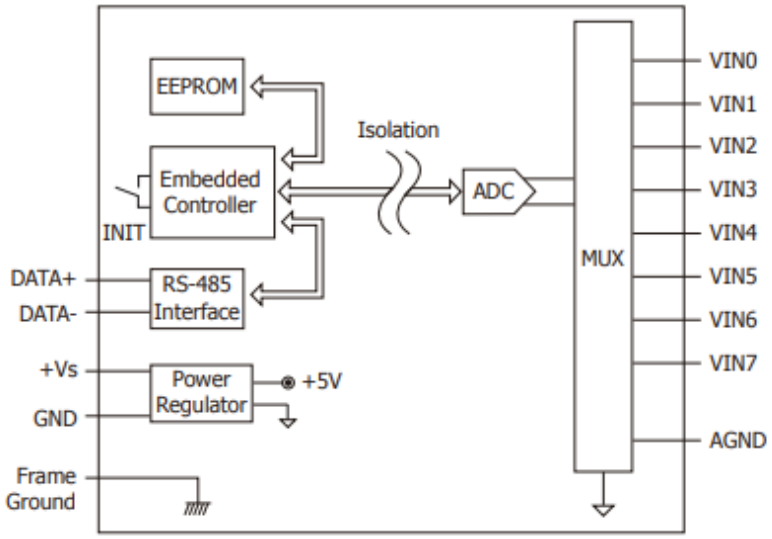


Current Input

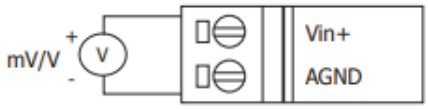


2.3.5 tM-AD8 Wiring

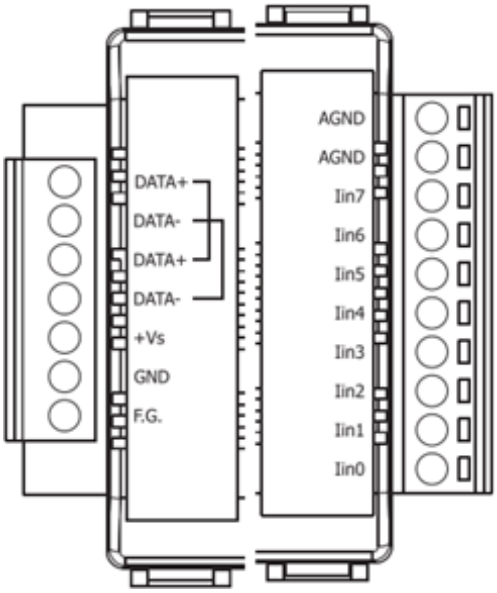


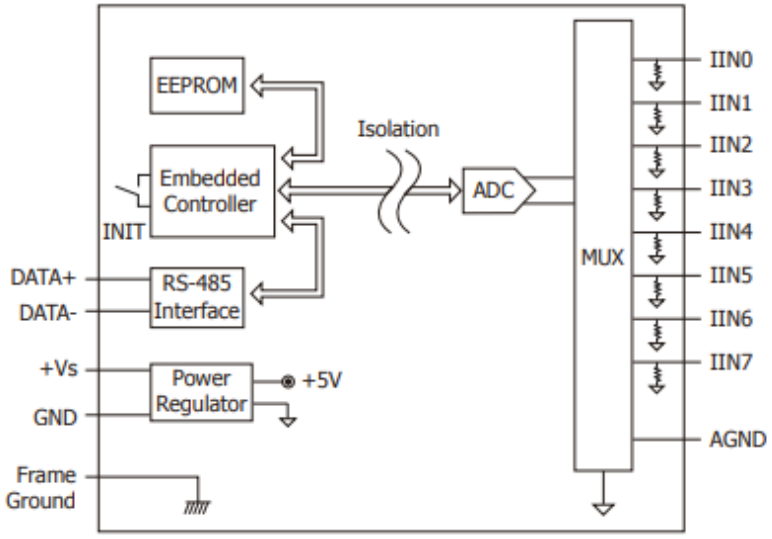


Voltage Input (Single-ended Mode)

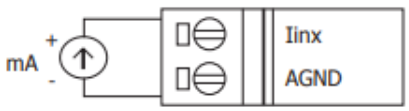


2.3.6 tM-AD8C Wiring

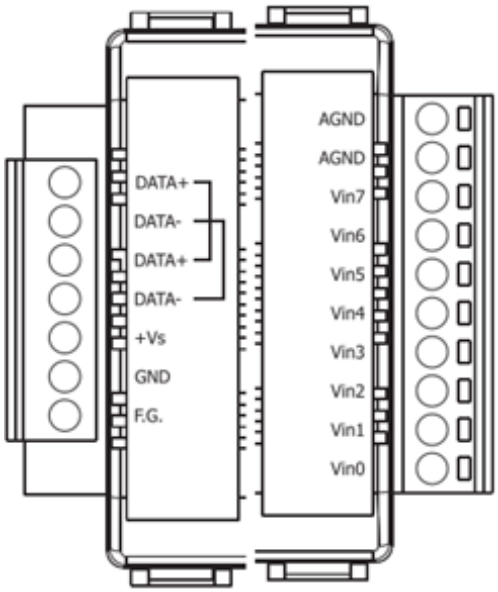


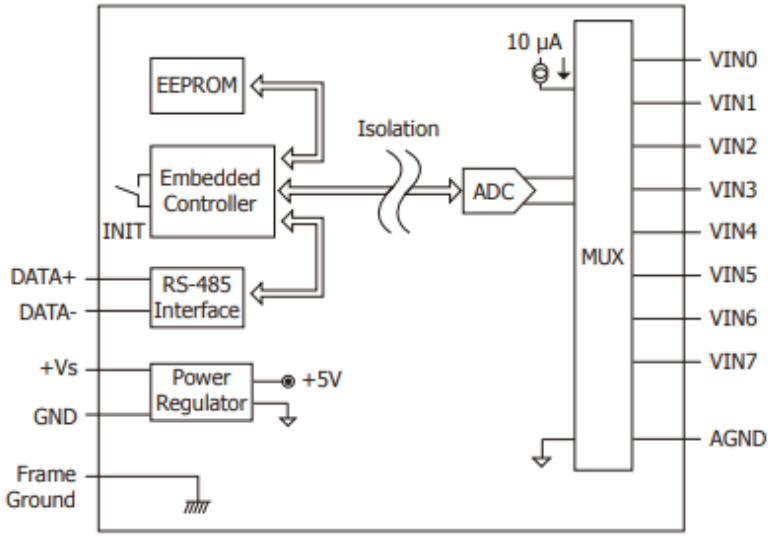


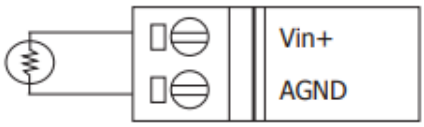
Current Input (Single-ended Mode)



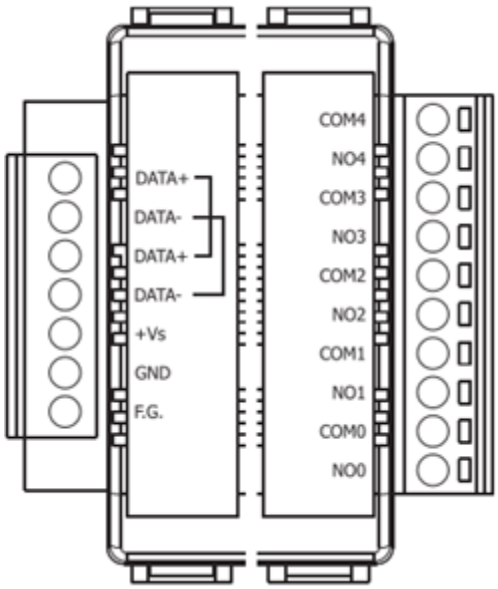
2.3.7 tM-TH8 Wiring

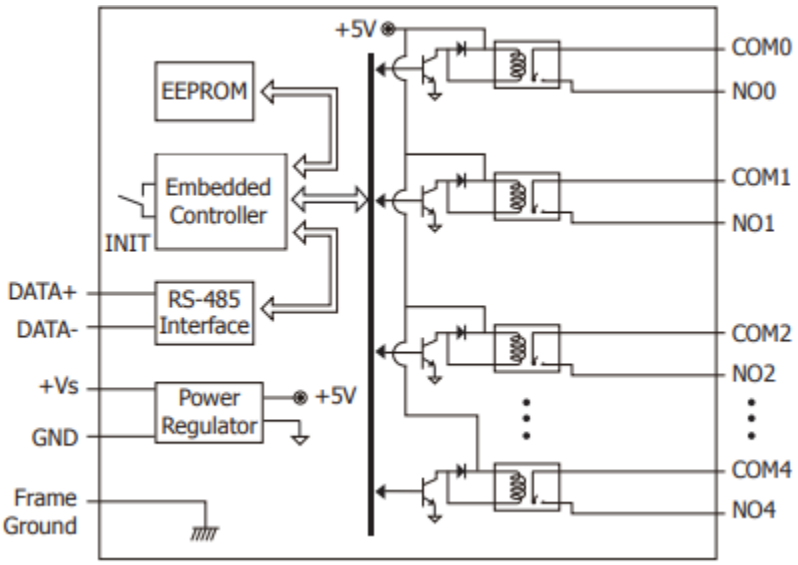


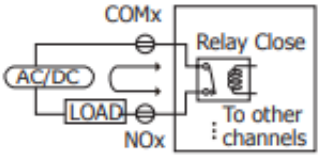
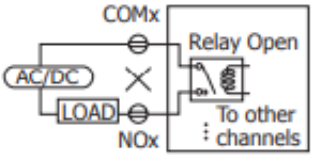


Thermistor Input


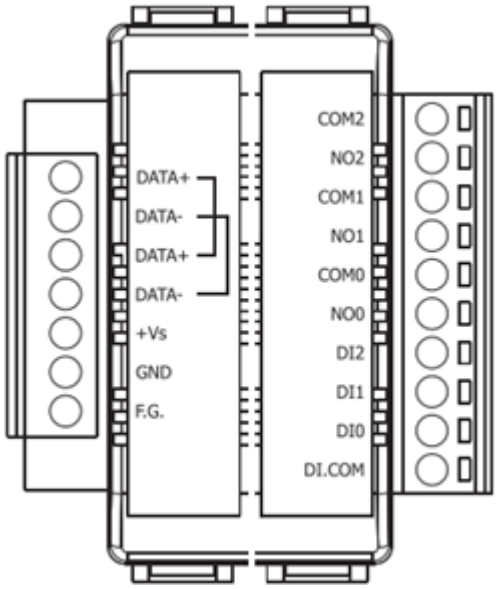
2.3.8 tM-R5 Wiring

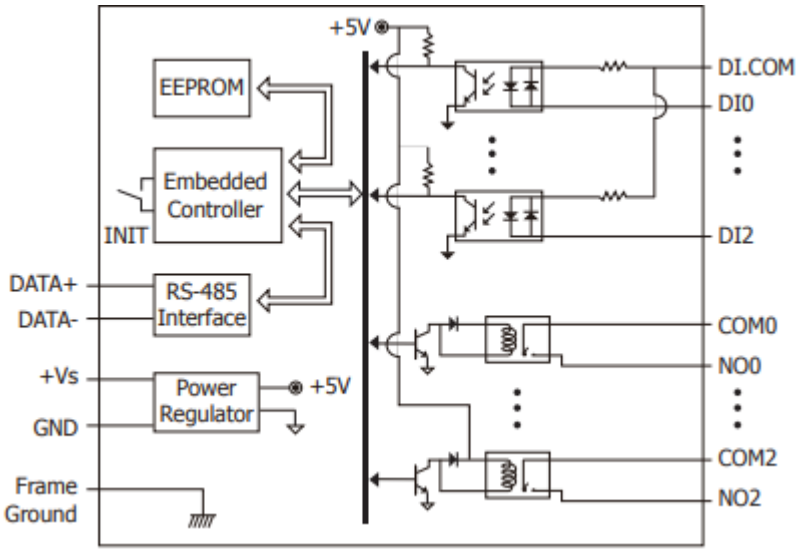


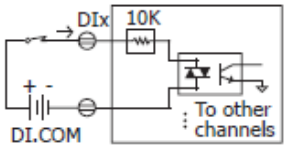
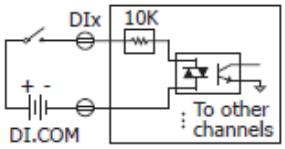
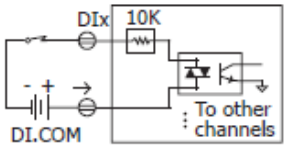
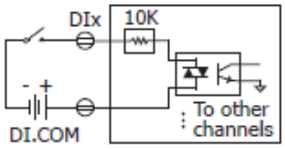


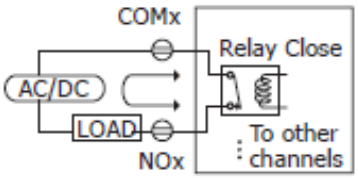
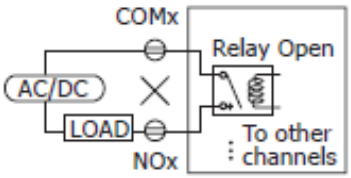
Power Relay	ON State Readback as 1	OFF State Readback as 0
Relay Output		

2.3.9 tM-P3R3 Wiring

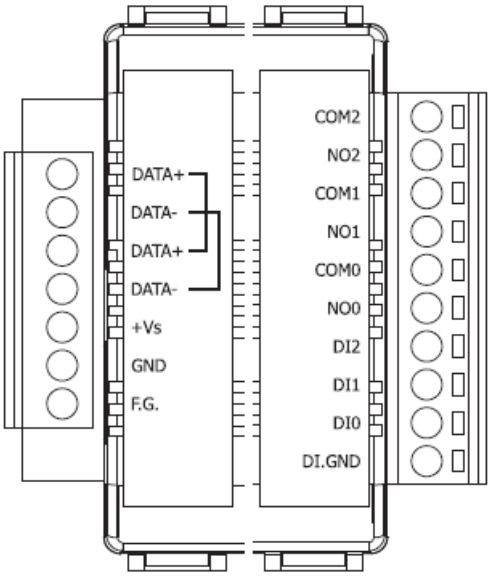


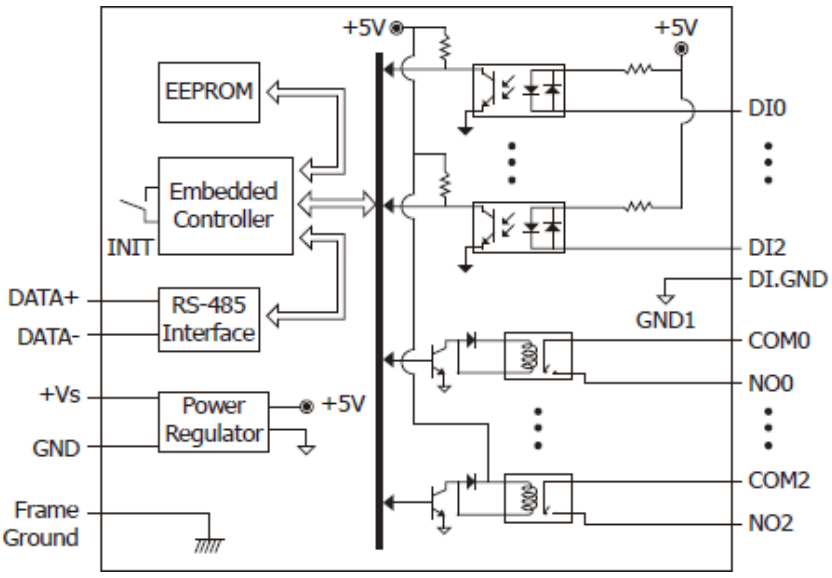


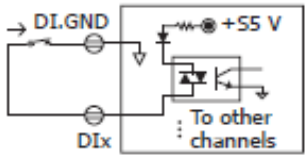
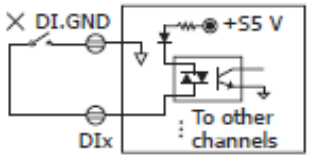
Input Type	ON State Readback as 1	OFF State Readback as 0
Wet Contact (Sink)	+3.5 ~ +50 VDC	OPEN or <1 VDC
		
Wet Contact (Source)	+3.5 ~ +50 VDC	OPEN or <1 VDC
		

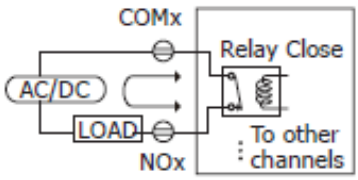
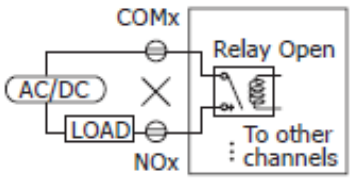
Power Relay	ON State Readback as 1	OFF State Readback as 0
Relay Output		

2.3.10 tM-PD3R3 Wiring

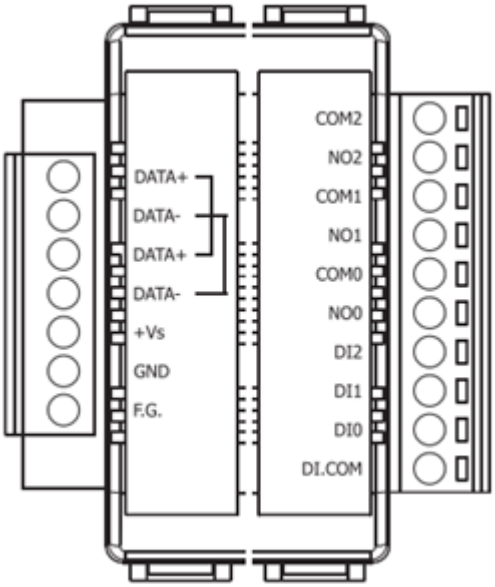


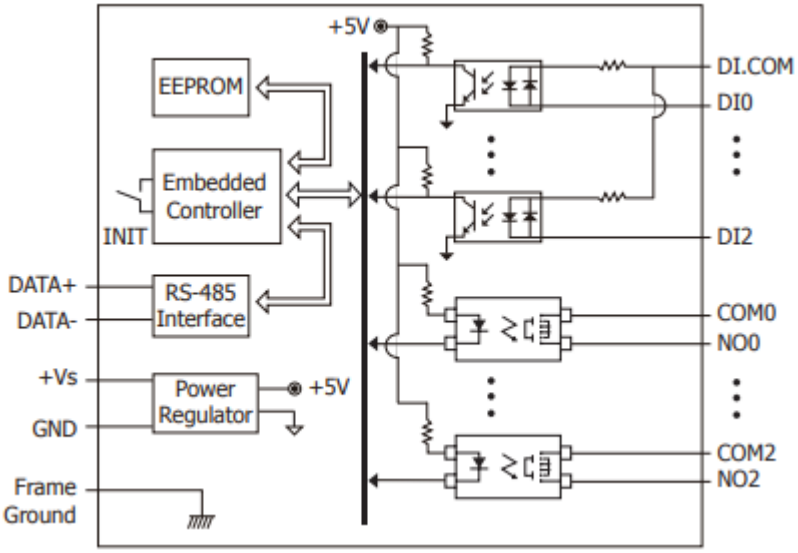


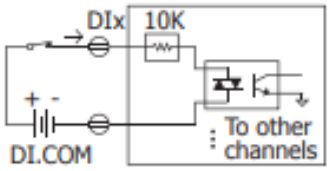
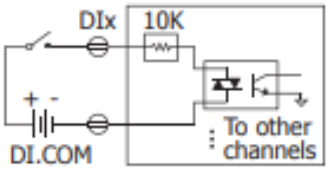
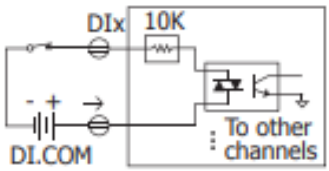
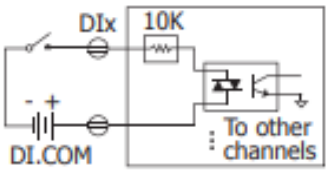
Input Type	ON State Readback as 1	OFF State Readback as 0
	Close to GND	Open
Dry Contact		

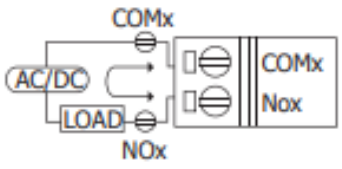
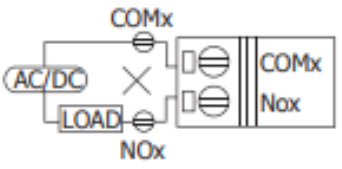
Power Relay	ON State Readback as 1	OFF State Readback as 0
Relay Output		

2.3.11 tM-P3POR3 Wiring

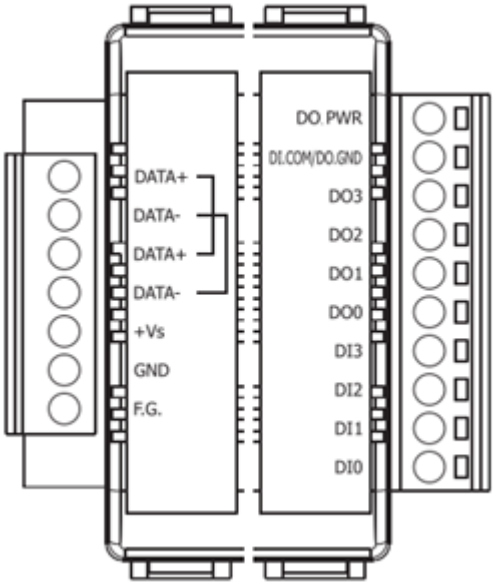


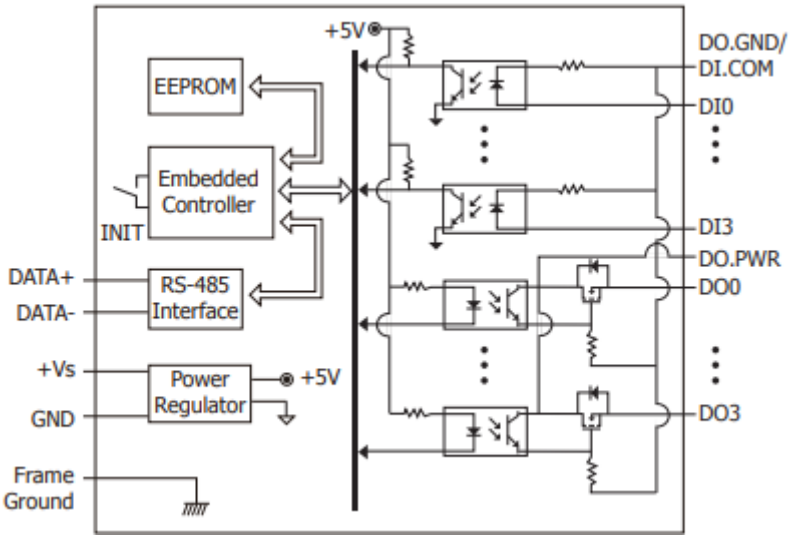


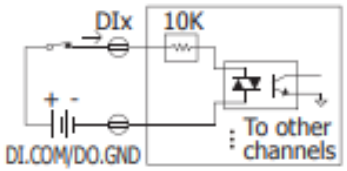
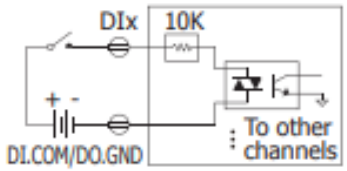
Digital Input/ Counter	ON State Readback as 1	OFF State Readback as 0
Sink	+3.5 ~ +50 V_{DC}	OPEN or <1.0 V_{DC}
		
Source	+3.5 ~ +50 V_{DC}	OPEN or <1.0 V_{DC}
		

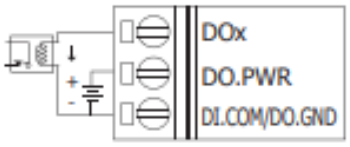
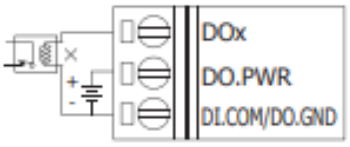
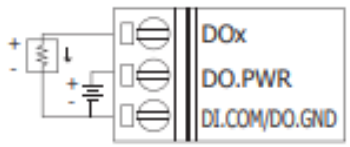
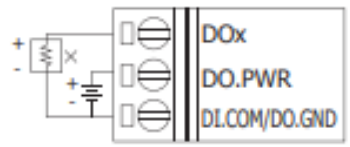
PhotoMOS Relay	ON State Readback as 1	OFF State Readback as 0
Relay Output		

2.3.12 tM-P4A4 Wiring

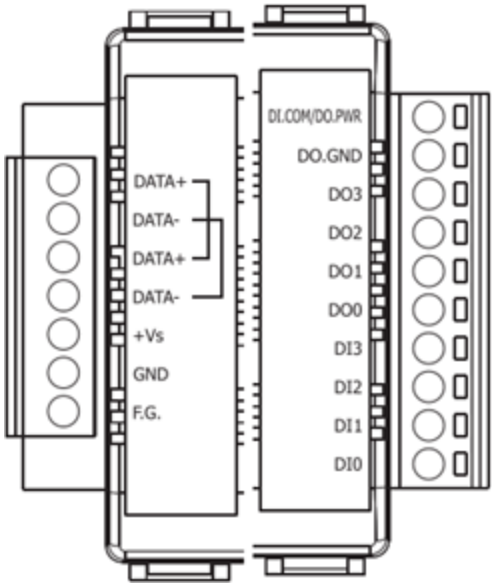


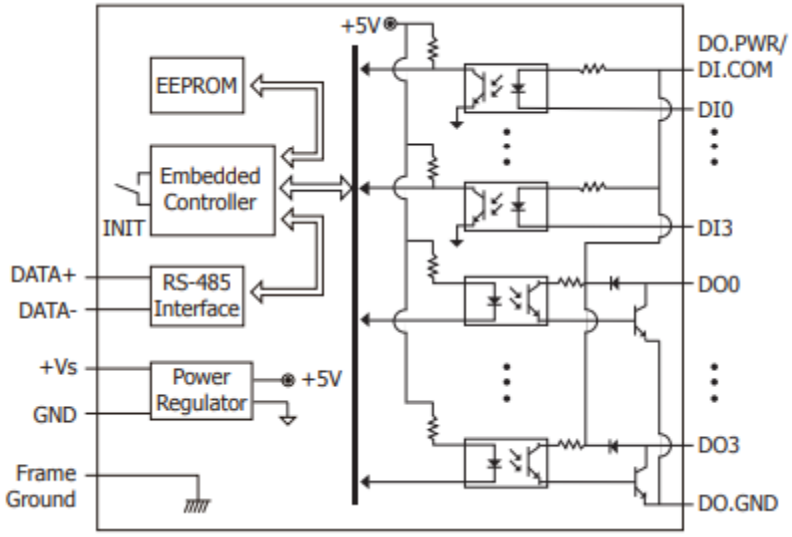


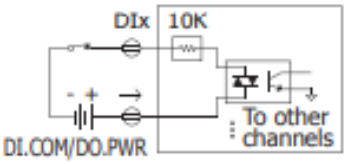
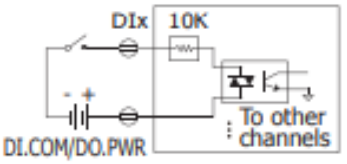
Digital Input/ Counter	ON State Readback as 1	OFF State Readback as 0
Sink	+3.5 ~ +50 VDC	OPEN or <1 VDC
		

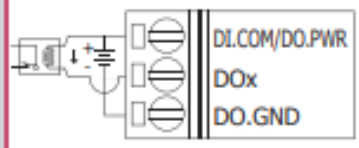
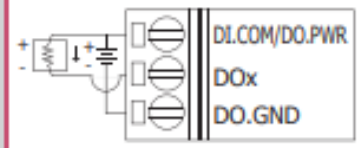
Output Type	ON State Readback as 1	OFF State Readback as 0
Drive Relay	Relay ON	Relay OFF
		
Resistance Load		
		

2.3.13 tM-P4C4 Wiring

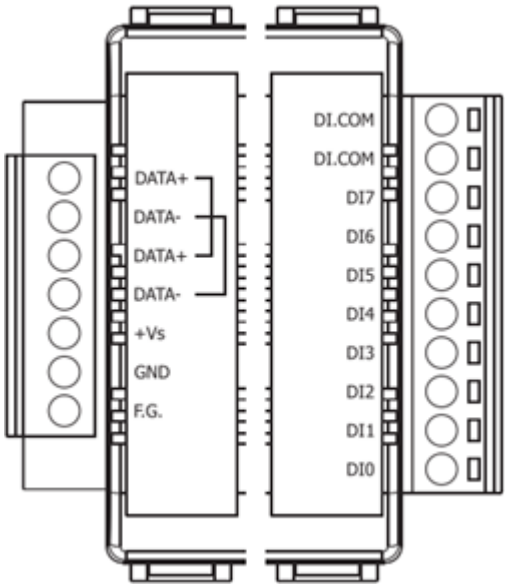


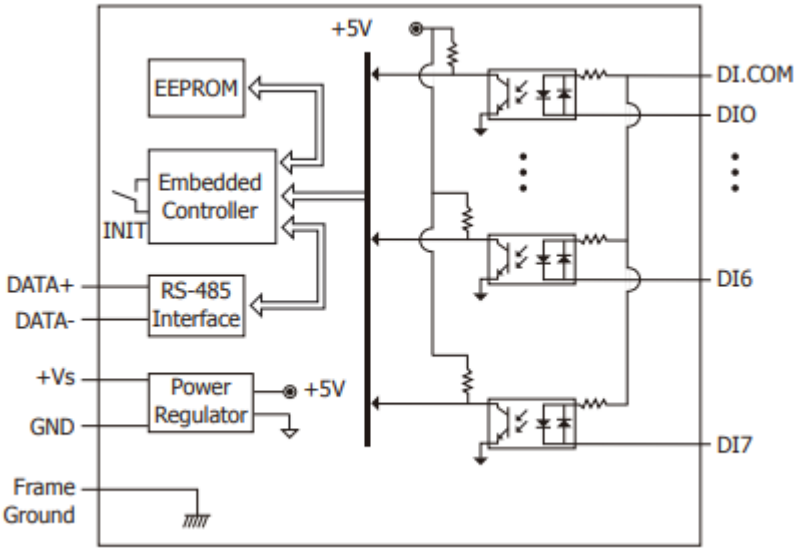


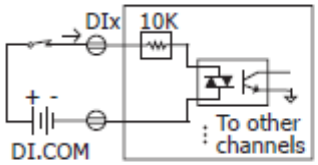
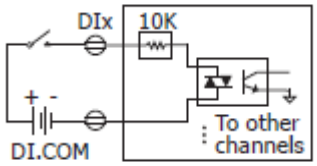
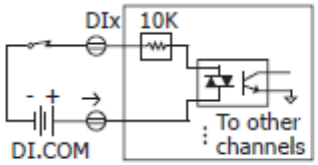
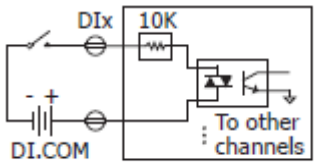
Digital Input/ Counter	ON State Readback as 1	OFF State Readback as 0
	+3.5 ~ +50 VDC	OPEN or <1 VDC
Source		

Output Type	ON State Readback as 1	OFF State Readback as 0
	Relay ON	Relay OFF
Drive Relay		
Resistance Load		

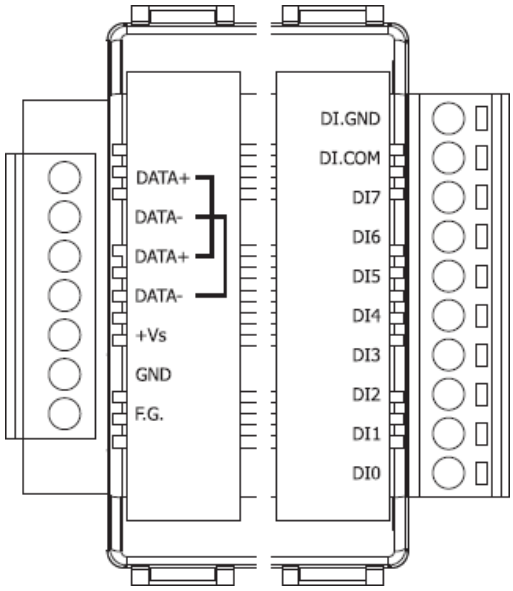
2.3.14 tM-P8 Wiring

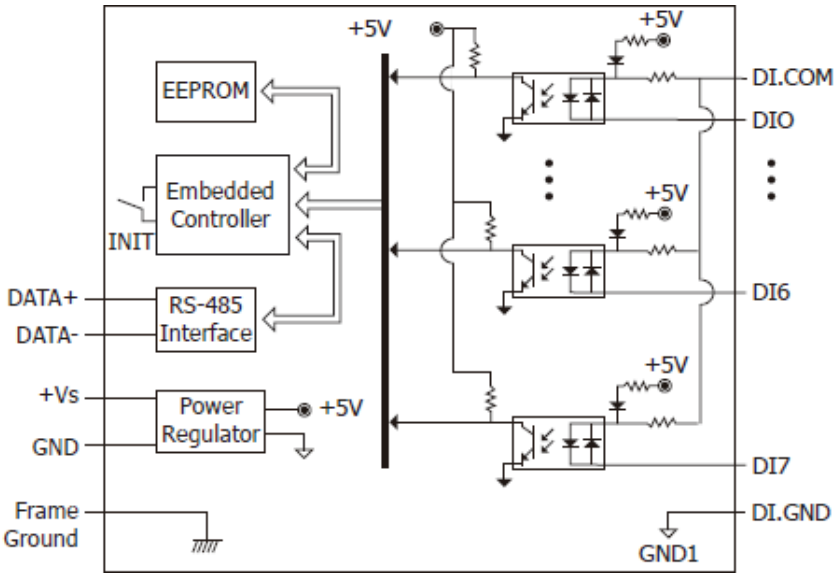


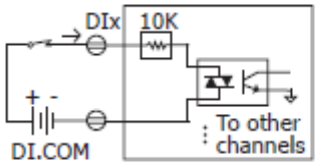
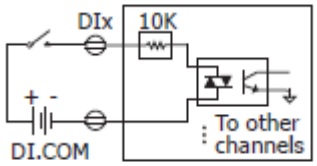
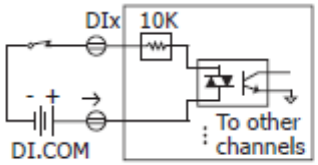
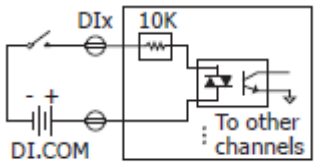


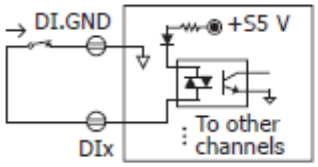
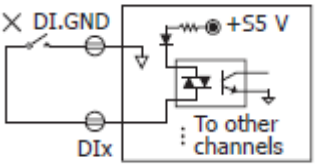
Input Type	ON State Readback as 1	OFF State Readback as 0
Wet Contact (Sink)	+3.5 ~ +50 VDC	OPEN or <1 VDC
		
Wet Contact (Source)	+3.5 ~ +50 VDC	OPEN or <1 VDC
		

2.3.15 tM-PDW8 Wiring

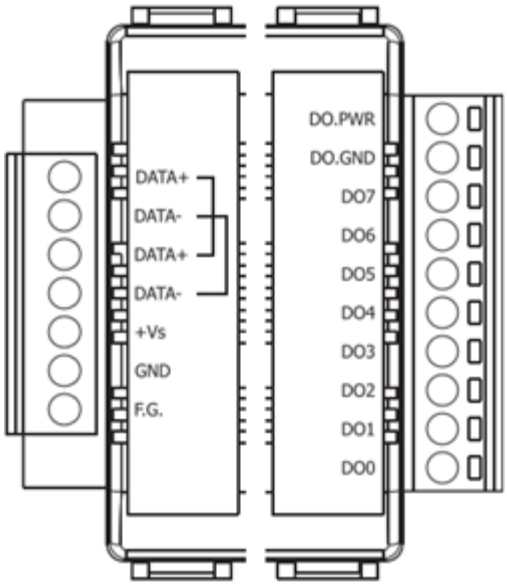


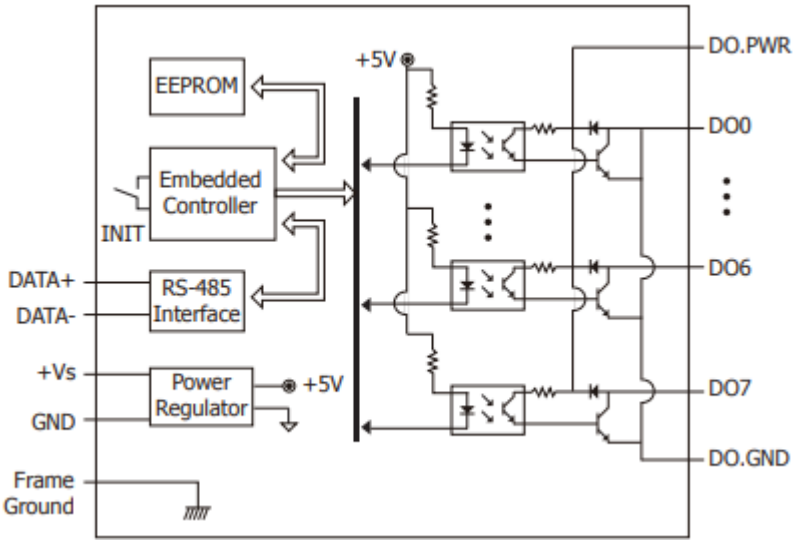


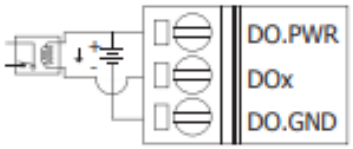
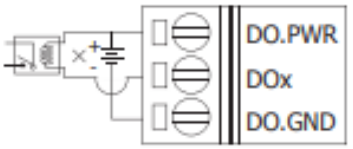
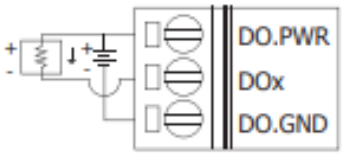
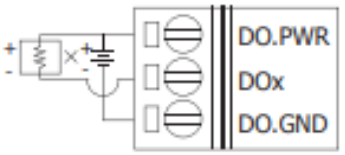
Input Type	ON State Readback as 1	OFF State Readback as 0
Wet Contact (Sink)	+3.5 ~ +50 VDC	OPEN or <1 VDC
		
Wet Contact (Source)	+3.5 ~ +50 VDC	OPEN or <1 VDC
		

Input Type	ON State Readback as 1	OFF State Readback as 0
Dry Contact	Close to GND	Open
		

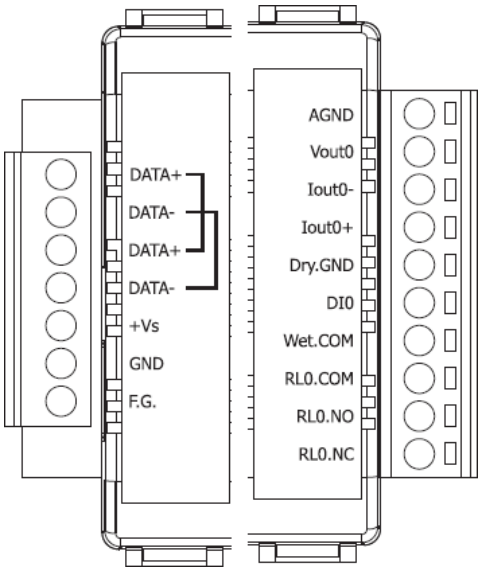
2.3.16 tM-C8 Wiring

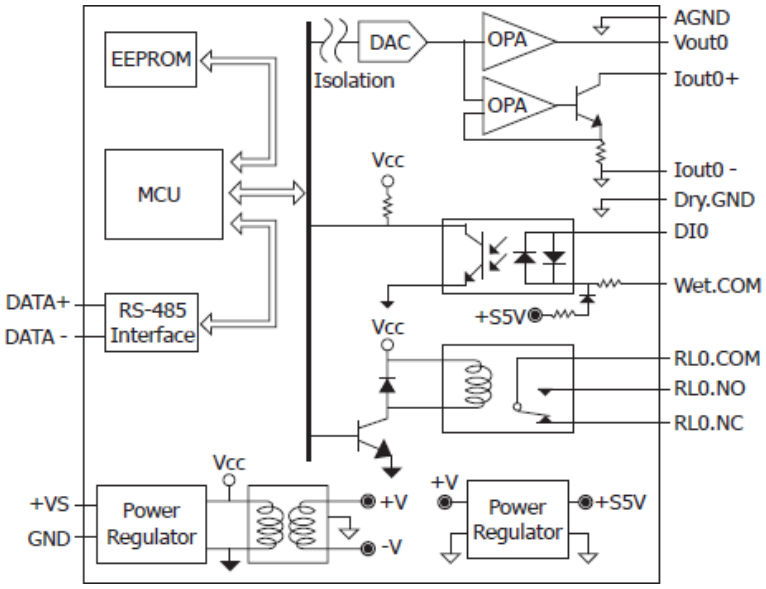




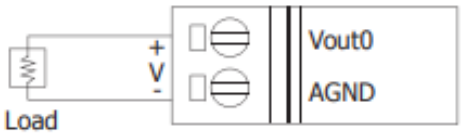
Output Type	ON State Readback as 1	OFF State Readback as 0
Drive Relay	Relay ON	Relay OFF
		
Resistance Load		
		

2.3.17 tM-DA1P1R1 Wiring

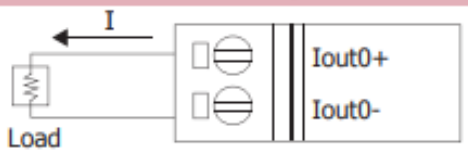




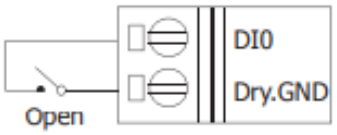
Voltage Output



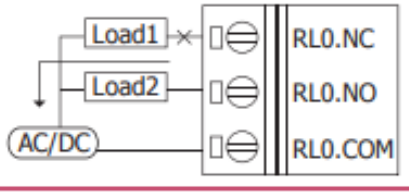
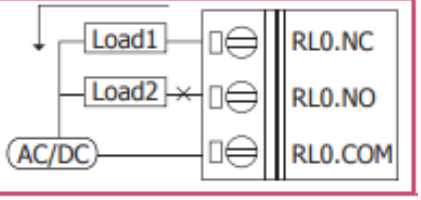
Current Output



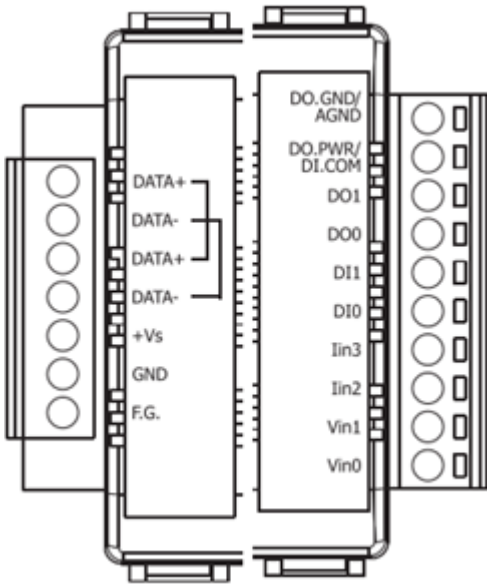
Digital Input/Counter

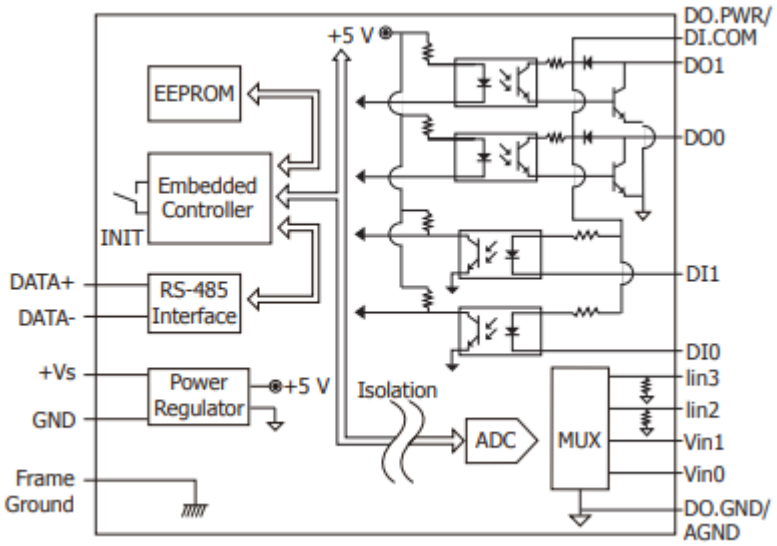
Wiring	ON State Readback as 1	OFF State Readback as 0
Dry Contact	Close to Dry.GND	Open
		
Wet Contact	+3.5 ~ +50 VDC	+1 VDC Max.
		

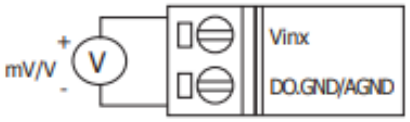
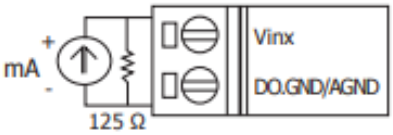
Relay Output

ON State	OFF State
	

2.3.18 tM-AD2P2C2 Wiring

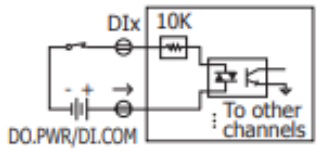
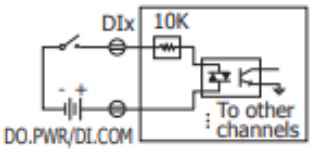


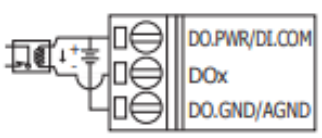
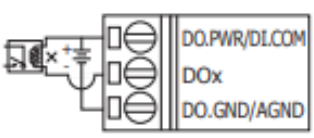
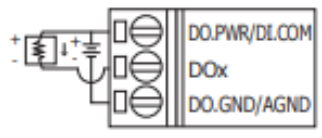
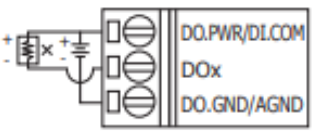


Voltage Input for Vin0 and Vin1	Current Input for Vin0 and Vin1
	

Requires Optional External 125 Ω Resistor

Current Input for Iin2 and Iin3

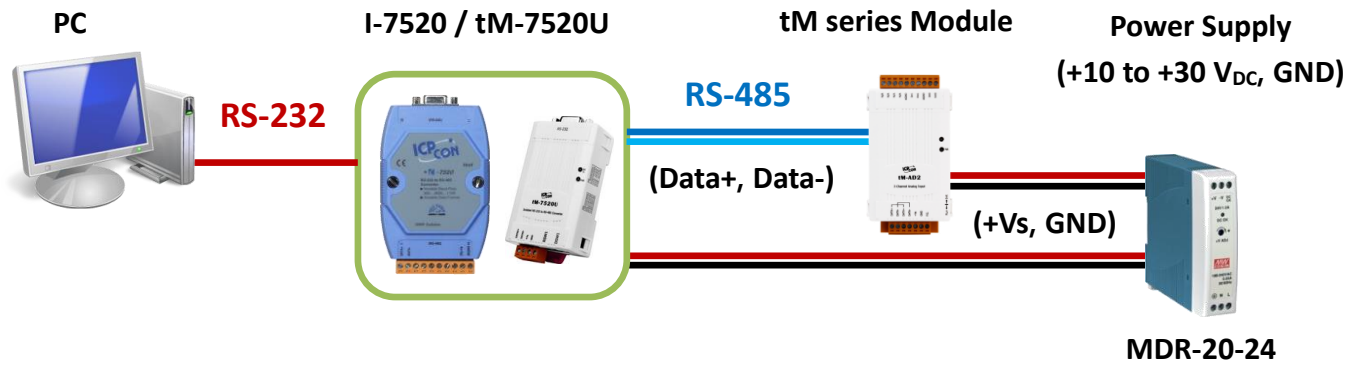

Digital Input/ Counter	ON State Readback as 1	OFF State Readback as 0
Source	+3.5 ~ +50 V_{DC}	OPEN or <1 V_{DC}
		

Output Type	ON State Readback as 1	OFF State Readback as 0
Drive Relay	Relay ON	Relay OFF
		
Resistance Load		

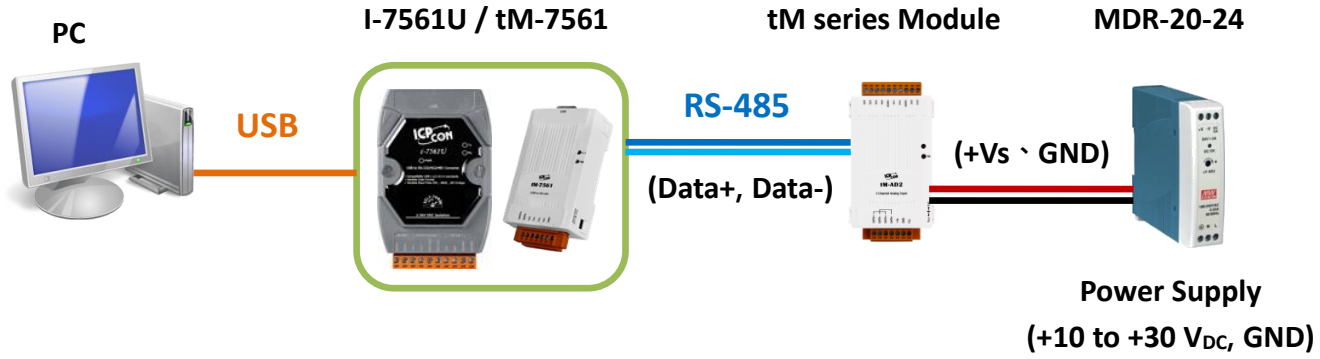
2.4 Hardware Connection

Before using tM series module, it must be configured by using DCON Utility Pro. tM series equips an RS-485 port that can be used to connect to a PC by using an RS-232 (or USB) to RS-485 converter.

1) Using **RS-232** to **RS-485** converter



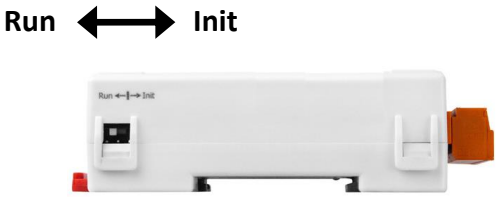
2) Using **USB** to **RS-485** converter



2.5 Operating Switch and Parameters

tM series provides two operating modes, i.e., Run and Init, which can be set by sliding the switch to the Init or Run position on the right side of modules.

Operating Switch



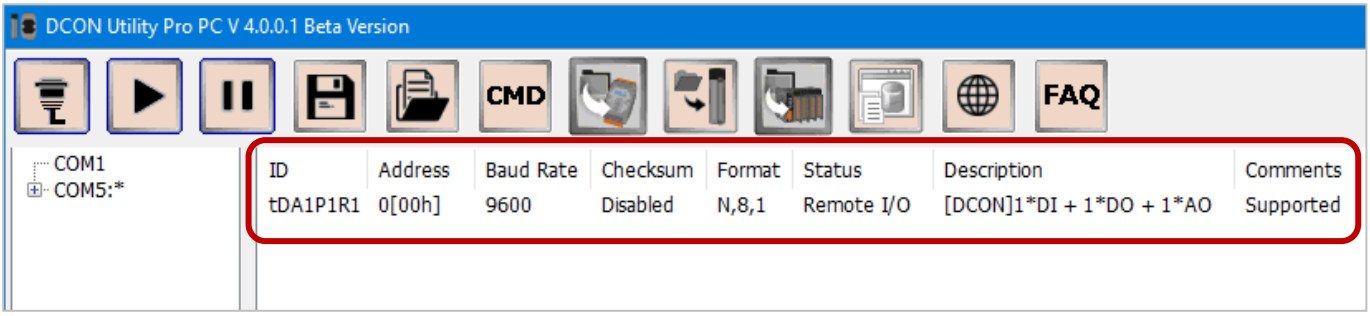
Run Mode:

Normally, the operating switch of tM series module is set to “Run”.

Init Mode:

If the communication parameters of the module are unknown, adjusting the switch to the “**Init**” position and then reboot the module. The setting values are always as follows:

Protocol: DCON, Address: 0, Baud Rate: 9600 (N,8,1), Checksum: Disable.



Default Communication parameters:

Parameters	Factory default settings (Run Mode)	Fixed Init settings (INIT Mode)
Protocol	Modbus RTU	DCON
Address	1	0
Baud Rate	9600 bps	
Parity	n,8,1-no parity	
Checksum	Disable	

Chapter 3 Software Tool - DCON Utility Pro

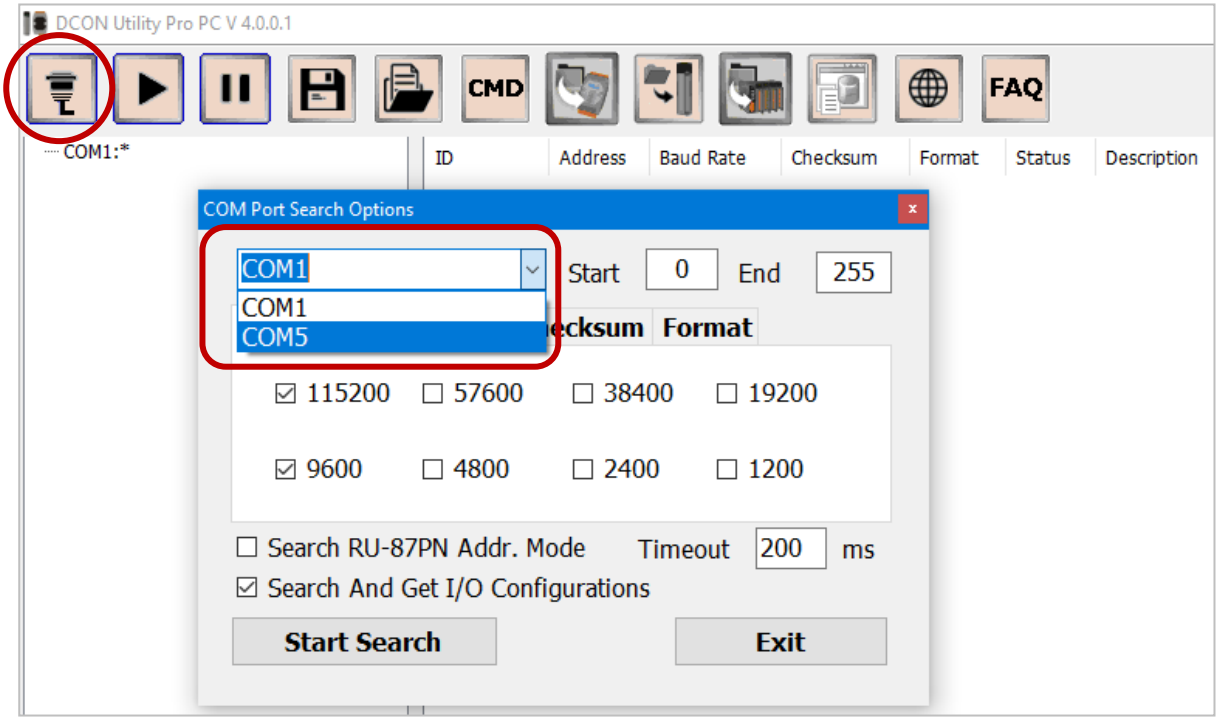
First, using DCON Utility Pro to configure the communication parameters and I/O settings for each tM series module according to the application. DCON Utility Pro is a software tool that supports DCON and Modbus RTU/ASCII protocols and can be used to search, configure, and test I/O module.

Download the software and user manual on the web site. DCON Utility Pro can run on PC or PAC.
https://www.icpdas.com/en/product/guide+Software+Utility_Driver+DCON__Utility__Pro

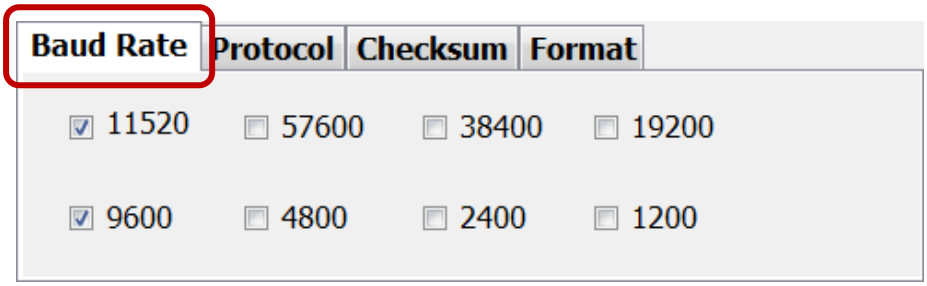
3.1 Search I/O Modules

In this case, PC (COM Port) and I/O module (RS-485) are connected through a tM-7561 (USB to RS-485 converter). Make sure that both PC and I/O module are connected properly, and the power supply is turned on. (The version is v3.0)

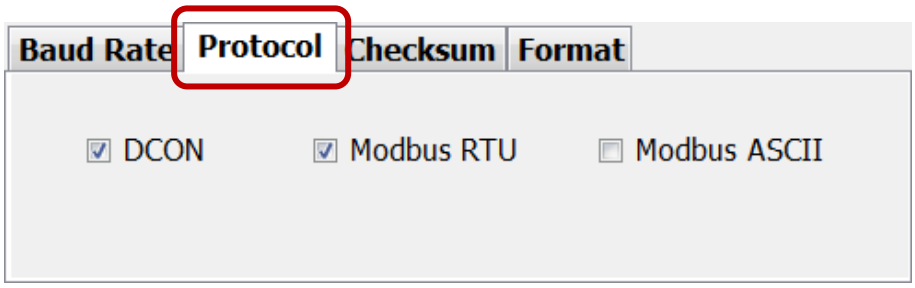
Step 1: Choose the COM port number that is used on PC and set conditions such as Baud Rate, Protocol, Checksum, and Format for searching.



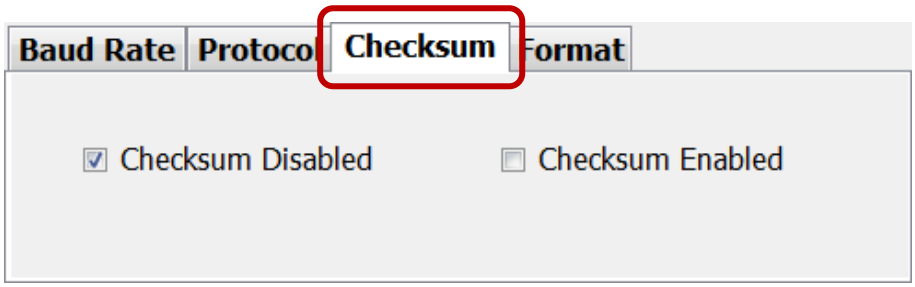
Baud Rate: The default Baud Rate setting of the module is 9600, also you can check multiple options.



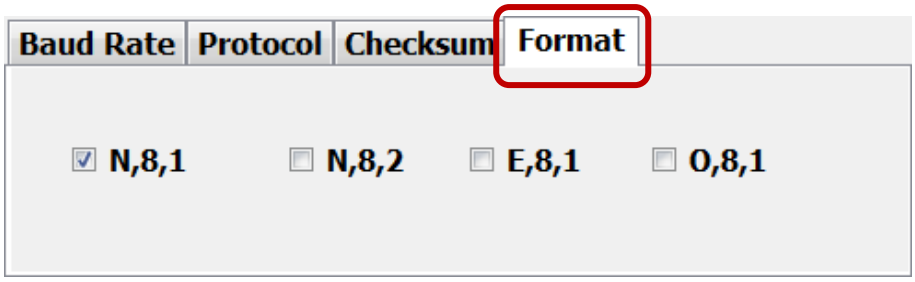
Protocol: When the module startup in “Init” mode, the default Protocol setting is DCON.
When the module startup in “Run” mode, the default Protocol setting is Modbus RTU.



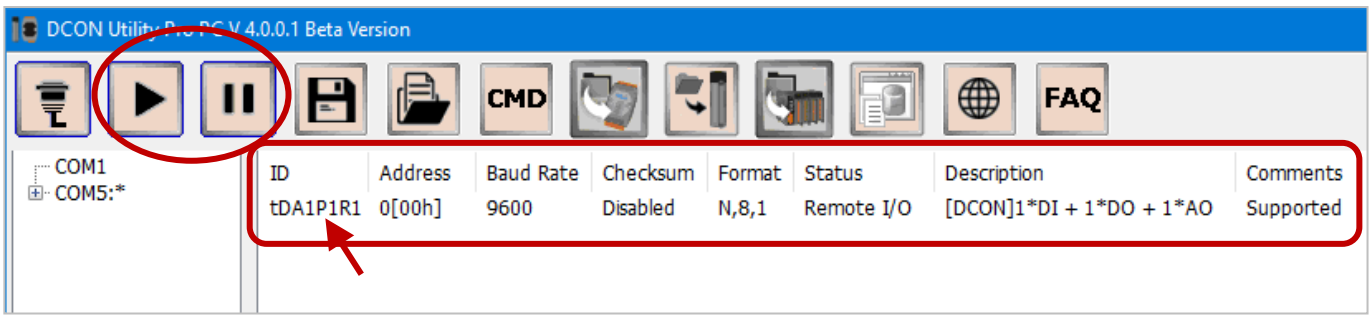
Checksum: The default Checksum setting is disabled.



Format: The default Format setting is “N,8,1”.



Step 2: After specifying search conditions, click the Start Search ( ) button to search the module.
Click Stop Search ( ) to stop searching, and click the model (ID) to display the configuration window.



3.2 “Configuration” Page

After searching and found the module, users can configure or test the I/O module.

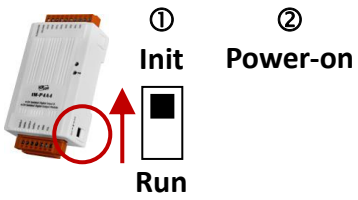
Note: DCON Utility Pro V3 (or later) support to open multiple setting window in Run mode.

In DCON Utility Pro, the “Configuration" page for each module can be used to set the communication parameters such as Protocol, Address, Baud Rate, etc. The rest of the setting pages for each module may be different so the following sections will describe each according to the IO type of modules.

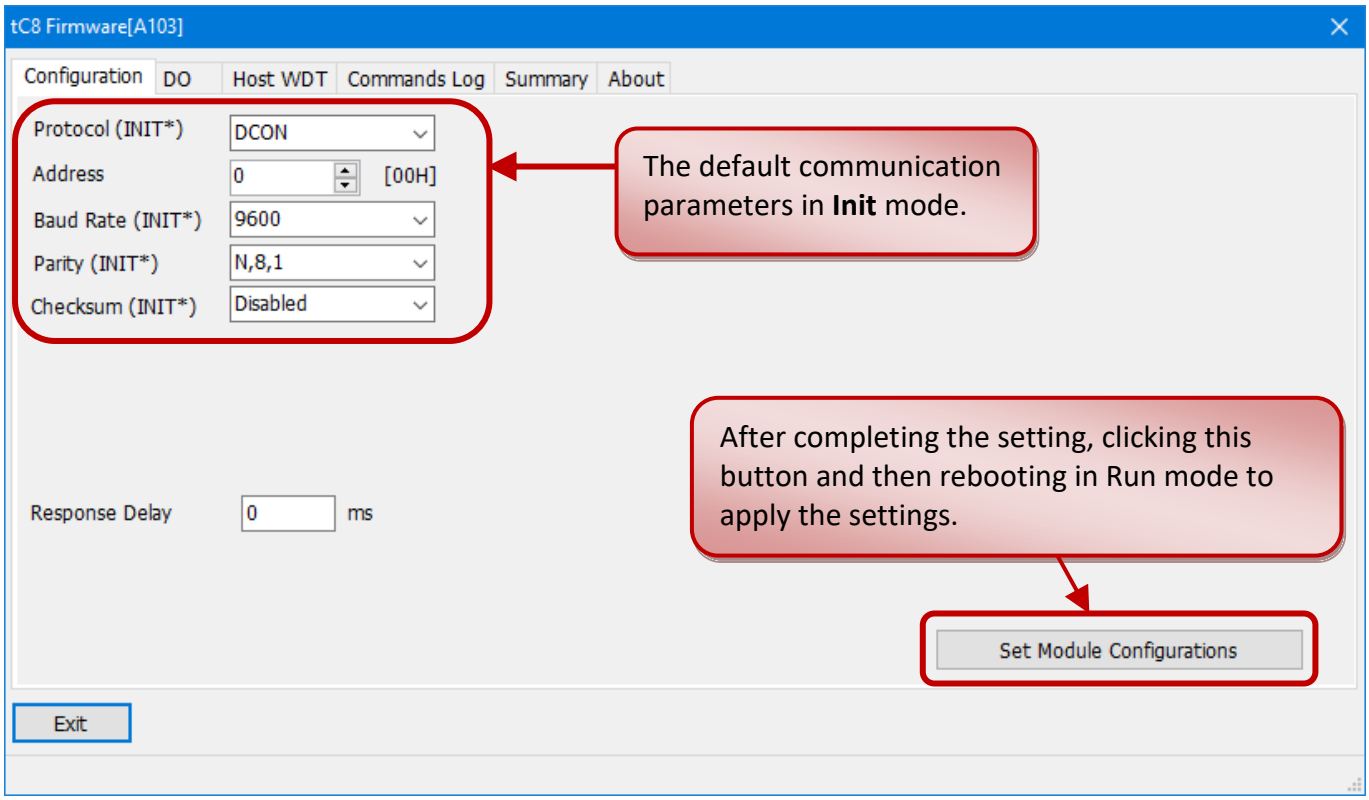
Model	Description	Section
tM-P3R3	3-channel Digital Input and 3-channel Relay Output Module	3.2.1
tM-PD3R3		
tM-P3POR3	3-channel Digital Input and 3-channel PhotoMOS Relay Output Module	
tM-P4A4	4-channel Digital Input and 4-channel Digital Output Module	
tM-P4C4	4-channel Digital Input and 4-channel Digital Output Module	
tM-R5	5-channel Relay Output Module	
tM-P8	8-channel Digital Input Module	
tM-PDW8		
tM-C8	8-channel Digital Output Module	
tM-AD2	2-channel Voltage/Current Input Module	3.2.2
tM-AD4P2C2	4-channel Analog Input, 2-channel Digital Input and 2-channel Relay Output Module	
tM-AD5	5-channel Differential Voltage Input Module	3.2.3
tM-AD8	8-channel Single-ended Voltage Input Module	
tM-AD5C	5-channel Differential Current Input Module	
tM-AD8C	8-channel Single-ended Current Input Module	
tM-TH8	8-channel Thermistor Input Module	3.2.4
tM-DA1P1R1	1-channel Analog Output, 1-channel Digital Input and 1-channel Relay Output Module	

3.2.1 Common Settings for tM-P3R3, PD3R3, P3POR3, P4A4, P4C4, R5, C8, P8, and PDW8

In Init mode, the module is searched with the default communication parameters. On the "Configuration" page, click the "Set Module Configurations" button, and then rebooting in Run mode to apply the settings.



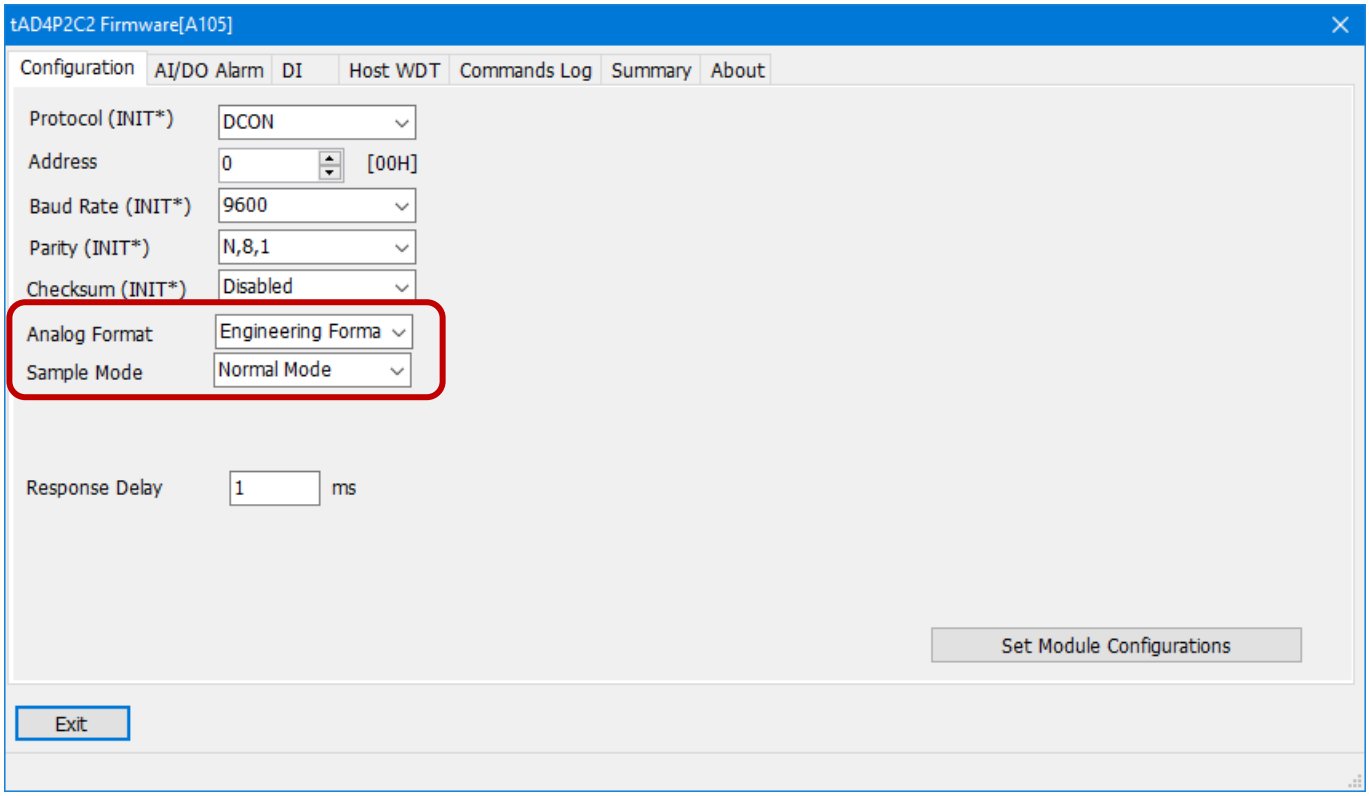
Note: Only when the Protocol is set to **DCON**, the items marked with (INIT*) must be set under the INIT mode. When using the Modbus protocol, these items can be set after adjusting the switch to Init, even if the module starts in Run mode. After completing the setting, adjusting the switch to Run and then rebooting.



Protocol	Specify the Protocol setting It can be DCON, Modbus RTU, and Modbus ASCII
Address	Specify the unique NET-ID for the module (Range: 0-255)
Baud Rate	Specify the Baud Rate setting It can be 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Parity	Specify the Parity, It can be "N,8,1-None Parity", "N,8,2-None Parity", "E,8,1-Even Parity", and "O,8,1-Odd Parity"
Checksum	Specify the Checksum, it can be Disabled or Enabled Note that the setting is "Disabled" when the Protocol is set to Modbus RTU/ASCII.
Respond Delay	Specify the delay time to respond. (Range: 0-30 ms)

3.2.2 AI items for tM-AD2 and AD4P2C2

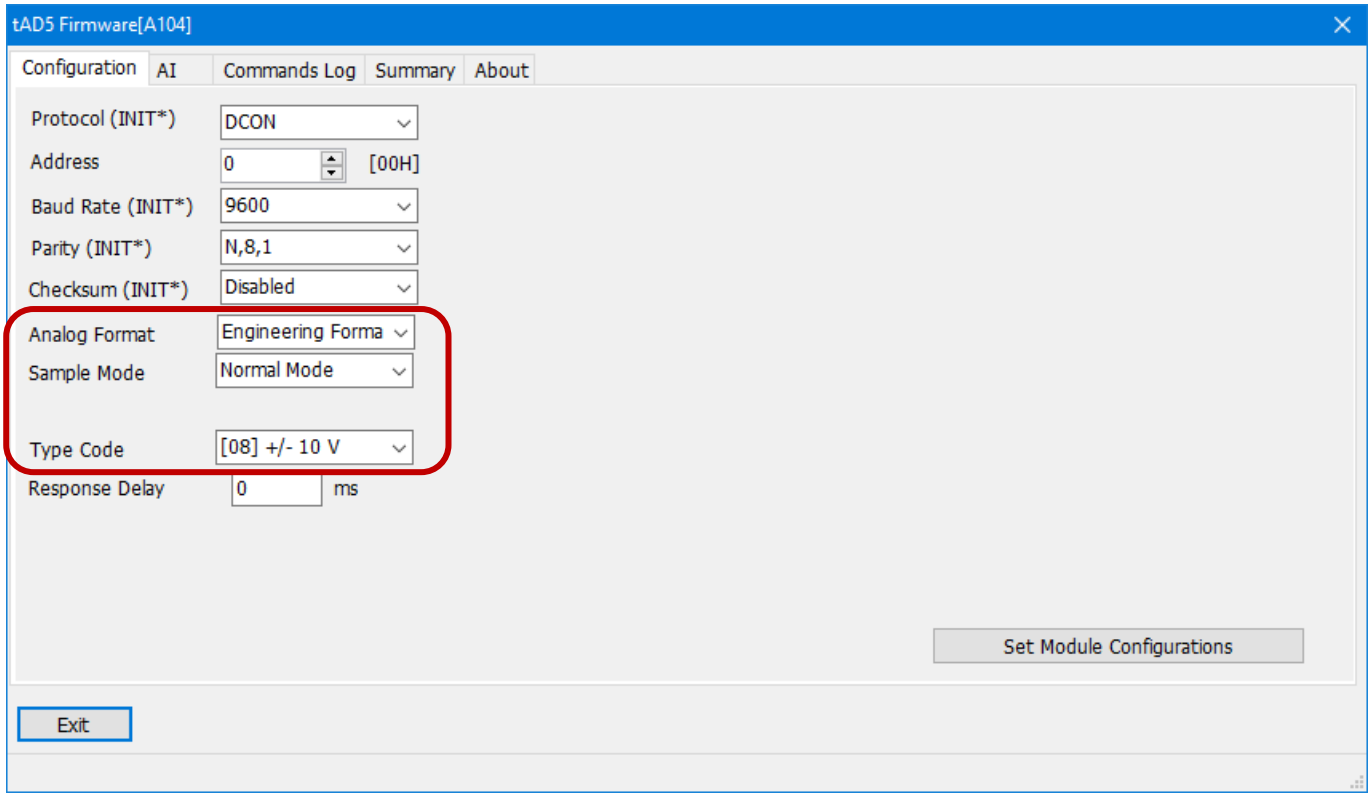
The “Configuration” page provides settings for communication parameters (Section 3.2.1) and analog data.



Analog Format	Specify the format for analog data DCON Protocol: It can be Engineering Format, Percent Format, or 2’s Complement Format Modbus RTU/ASCII Protocol: It can be Engineering Format or 2’s Complement Format
Sample Mode	Specify the mode for sampling data It can be Normal Mode or Fast Mode

3.2.3 AI Items for tM-AD5, AD8, AD5C, and AD8C

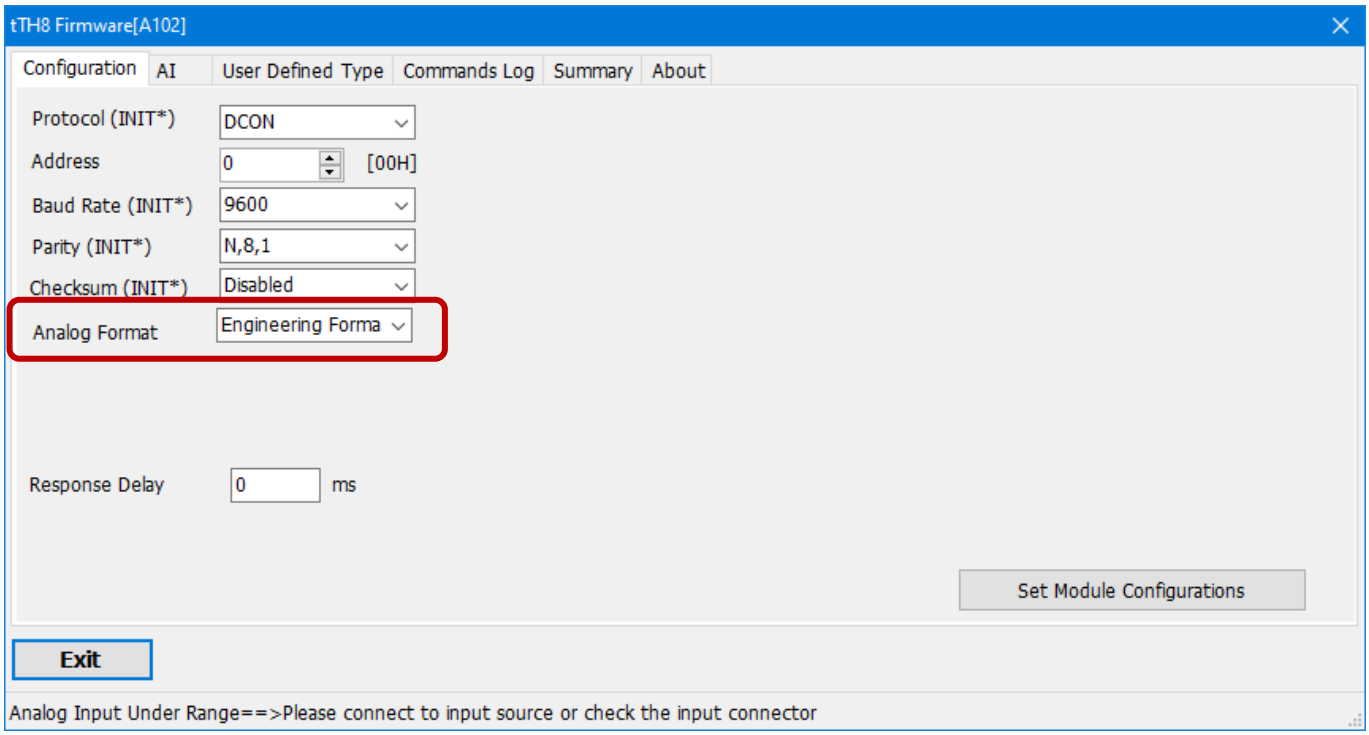
The “Configuration” page provides settings for communication parameters (Section 3.2.1) and analog data.



Analog Format	Specify the format for analog data DCON Protocol: It can be Engineering Format, Percent Format, or 2’s Complement Format Modbus RTU/ASCII Protocol: It can be Engineering Format or 2’s Complement Format
Sample Mode	Specify the mode for sampling data It can be Normal Mode or Fast Mode.
Type Code	Specify a type code with data range AD5: +/- 1 V, +/- 2.5 V, +/- 5 V, +/- 10 V AD8: 0 to 500 mV, 0 to 1 V, 0 to 2.5 V, 0 to 5 V, 0 to 10 V AD5C: +/- 20 mA, 0 to 20 mA, 4 to 20 mA AD8C: 0 to 20 mA, 4 to 20 mA (Reference: B.2 to B.5 Data Ranges)

3.2.4 AI Items for tM-TH8 and DA1P1R1

The “Configuration” page provides settings for communication parameters (Section 3.2.1) and analog data.



Analog Format	Configure the format of analog data. DCON Protocol: **tM-TH8**: It can be Engineering Format, Percent Format, 2’s Complement Format, or Ohms **tM-DA1P1R1**: It can be Engineering Format, Percent Format, or 2’s Complement Format Modbus RTU/ASCII Protocol: **tM-TH8, tM-DA1P1R1**: It can be Engineering Format or 2’s Complement Format

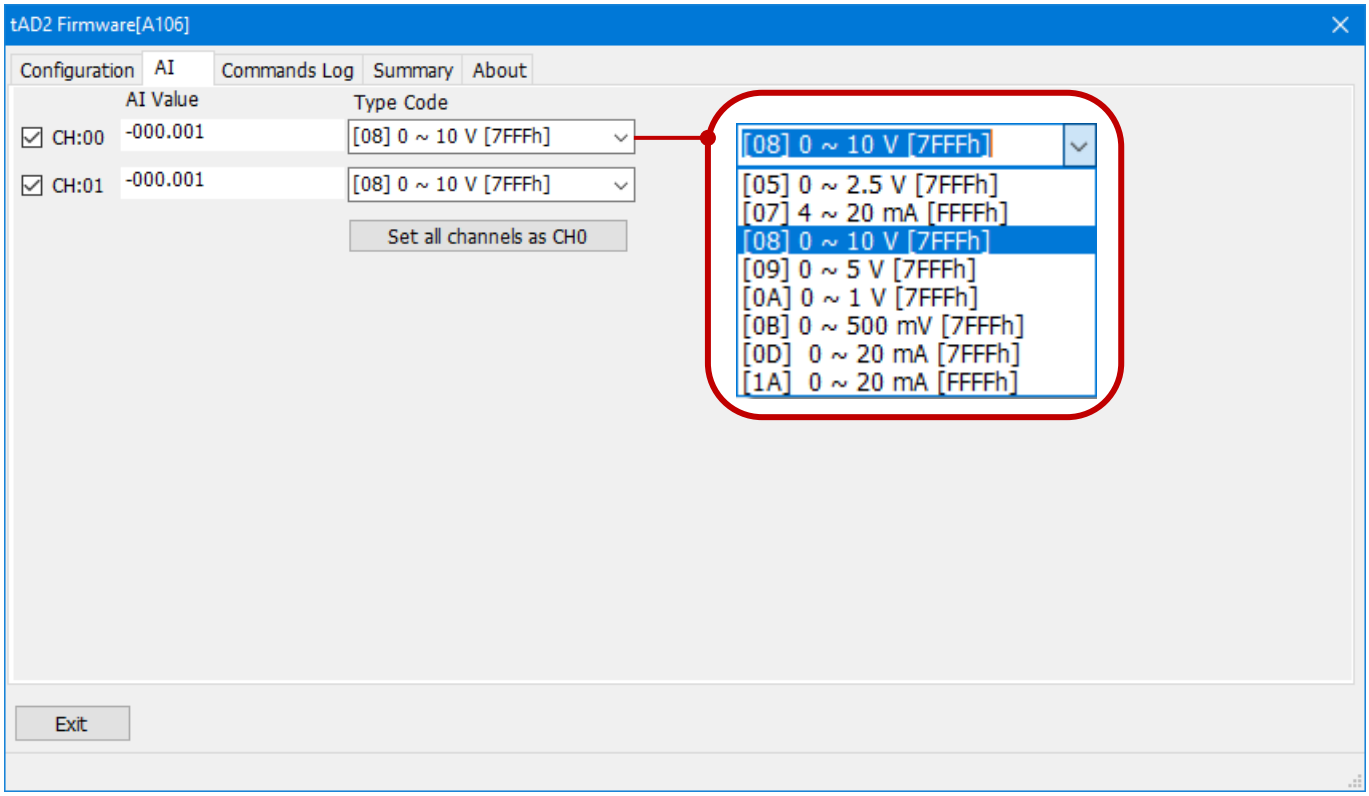
3.3 “AI” Page

The “AI” page provides settings for detecting or configuring analog input channels of the module.

Model	Description	Section
tM-AD2	2-channel Voltage/Current Input Module	3.3.1
tM-AD5	5-channel Voltage Input Module	3.3.2
tM-AD8	8-channel Voltage Input Module	
tM-AD5C	5-channel Current Input Module	
tM-AD8C	8-channel Current Input Module	
tM-TH8	8-channel Thermistor Input Module	3.3.3

3.3.1 AI Page01 for tM-AD2

The “AI” page can be used to detect data and specify the type code for each analog input channel.

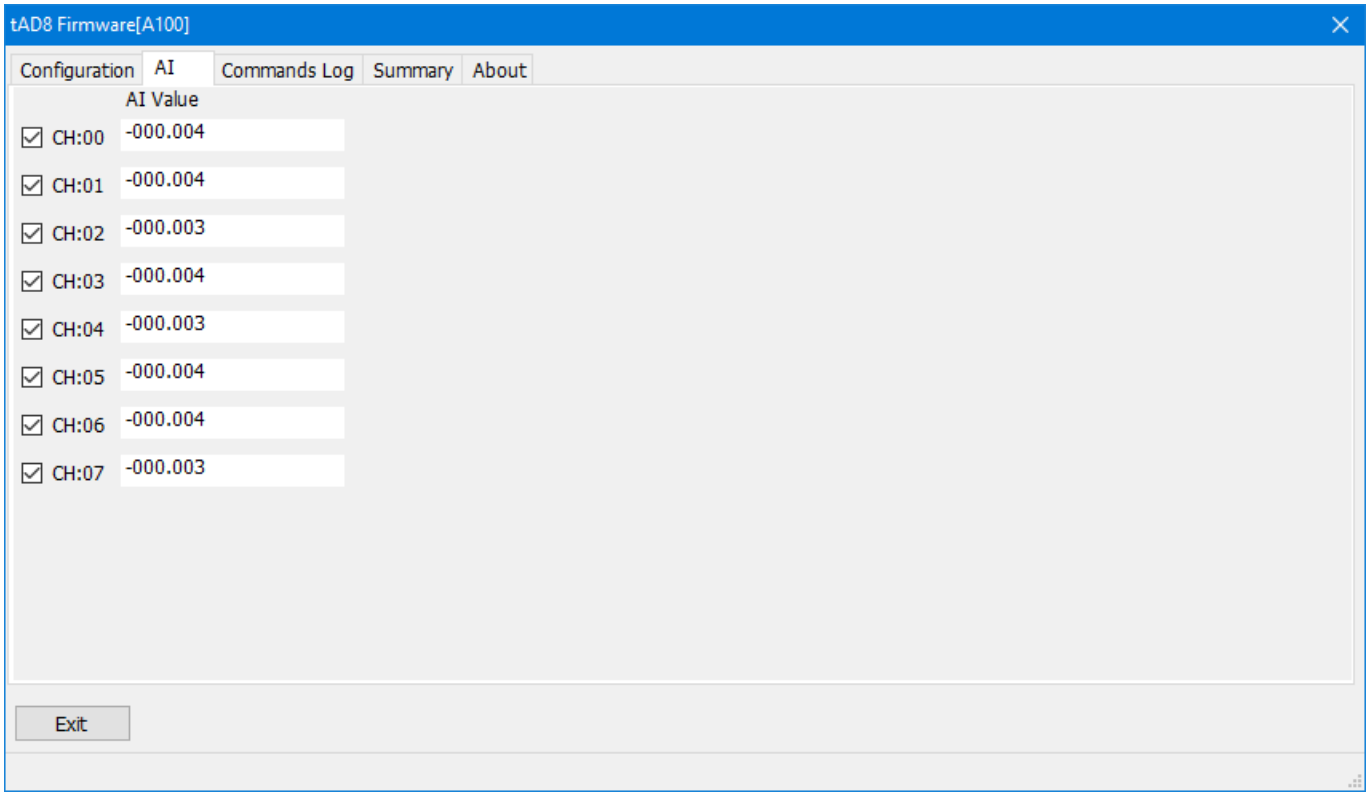


CH: xx	Check the box to detect the current value of the channel
Type Code	Specify a type code with data range tM-AD2: 0 to 500 mV, 0 to 1 V, 0 to 2.5 V, 0 to 5 V, 0 to 10 V, 0 to 20 mA, 4 to 20 mA (Reference: B.1 Data Ranges)
Set all channels as CH0	Configure all channels to the same type code as CH: 00

3.3.2 AI Page02 for tM-AD5, AD8, AD5C, and AD8C

The “AI” page can be used to detect data for each analog input channel.

Note: tM-AD5 and tM-AD5C equip five analog input channels, i.e., CH: 00 to CH: 04.
tM-AD8 and tM-AD8C equip eight analog input channels, i.e., CH: 00 to CH: 07.

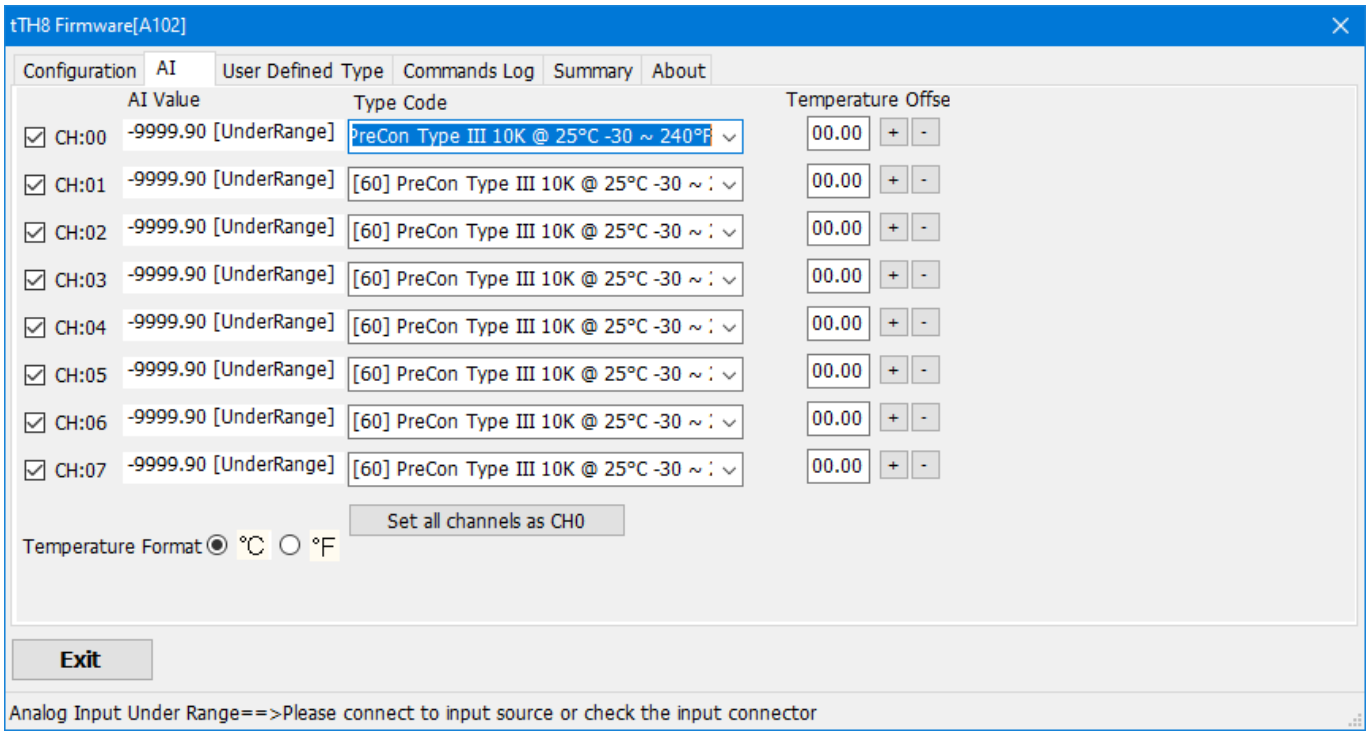


CH: xx	Check the box to detect the current value of the channel **DCON, Modbus RTU/ASCII** **Note:** **Only applicable for the measurement ranges of 0-20 mA and 4-20 mA.**

Protocol	Analog Format	Under Range
DCON	Engineering	-9999.9
Modbus		-32768
DCON, Modbus	2’s Complement	8000
DCON	Percent	-999.99

3.3.3 AI Page03 for tM-TH8

The “AI” page can be used to detect data and configure the type code and temperature settings for each analog input channel.



CH: xx	Check the box to detect the current value of the channel **DCON, Modbus RTU/ASCII** **Protocol** / **Analog Format** / **Under Range** / **Over Range** DCON / Engineering / -9999.9 / +9999.9 Modbus / Engineering / -32768 / 32767 DCON, Modbus / 2’s Complement / 8000 / 7FFF DCON / Percent / -999.99 / +999.99
Type Code	Specify a type code with temperature range for the thermistor (Reference: B.8 Data Ranges)
Set all channels as CH0	Configure all channels to the same type code as CH: 00
Temperature Offset	Specify an offset to adjust the temperature reading
Temperature Format	Specify the temperature scale It can be Celsius (℃) or Fahrenheit (℉)

3.4 Other I/O Pages

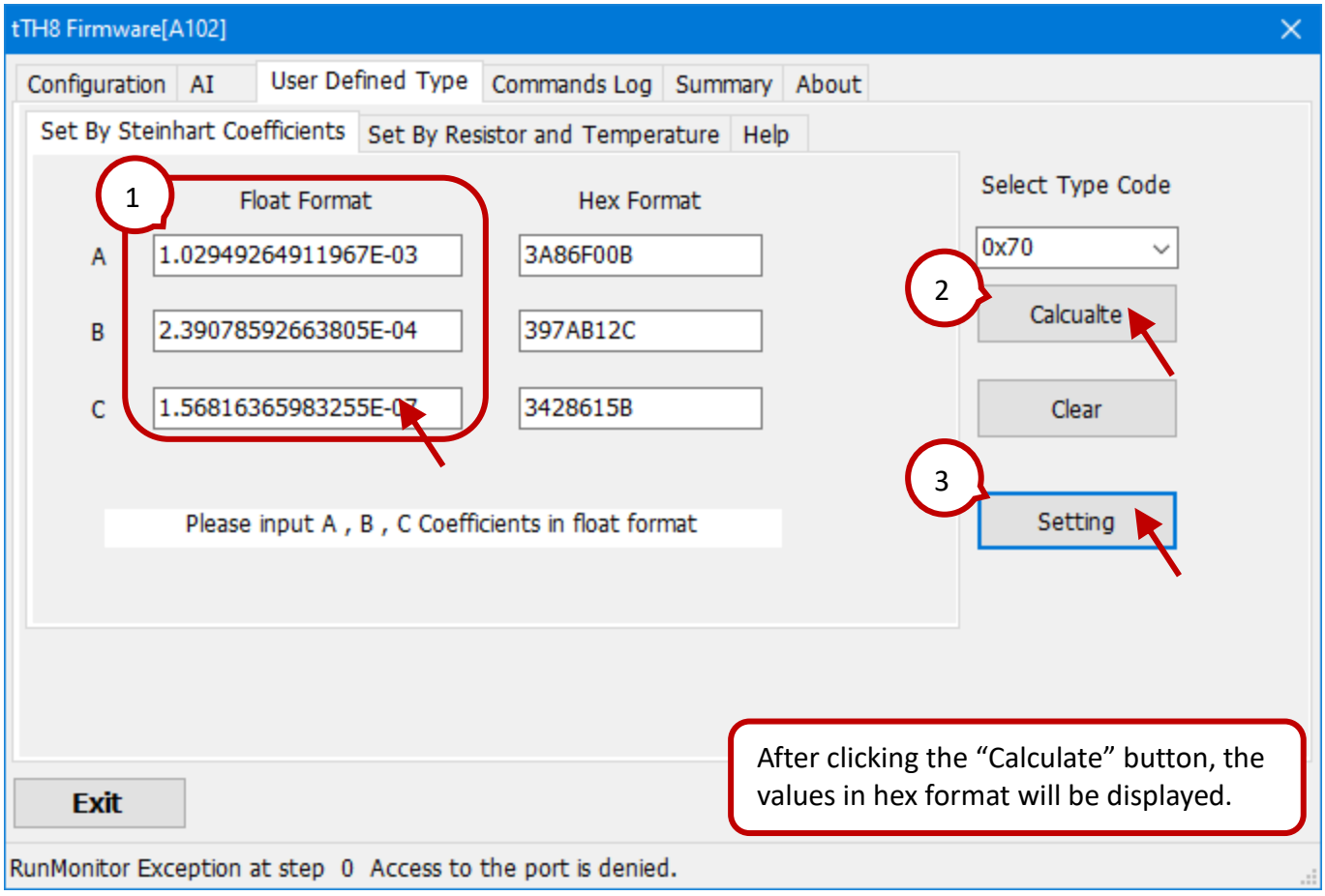
3.4.1 Configuration Page - User Defined Type

"User Defined Type" is a customized configuration page that is used for tM-TH8, a thermistor input module. The following example shows how to set the type code for a custom thermistor.

1. To get Steinhart-Hart coefficients from the thermistor vendor. On the “Set By Steinhart Coefficients” page, enter A, B, and C coefficients in **Float Format**, click the “Calculate” button and then click the “Setting” button.

   For example, there is a thermistor with the part number 'YSI H Mix 10000', and three Steinhart-Hart coefficients are listed in the table below.

Coefficient	Float Format	Hex Format
A	1.02949264911967E-03	3A86F00B
B	2.39078592663805E-04	397AB12C
C	1.56816365983255E-07	3428615B



2. Otherwise, you can also get a Resistance-Temperature table from the thermistor vendor. On the “Set By Resistor and Temperature” page, enter three sets of Resistance-Temperature values (Ohms, °C) and click the “Calculate” button to calculate three Steinhart-Hart coefficients.

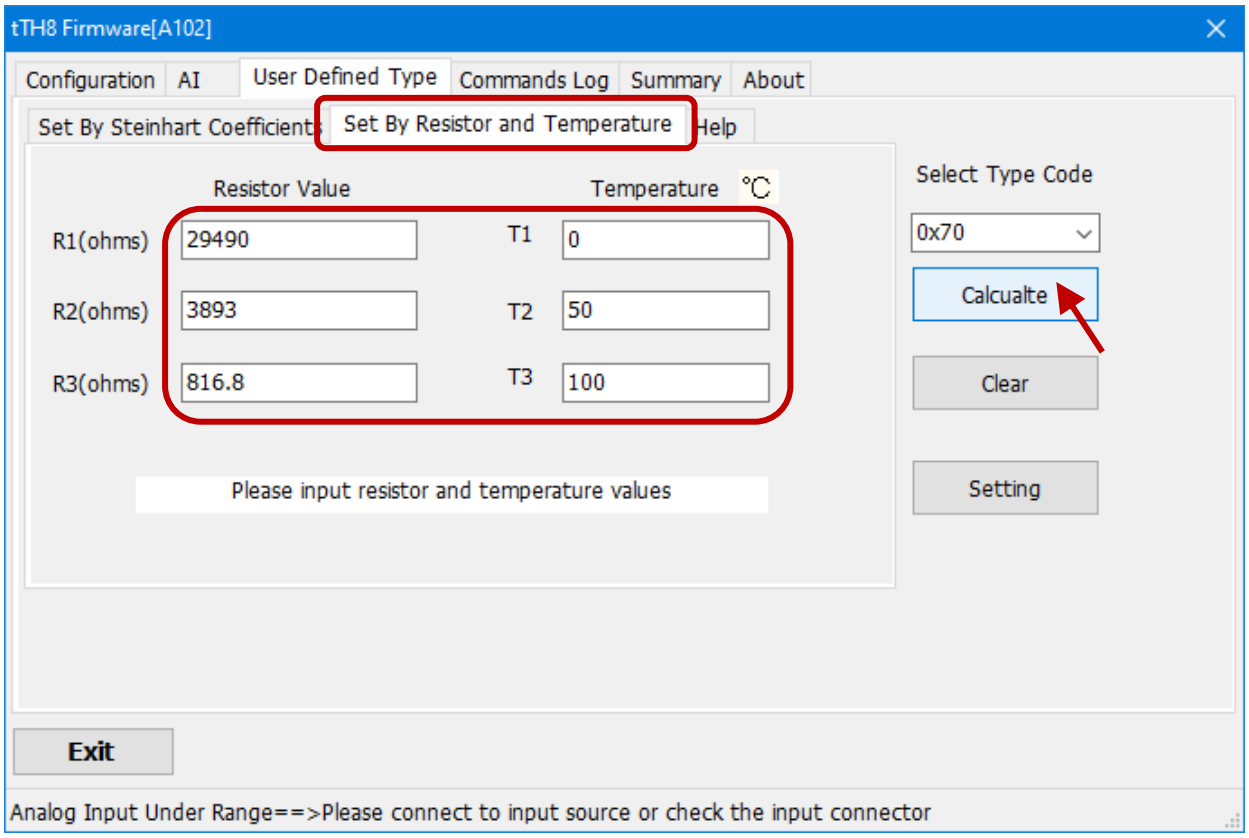
For accurate results, it is recommended to configure (Rx, Tx) by using the following rules:

(1) -40°C ≦ **T1, T2, T3** ≦ 150°C (2) |**T2 - T1**| ≦ 50°C (3) |**T3 - T2**| ≦ 50°C

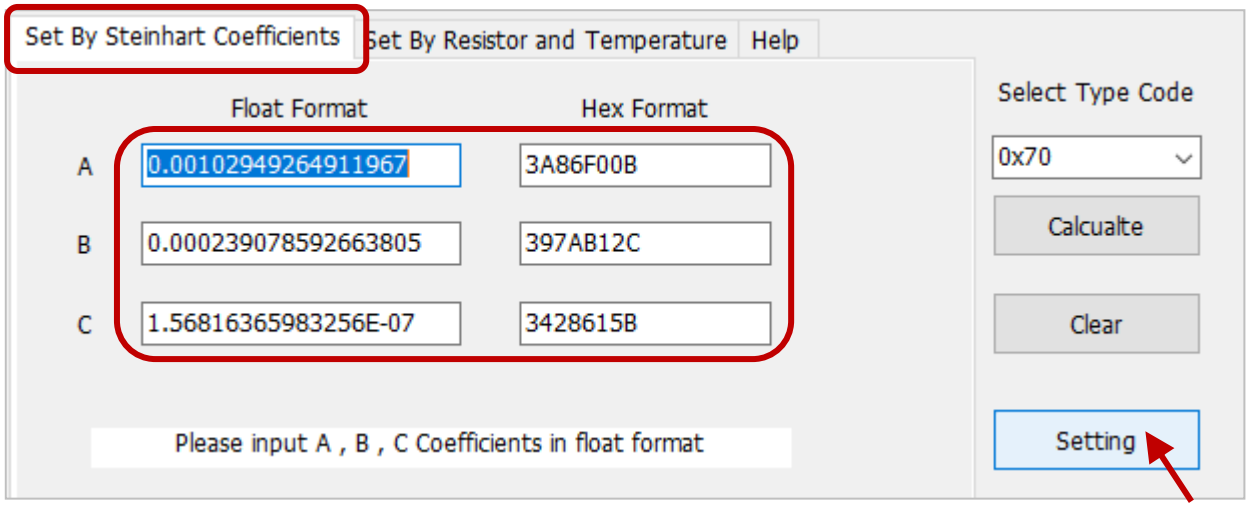
For example,

Resistance (Ω)	Temperature (°C)
R1: 29490	T1: 0
R2: 3893	T2: 50
R3: 816.8	T3: 100

Note: If the resistance is greater than 204700 ohms, it will be treated as under range.

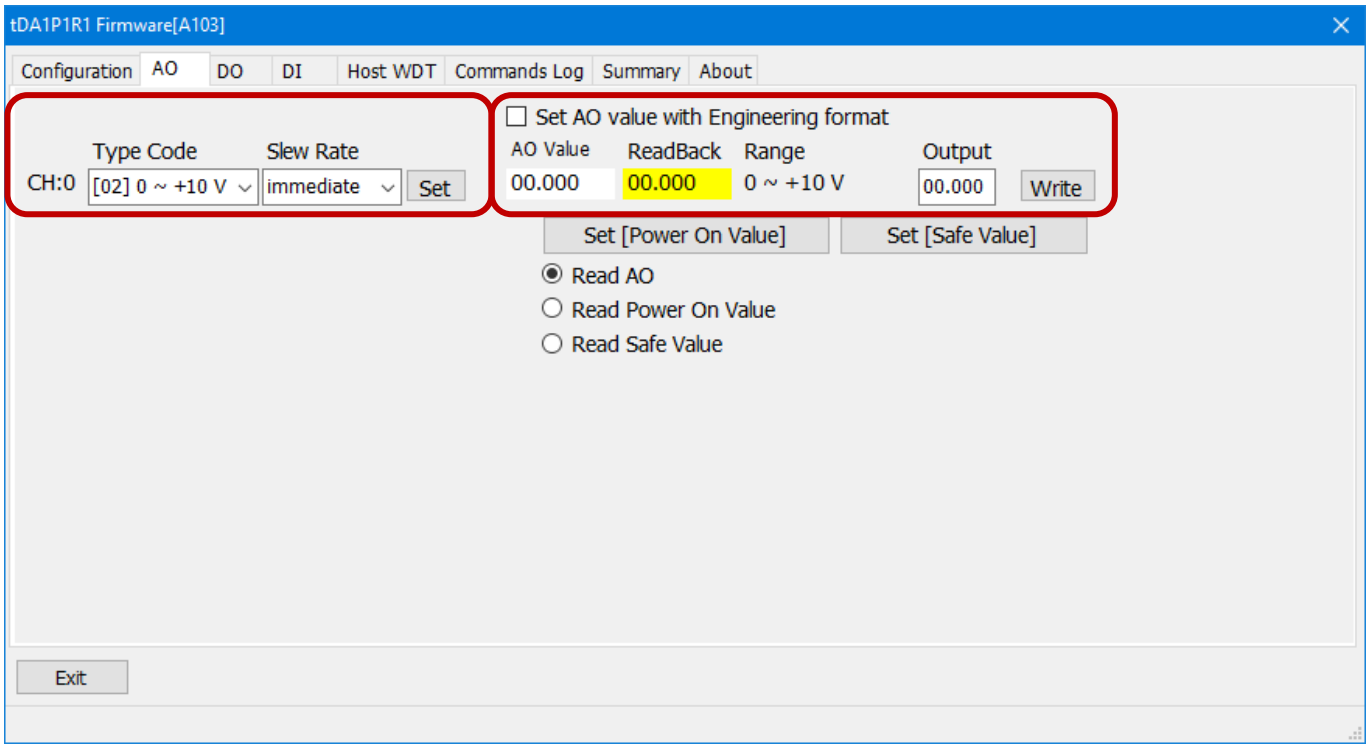


3. After clicking the “Calculate” button, it will automatically switch to the “Set By Steinhart Coefficients” page and show the results. Finally, click the “Setting” button to apply the settings.



3.4.2 Configuration Page - AO

tM-DA1P1R1 equips one analog output channel.



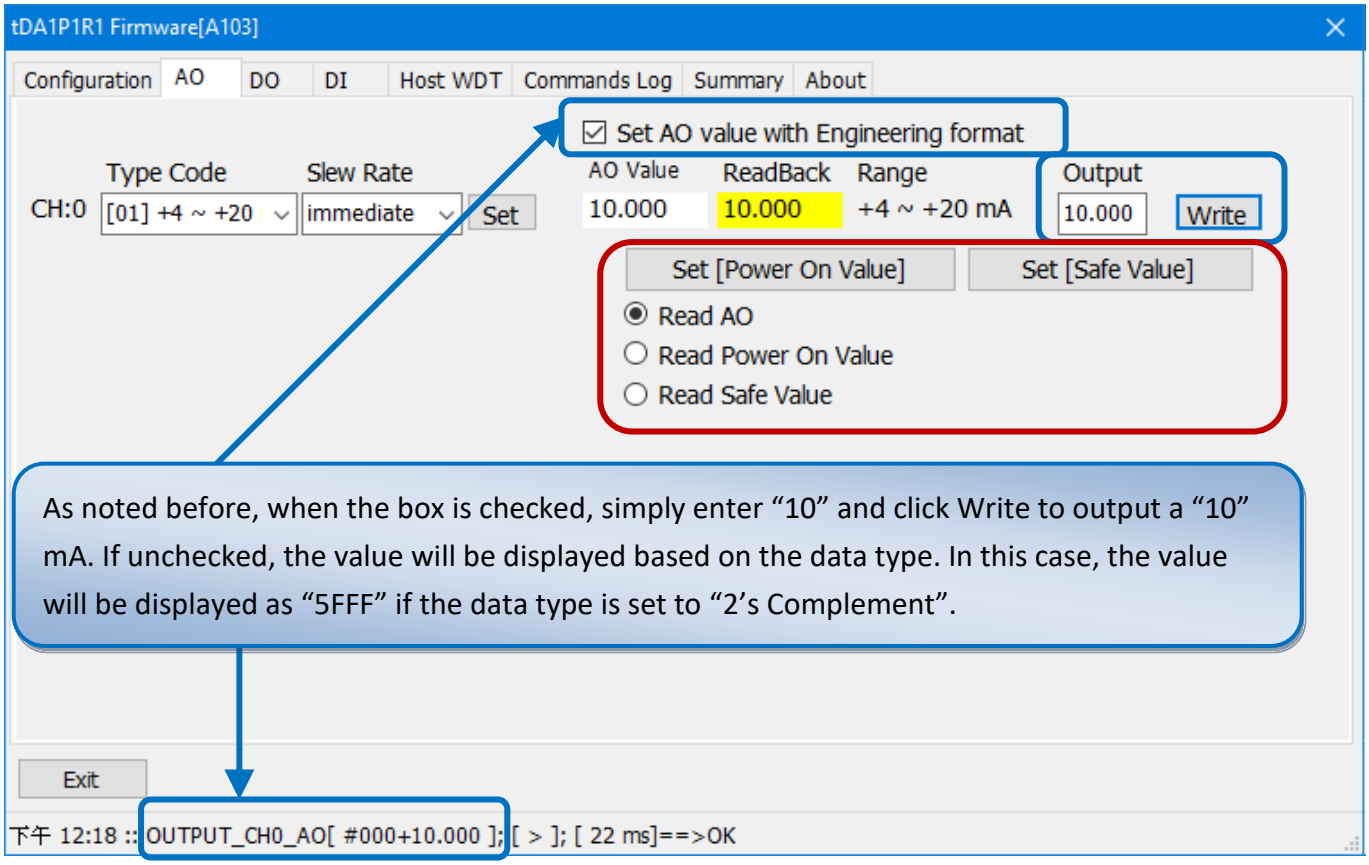
Type Code	Specify a type code with data range (tM-DA1P1R1: 0 to 10 V, 0 to 20 mA, 4 to 20 mA, reference: B.7 Data Ranges)
Slew Rate	Specify the change of voltage or current per second (V/sec, mA/sec) (Reference: A.4 Slew Rate Control)
Set	Click this button to apply the Type Code or Slew Rate settings

☑ **Set AO value with the Engineering format:**

1) **Check the box** to display or set the AO Value and the ReadBack value with the Engineering format.
2) **Unchecked**, to display value with the data format setting (reference: Section 3.2.4 Configuration page - Analog Format)

For example, the Type Code is set to "+4 to +20mA" and Analog Format is set to "**2's Complement**". To output 10 mA, enter "10" in the Output field, and click the Write button. If the box is checked, the value will display as "10", otherwise, the value will display as "**5FFF**".

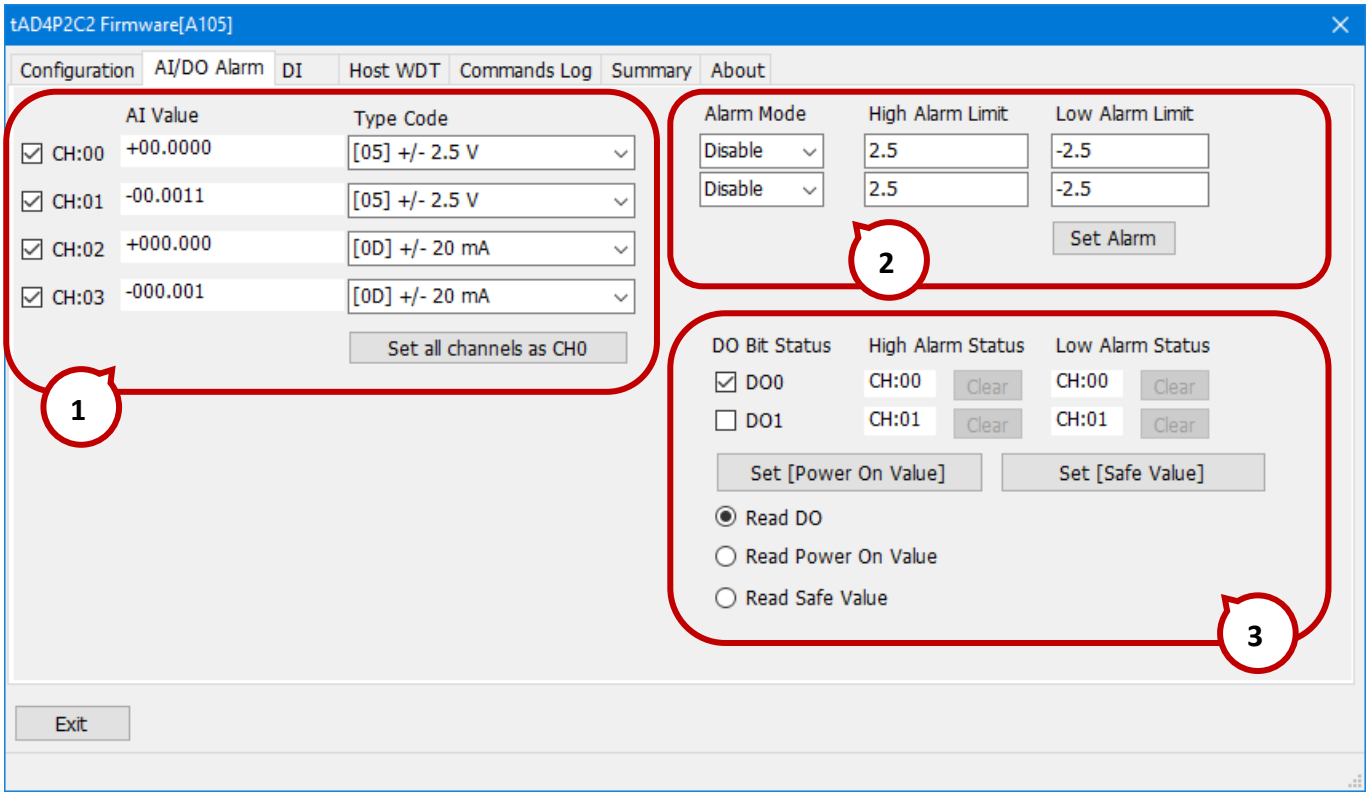
AO Value	Display the title and value according to the selected item such as Read AO/Power-on value/Safe value
ReadBack	
Note: The Readback value varies according to the Slew Rate setting so that may be different from the configured AO value. Reference: A.5 AO Read-back Value	
Range	Display the data range according to The type code setting
Output / Write	Specify an output value, and then click "Write" to apply the settings



Set to Power-on Value	Enter a value in the output field, and click this button to apply the settings Reference: A.2 Dual Watchdog
Set to Safe Value	
Read AO	Choose the radio button to read the AO value
Read Power-on Value	Choose the radio button to read the current Power-on value or Safe value, at which point the values cannot be set.
Read Safe Value	

3.4.3 Configuration Page - AI/DO Alarm

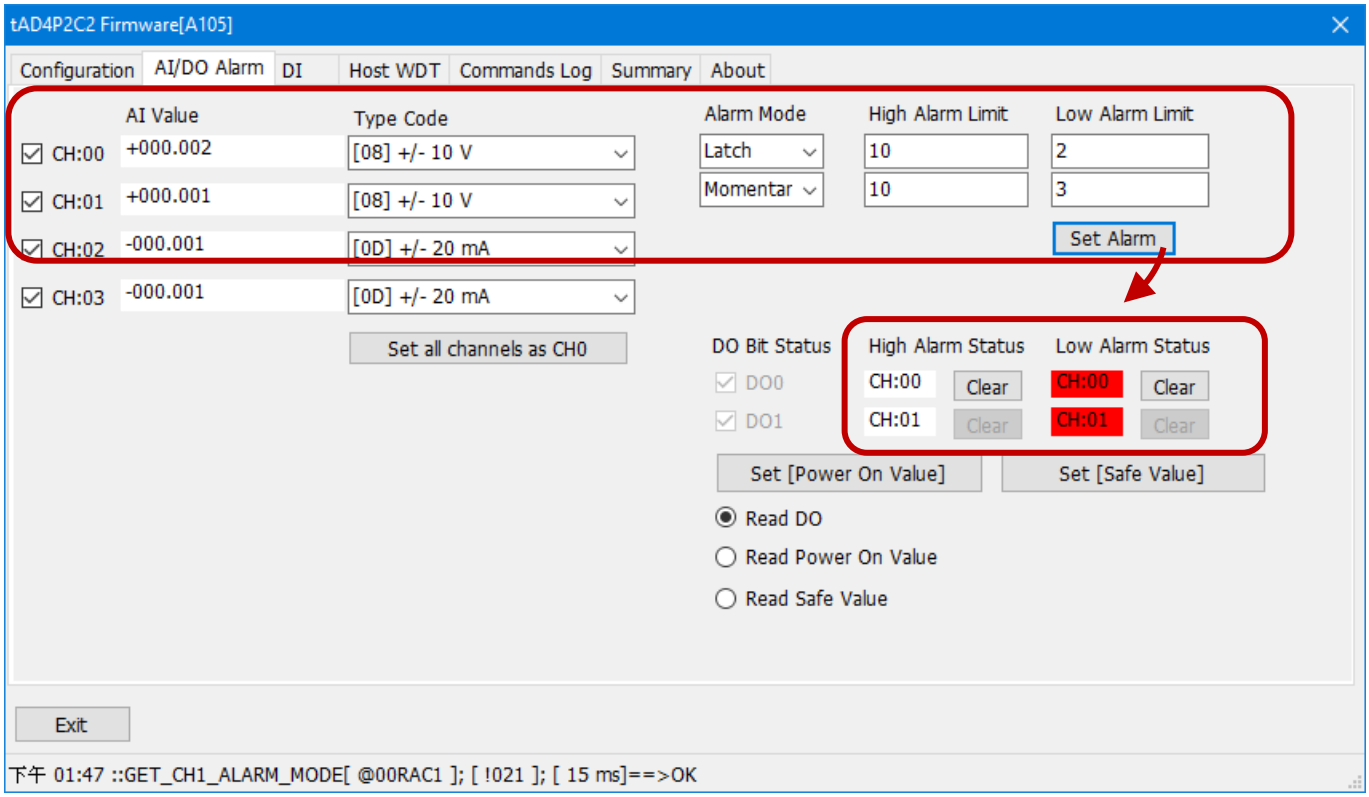
tM-AD4P2C2 equips four analog input channels and two relay output channels. Both the Alarm Mode and High/Low Alarm Limit setting are used to configure a warning light or an alarm device that be connected to a relay output channel.



CH: xx	Check the box to detect the current value of the analog input channel
Type Code	Specify a type code with data range tM-AD4P2C2: +/- 1 V, +/- 2.5 V, +/- 5 V, +/- 10 V, +/- 20 mA, 0 to 20 mA, 4 to 20 mA (Reference: B.6 Data Ranges)
Set all channels as CH0	Configure all channels to the same type code as CH: 00

Only when Alarm Mode = Disable, you can configure the following items:

DOx	Check the box to set the DO value, Power-on value, or Safe value
Set to Power-on Value	After selecting the DOx box, click this button to apply the settings (Reference: A.2 Dual Watchdog)
Set to Safe Value	
Read DO	Choose the radio button to read the DO status
Read Power-on Value	Choose the radio button to read the current Power-on value or Safe value, at which point the values cannot be set.
Read Safe Value	

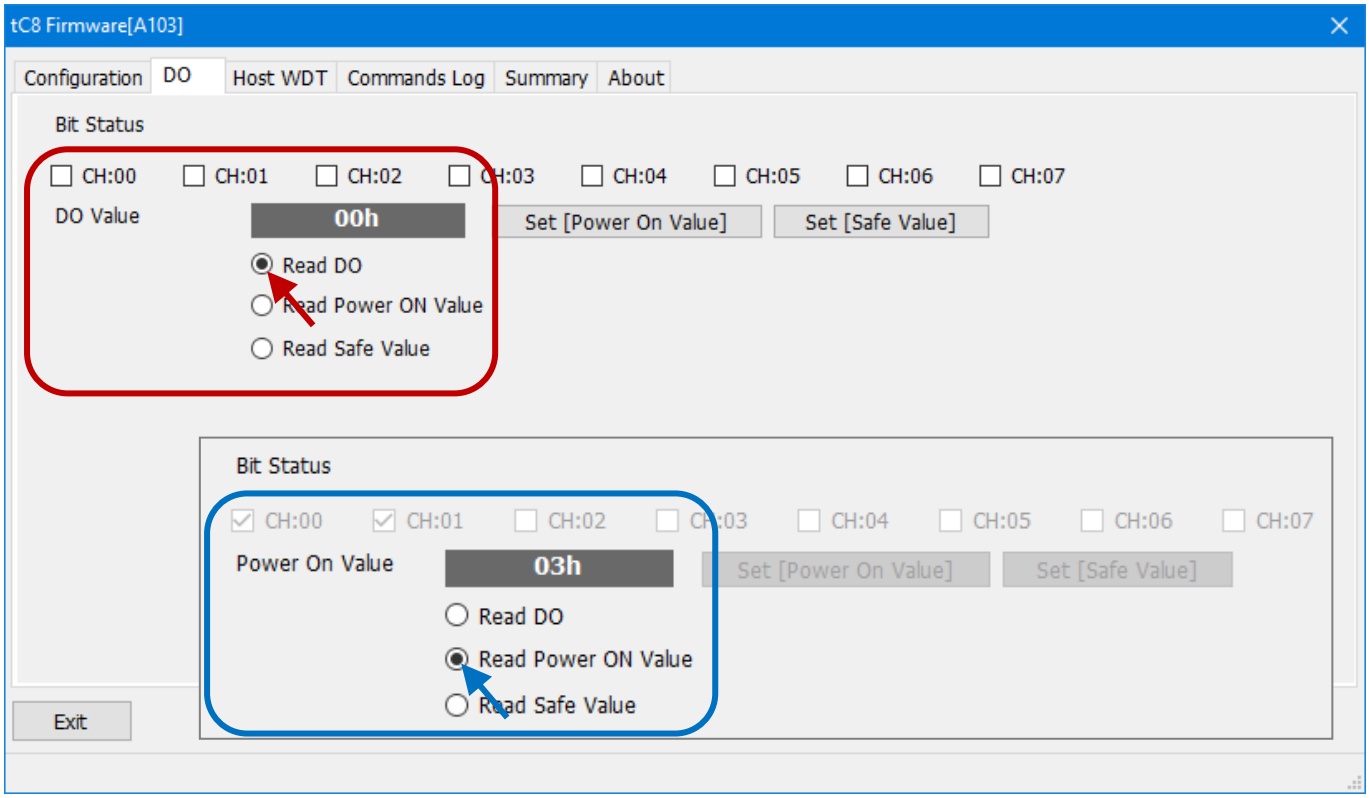


Alarm Mode	Specify the alarm mode, it can be Disable: Disabled Momentary: An alarm occurred when the value is out of the limit Latch: An alarm occurred when the value is out of the limit. The status of the alarm can be cleared.
High Alarm Limit	After choosing a type code, set the upper limit of the alarm
Low Alarm Limit	After choosing a type code, set the lower limit of alarm
Set Alarm	Click the “Set Alarm” button to apply the settings for Alarm Mode, High Alarm Limit, and Low Alarm Limit
(High/Low) Alarm Status	Used to display or clear the status of the alarm 1) CH: xx, if the value is out of the limits, the text-box will be displayed in red 2) Clear, when Alarm Mode is set to Latch, the button can be used to clear the status of the alarm by clicking it

3.4.4 Configuration Page - DO

The “DO” page can be used to set or display the DO value, Power-on value, or Safe value.

Model	Description
tM-DA1P1R1	One Digital Output Channel
tM-P3R3, tM-PD3R3, tM-P3POR3	Three Relay Output Channels
tM-P4A4, tM-P4C4	Four Digital Output Channels
tM-R5	Five Relay Output Channels
tM-C8	Eight Digital Output Channels



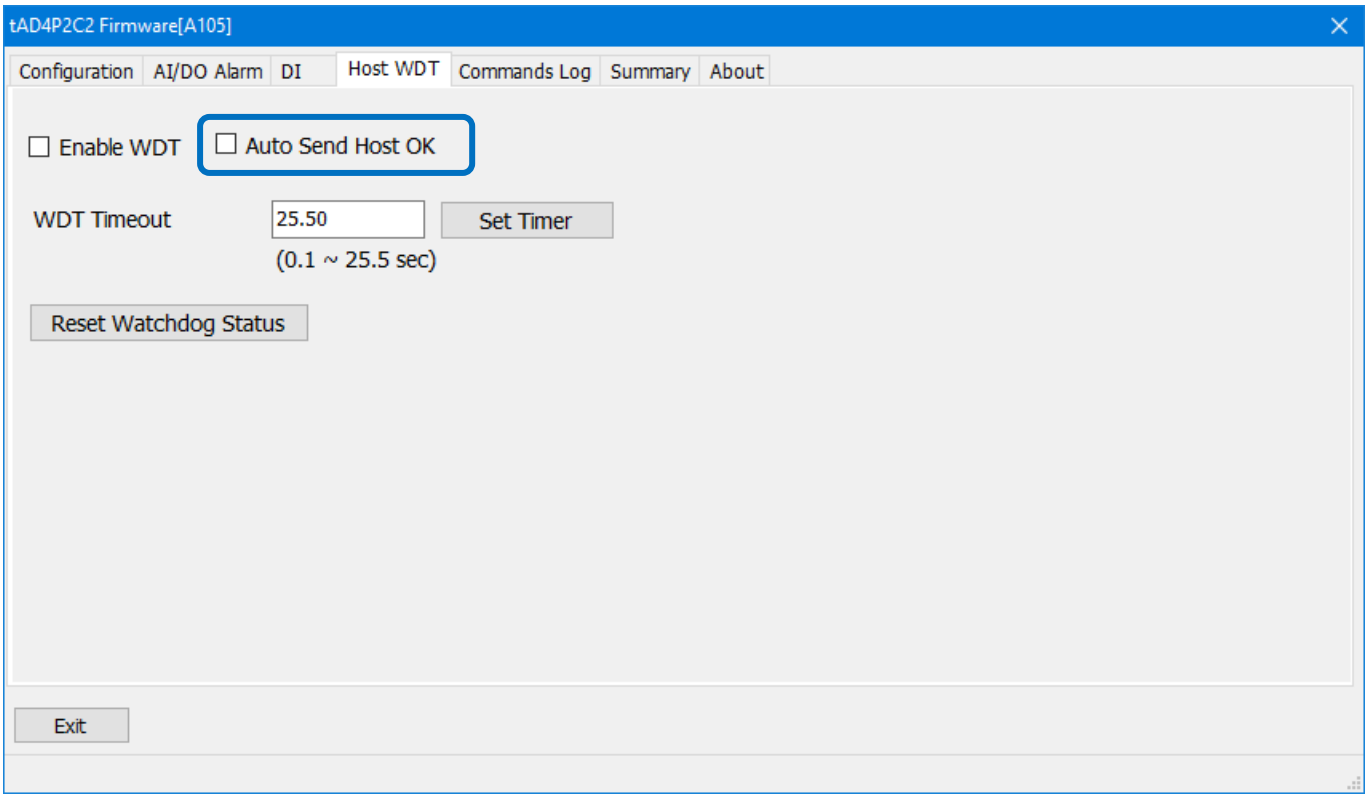
Bit Status (CH: xx)	Check the CH: xx box to specify the value
DO Value	After selecting the CH: xx boxes, a hexadecimal value will be displayed
Set to Power-on Value	After selecting the CH: xx boxes, click the button to apply the settings (Reference: A.2 Dual Watchdog)
Set to Safe Value	
Read DO	Choose the radio button to read the DO status
Read Power-on Value	Choose the radio button to read the current Power-on value or Safe value (While the radio button is selected, these two values cannot be set.)
Read Safe Value	

3.4.5 Configuration Page - Host WDT

The “Host WDT” page can be used to enable the host watchdog and reset the status of the host watchdog timeout. (Reference: Appendix A.2 Host Watchdog)

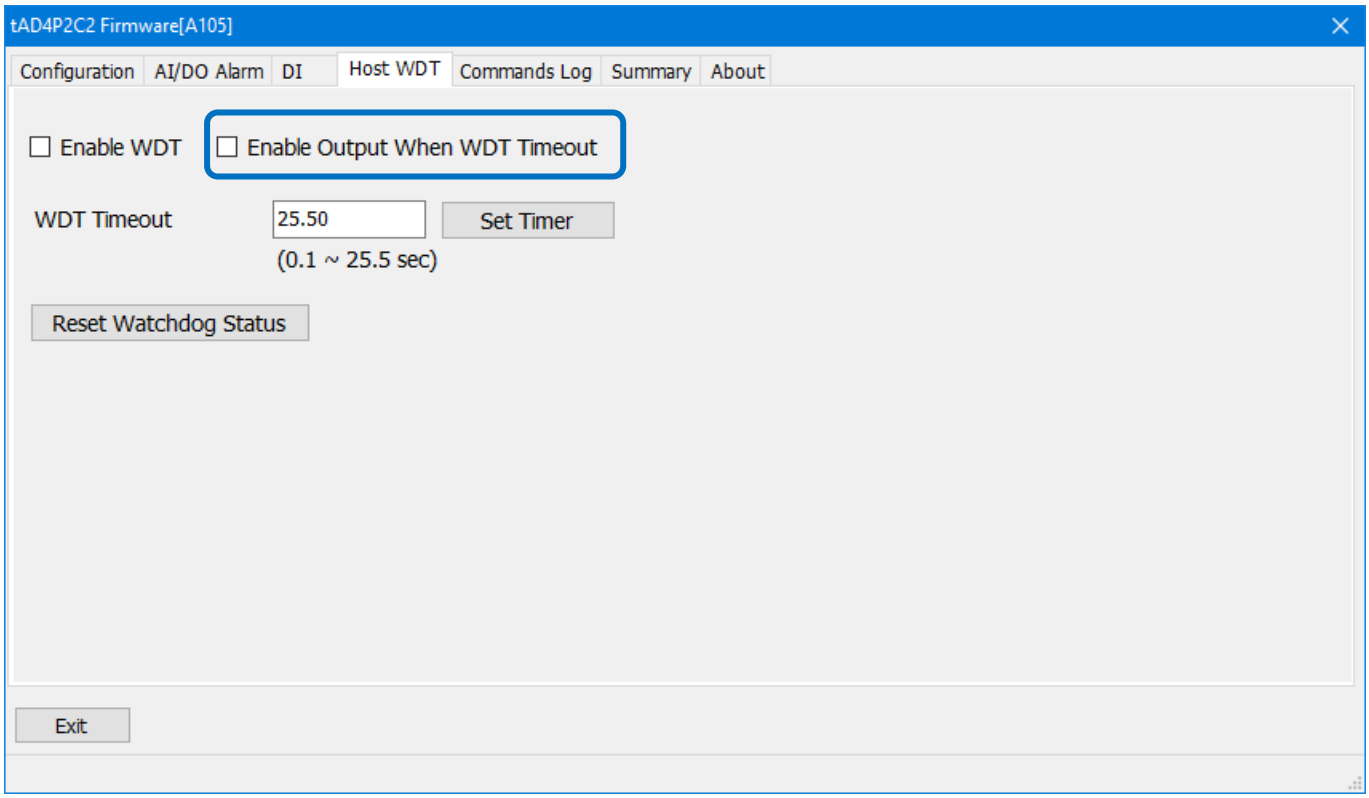
Model	Description
tM-DA1P1R1	One Analog Output and One Digital Output Channels
tM-AD4P2C2	Two Digital Output Channels
tM-P3R3, tM-PD3R3, tM-P3POR3	Three Relay Output Channels
tM-P4A4, tM-P4C4	Four Digital Output Channels
tM-R5	Five Relay Output Channels
tM-C8	Eight Digital Output Channels

1) When using the **DCON** protocol, the Host WDT page will be displayed as follows.
(Reference: Section 3.2.1 Common Settings)



Enable WDT	Check the box to enable the watchdog
Auto Send Host OK	Check the box to automatically notify that the host works normally Note that the function is for testing and the timer is disabled
WDT Timeout / Set Timer	Specify a watchdog timeout value and click the “Set Timer” button to apply the setting
Reset Watchdog Status	After the timeout, it needs to clear the Timeout status to write an AO/DO value.

2) When using the **Modbus RTU (or Modbus ASCII)** protocol, the Host WDT page will be displayed as follows. (Reference: Section 3.2.1 Common Settings)



Enable WDT	Check the box to enable the watchdog
Enable Output When WDT Timeout	Check the box to allow modifying an AO/DO value, even if the Watchdog timeout has occurred
WDT Timeout / Set Timer	Specify the duration of the Watchdog timeout, and click the “Set Timer” button to apply the setting
Reset Watchdog Status	Click the button to clear the Timeout status

For example:

1. Check the Enable WDT checkbox, set WDT Timeout as 5 seconds, and click the Set Timer button. After 5 seconds, a warning message “Watchdog Timeout” will be displayed on the status bar. The output value will be set as the predefined Safe Value, which cannot be changed.

   Note: When using the Modbus protocol, you can check the “Enable Output When WDT Timeout” box to allow the change of an AO/DO value, even if the timeout occurred.

2. Click the “Reset Watchdog Status” button to clear the Timeout status, and the predefined Power-on value will be loaded after a reboot. Otherwise, the predefined Save value will be loaded.

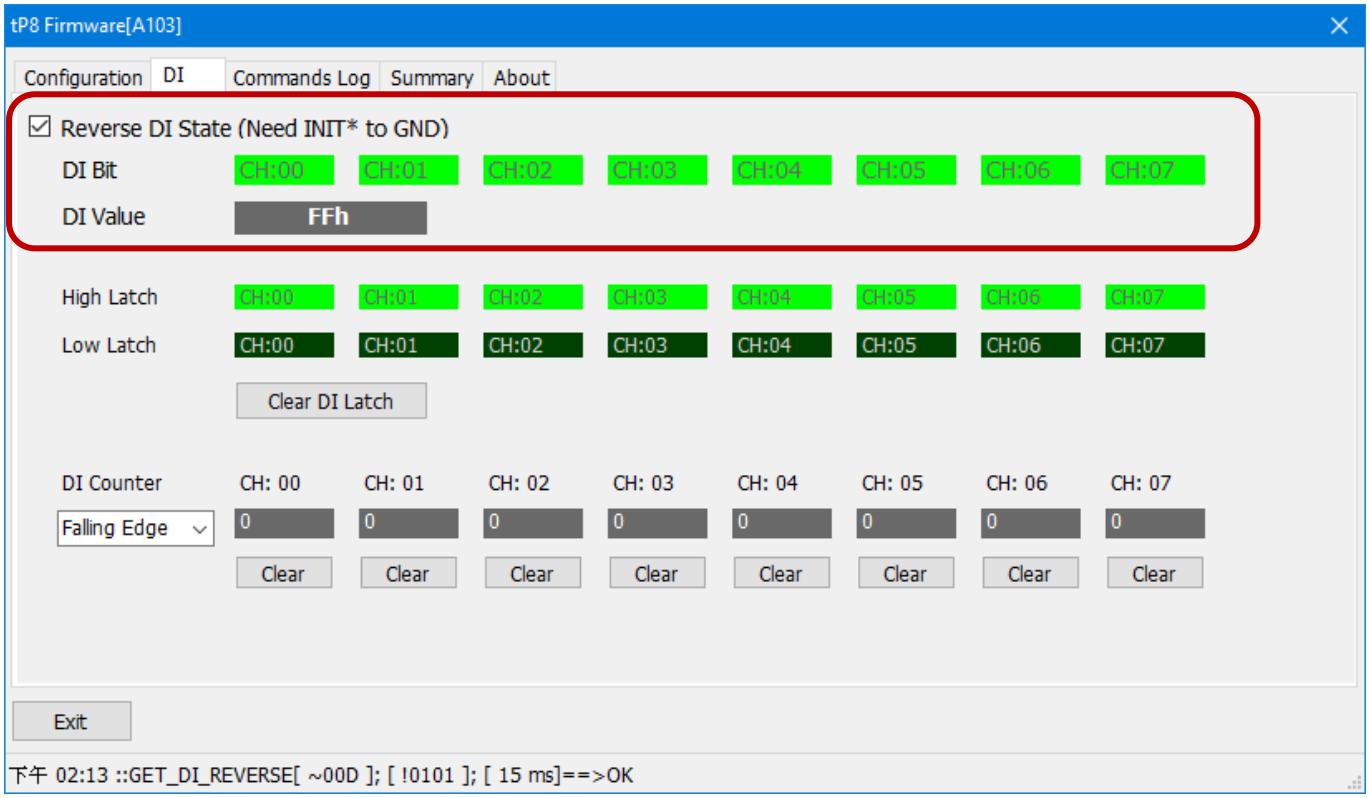
   Note: When using the DCON protocol, you can check the "Auto Send Host OK” box to automatically send a signal which indicates the Host works normally. Also, uncheck the box and click the “Set Timer” box again to start the watchdog timer.

3.4.6 Configuration Page - DI

The DI page is used to display or clear the status of DI, DI Latch, or DI Counter.

Model	Description
tM-DA1P1R1	One digital input channel
tM-AD4P2C2	Two digital input channels
tM-P3R3, tM-PD3R3, tM-P3POR3	Three digital input channels
tM-P4A4, tM-P4C4	Four digital input channels
tM-P8, tM-PDW8	Eight digital input channels

DI Page

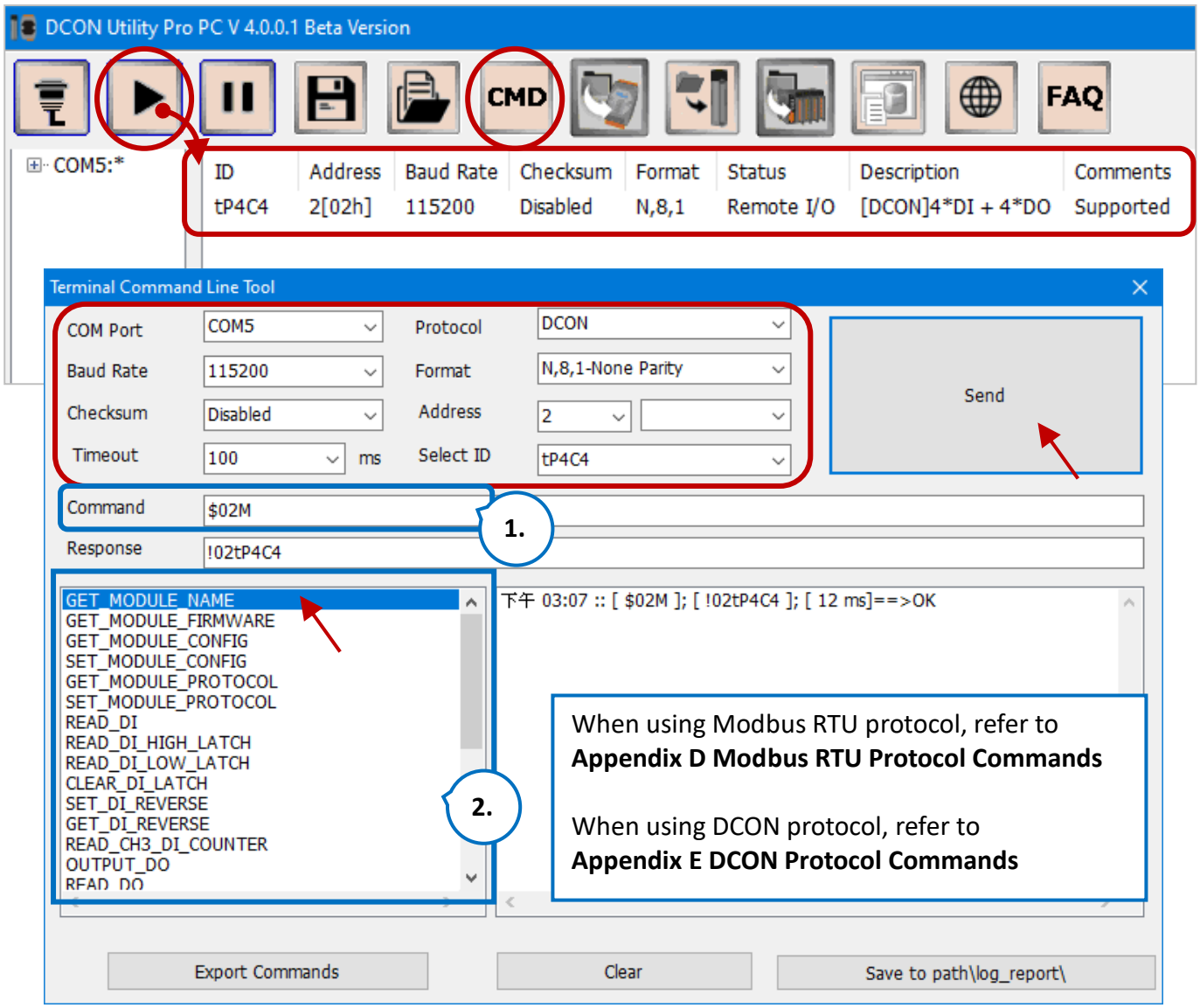


Reverse DI Status		Check the box to reverse the status of all DI channels
DI Bit Status		Display the status for each DI channel (ON or OFF)
DI Value		Using a hexadecimal value to indicates the status of DI channels
DI Latch	High Latch	Display the status of active-high for each DI channel
	Low Latch	Display the status of active-low for each DI channel
	Clear DI Latch	Click the button to clear the latched status for all DI channels (Reference: Appendix A.6 Advanced DI Features)
DI Counter	Falling/Rising Edge	The counter value will be increased by 1 when received a high-to-low or a low- to-high signal
	Clear	Click the button to clear the counting value of the DI channel

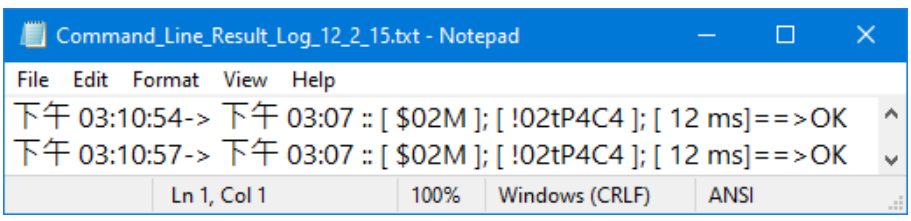
3.5 Command Line Tool

"Command Line" tool can be used to test and debug tM series modules by using the Modbus RTU protocol and DCON protocol commands.

1. Search the tM module and then click the **Command Line** button to open the configuration page.
2. In the "Terminal Command Line Tool" window, the module settings will be filled in automatically. Either enter a command in the **Command** field (1) or use the preset commands (2) and click the Send button.



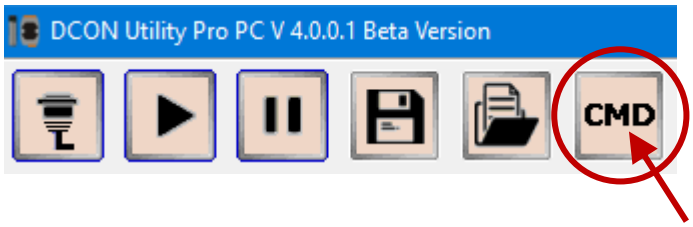
3. Click the "Save to \logger_report\" button to save results to a file in the "log_report" folder under the installation folder of DCON Utility Pro. The file name is "Command_Line_Result_Log_**mm_dd_xx**.txt".



3.6 Calibration

Normally, the module has been calibrated before shipping. Also, users can use the following commands to calibrate the module by using Command Line in the DCON Utility Pro.

Warning:
It is not recommended that calibration be performed until the process is fully understood.

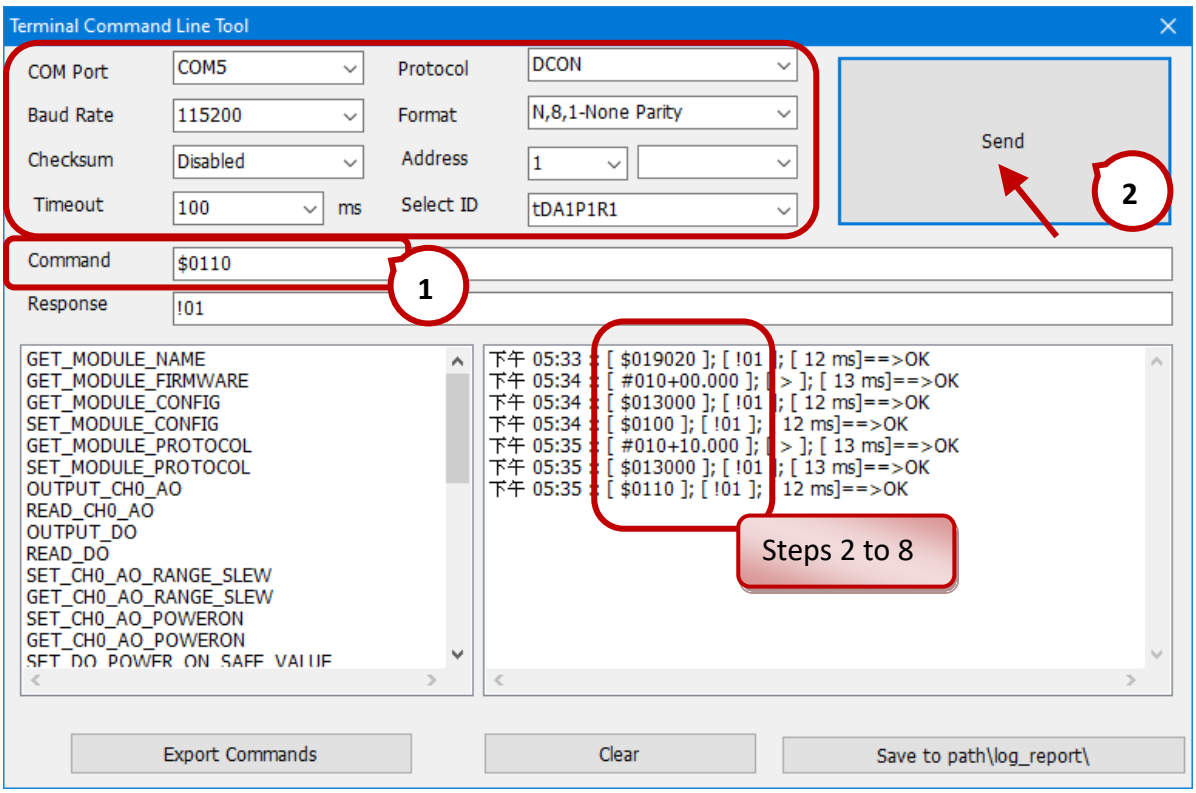


3.6.1 Calibrate AO Module

tM-DA1P1R1 provides a single analog output channel. Follow these steps to calibrate the module by using DCON Utility Pro.

1. Warm up the module for 30 minutes.
2. Set ($AA9NTS) / read ($AA2) the type code.
   E.g., **$019020**, means to set the type code of analog output channel **0** of module 0**1** as **2** (i.e., 0 to +10V), and the slew rate is set to **0** (Immediate).

Note: The module must be set to the DCON protocol mode before calibrating.



Calibration voltages or current used by the **tM-DA1P1R1**:

Type Code (T)	0	1	2	4
Zero Output	0mA	4mA	0V	0V
Span Output	+20mA	+20mA	+10V	+5V

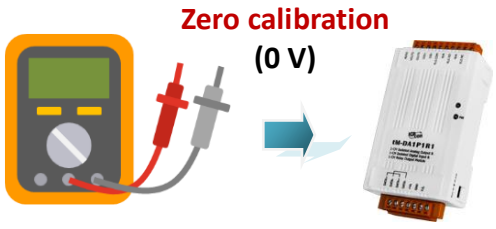
3. Send command to output **Zero calibration** voltage/current. (E.g., **0 V**)

   Format: **#AAN(Data)** = # [AA] Address [N] Channel No. [Data] +00.000 (V, mA)

   Command: **#010+00.000**

4. Adjust the value

   Repeat to send the trim command and observe the meter until the meter's reading is close to zero calibration voltage/current.



   Format: **$AA3NVV** = $ [AA] Address [3] Set analog output [N] Channel No.
   [VV] 00 to 5F means to increase 0 to 95 counts ;
   FF to A1 means to decrease 1 to 95 counts

   Command: **$01301F** (**1F** stands for add **31** * 2.44 mV or add **31** * 4.88 uA)

5. After adjusting the value, send the **Zero calibration** command. Note that the value will be stored in an EEPROM.

   Format: **$AA0N** = $ [AA] Address [0] Zero calibration [N] Channel No.

   Command: **$0100**

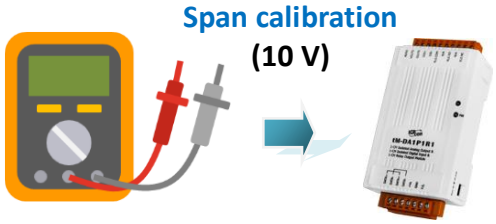
6. Send command to output **Span calibration** voltage/current. (E.g., 1**0 V**)

   Format: **#AAN(Data)** = # [AA] Address [N] Channel No. [Data] +10.000 (V, mA)

   Command: **#010+10.000**

7. Adjust the value

   Repeat to send the trim command and observe the meter until the meter's reading is close to span calibration current/voltage.



   Format: **$AA3NVV** = $ [AA] Address [3] Set analog output [N] Channel No.
   [VV] 00 to 5F means to increase 0 to 95 counts ;
   FF to A1 means to decrease 1 to 95 counts

   Command: **$01301F** (**1F** stands for add **31** * 2.44 mV or add **31** * 4.88 uA)

8. After adjusting the value, send the **Span calibration** command. Note that the value will be stored in an EEPROM.

   Format: **$AA1N** = $ [AA] Address [1] Span calibration [N] Channel No.

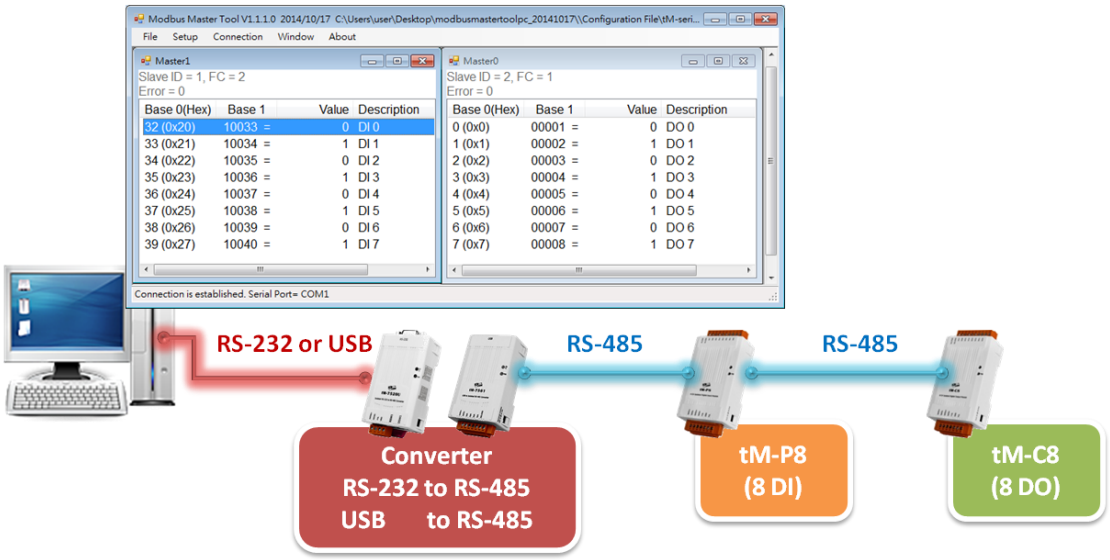
   Command: **$0110**

Chapter 4 Software Tool – Modbus Master Tool

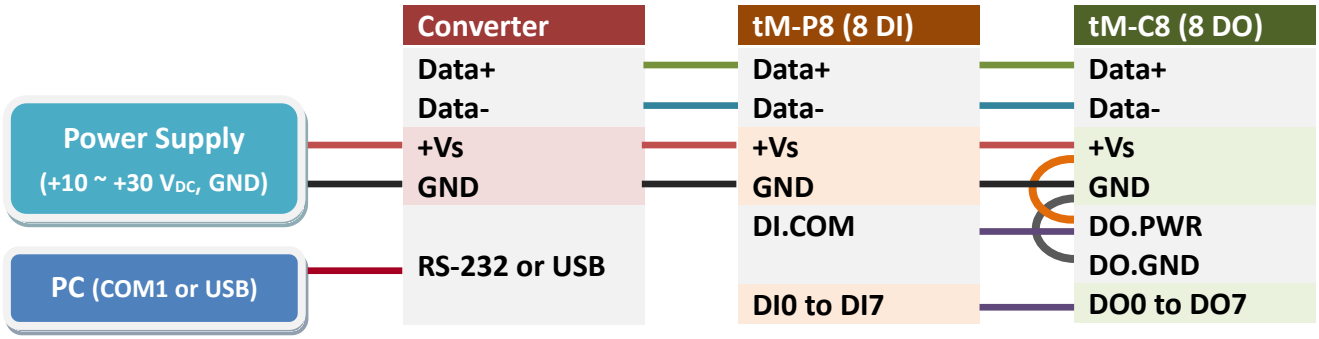
Modbus Master Tool supports Modbus RTU/ASCII protocol that can be used to simulate and test I/O modules on a PC (or PAC). First, download the software and the user manual on FTP.

Web: https://www.icpdas.com/en/product/guide+Software+Development__Tools+Modbus__Tool#674

Tool: https://www.icpdas.com/en/download/show.php?num=1026 (PC or WinCE)



Hardware Wiring for testing:



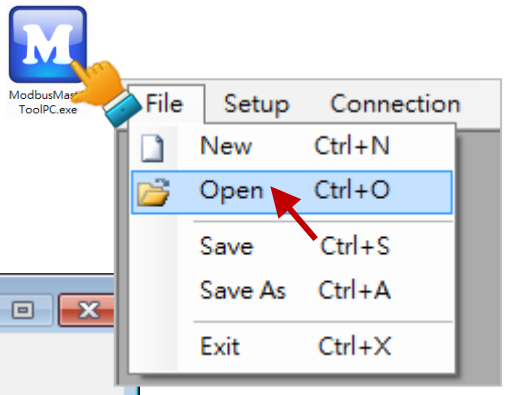
In this example, using the tM series DI and DO module. Follow the steps below to conduct the setting.

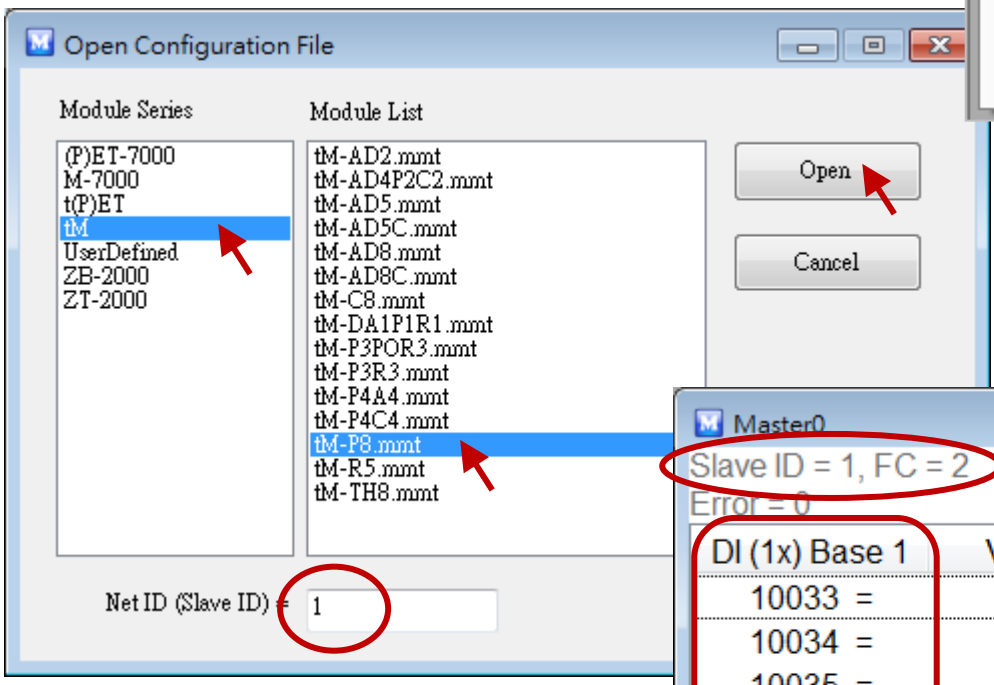
Model	Description	Slave ID	Baud Rate	Protocol	Modbus Address
tM-P8	8 DI	1	9600	Modbus RTU	10033 to 10040
tM-C8	8 DO	2			00001 to 00008

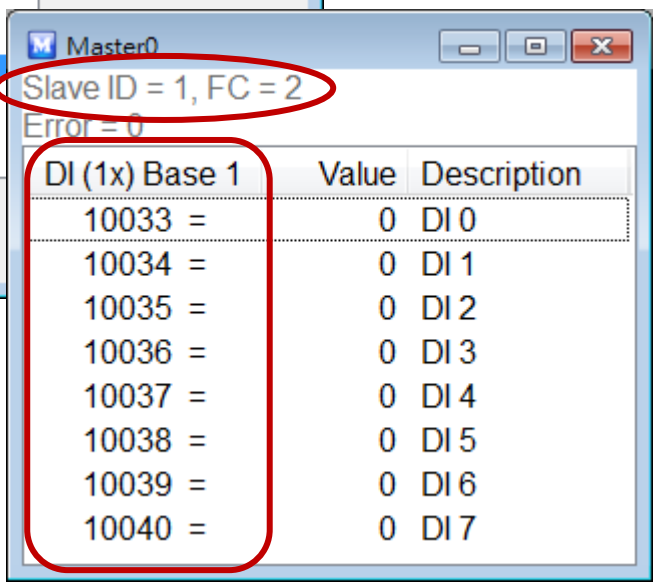
Note: Refer to Appendix C to look up the Modbus registers for tM series modules.

Step 1: Open the configuration file

After extracting the downloaded file, open the Modbus Master Tool. Select “Open” from the File menu and choose both the Module Series (e.g., tM) and the Module List (e.g., tM-P8), and then enter the Slave ID (e.g., 1), also click the “Open” button to open the configuration file.





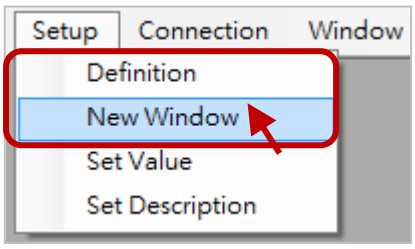


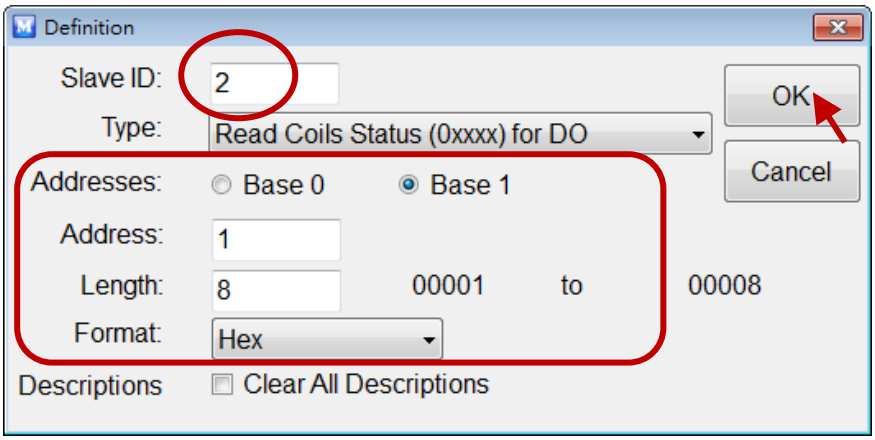
Model	Slave ID	Modbus Address
tM-P8 (8 DI)	1	10033 to 10040

(Base 1)

Step 2: Add a window and configure the DO

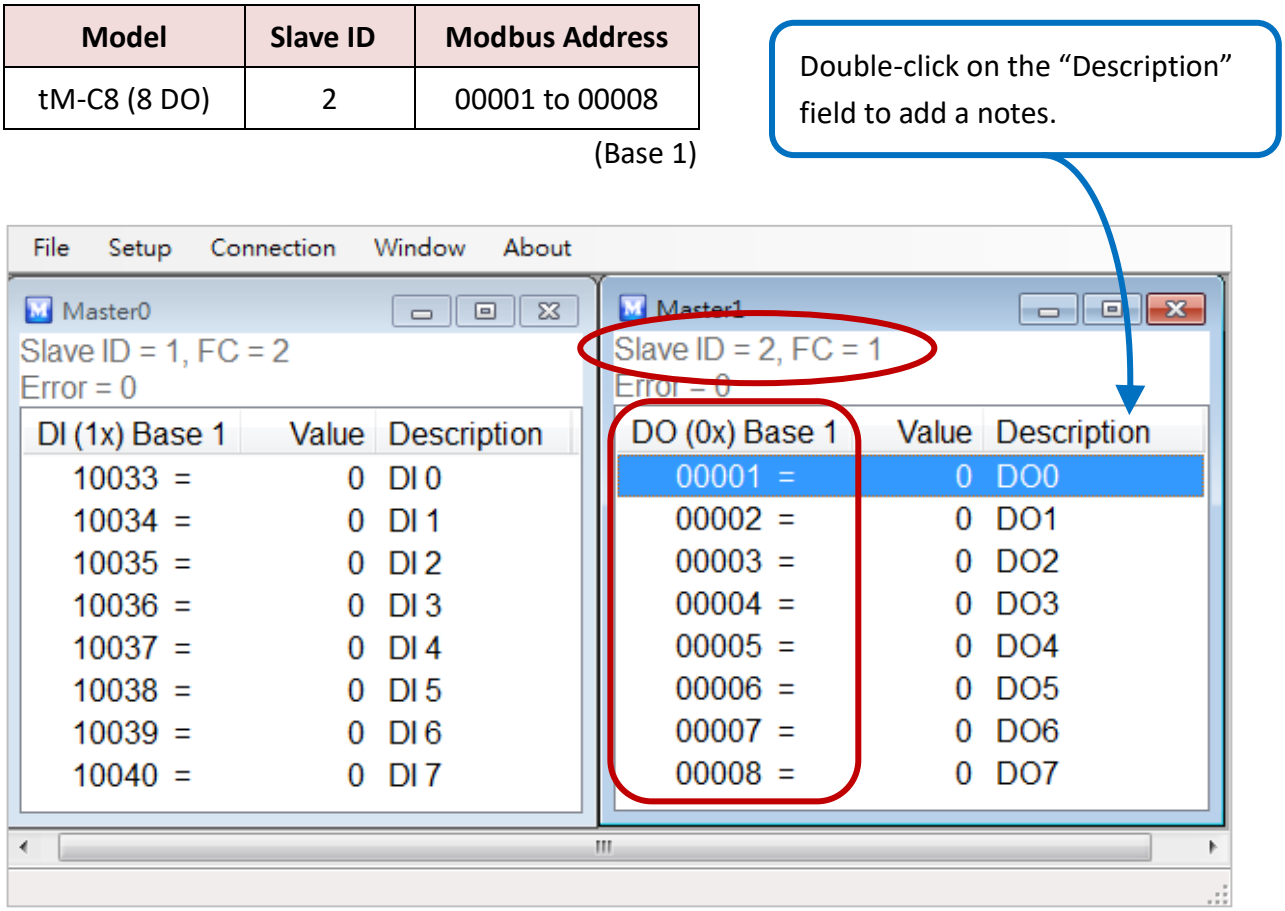
Select “New Window” from the Setup menu to add a window, and select “Definition” to configure the DO settings, and then click the OK button.





Model	Slave ID	Modbus Address
tM-C8 (8 DO)	2	00001 to 00008

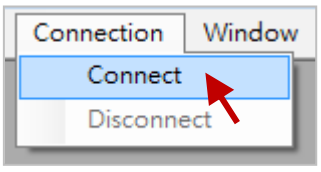
(Base 1)

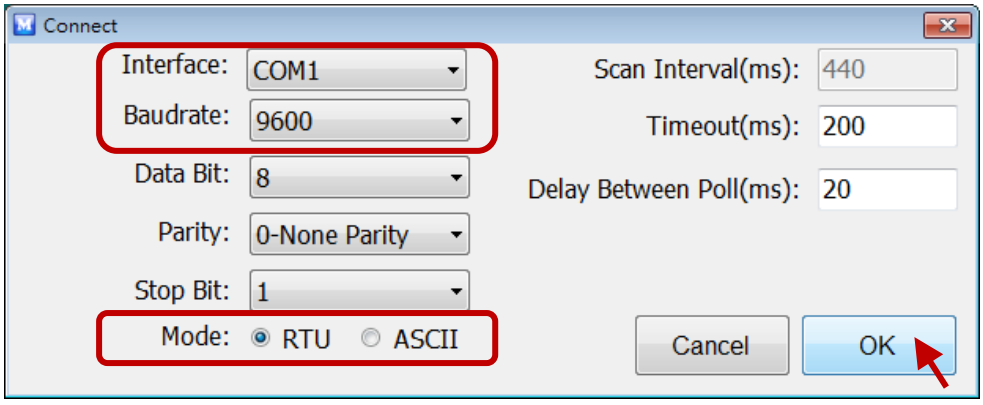


Step 3: Create a Serial Port connection

In this case, the tM module communicates with the COM1 of a PC through the Modbus RTU protocol. Select “Connect” from the “Connection” menu, and configure all parameters as illustrated in the figure below, and then click the “OK” button to establish the connection.

Note: The communication parameters of the tM module must be predefined by using the DCON Utility Pro, refer to Chapter 3 for more information.

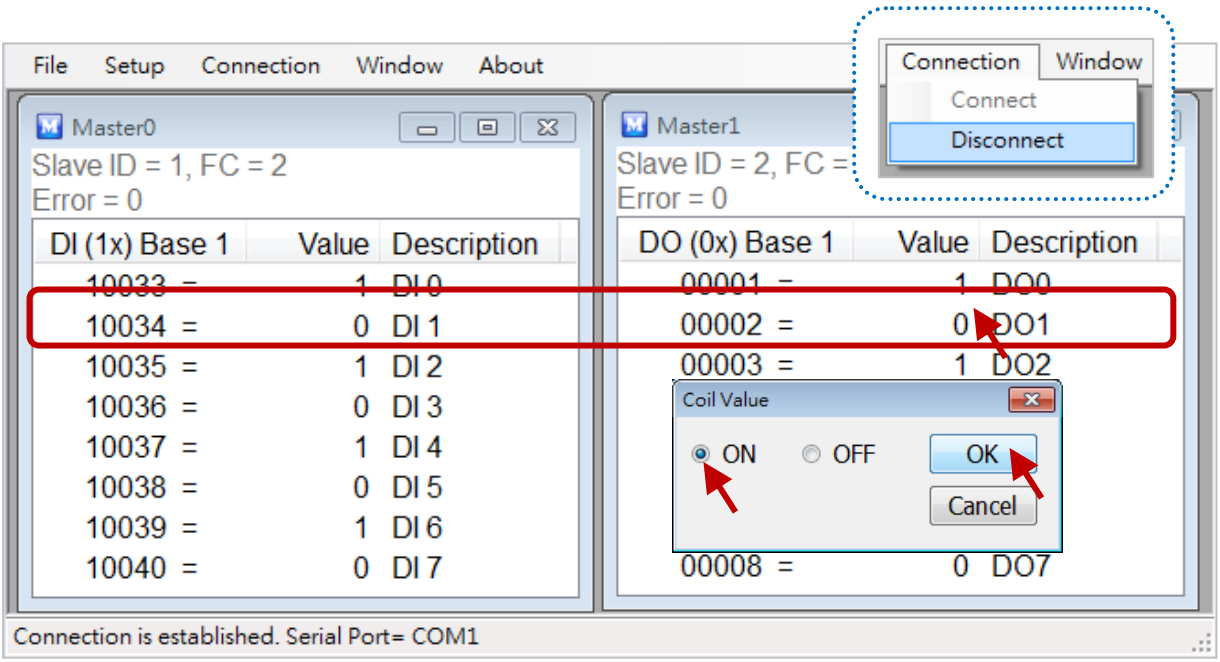


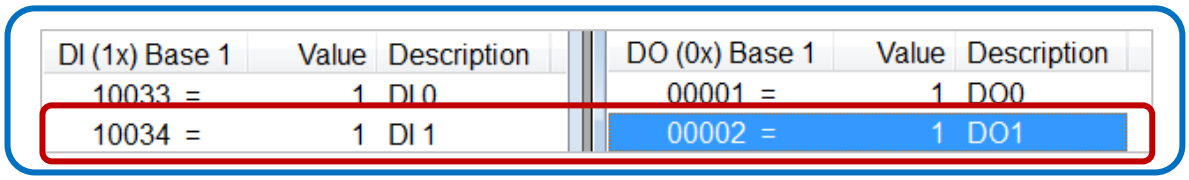


Step 4: Test the DI and DO modules

After connecting with modules, both the current DI and DO status of modules will be displayed. Double-click on any DO channel to modify the DO value and then to view the change of DI value.

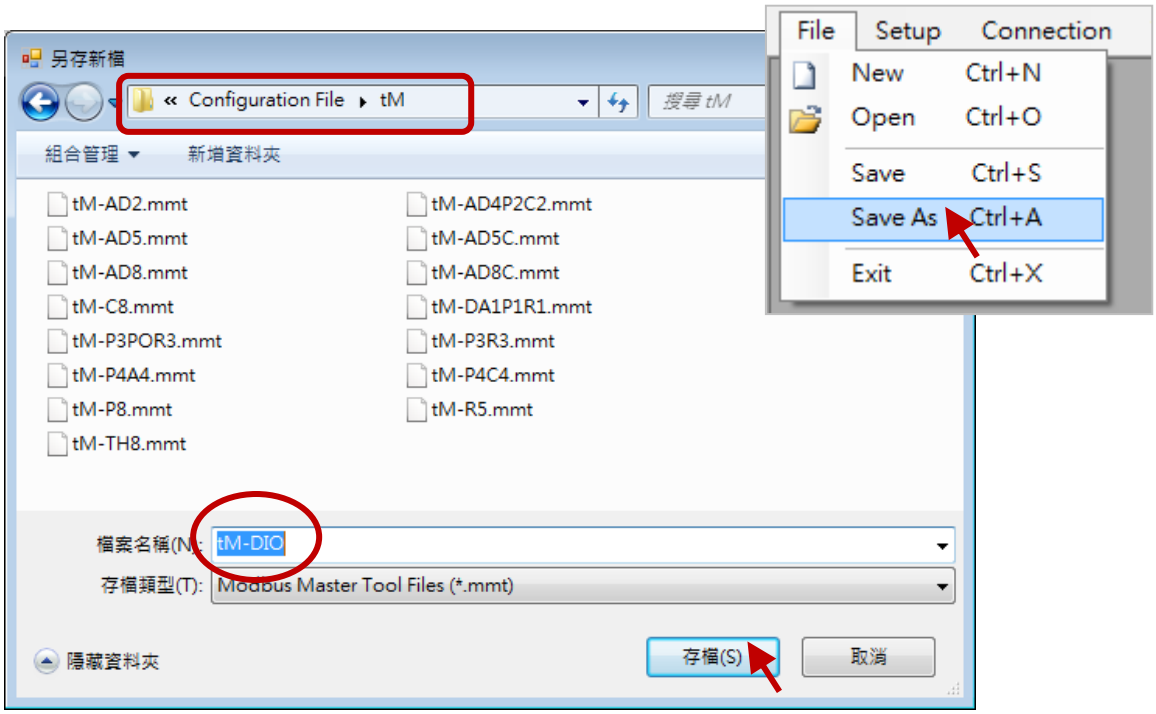
Note: To disconnect with modules, simply click “Disconnect” from the “Connection” menu.





Step 5: Save the configuration file

After completing the test, click “Save As” from the “File” menu to save the configuration file.



Chapter 5 Software Tool – EZ Data Logger

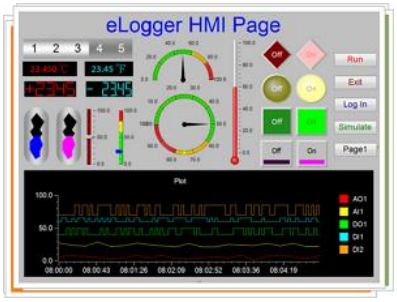
Note: **EZ Data Logger no longer provides a new function since July 2019. It is recommended to use eLogger HMI.**

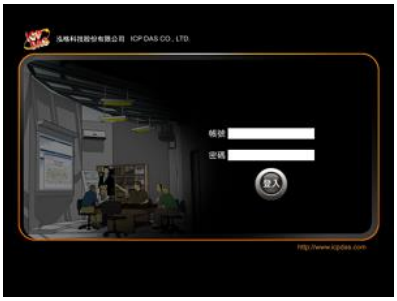
Support Items	EZ Data Logger	eLogger
Hardware platform	PC	PC, XPAC, WinPAC, ViewPAC, iPPC Win-GRAF series PAC
OS	Windows XP, 7, 10	Windows XP, 7, 10, WinCE 6, 7
HMI	Yes	Yes
Web HMI	No	Yes
Data Logger	Access, MS SQL, MySQL	CSV, MS SQL, MySQL
Protocol	Modbus RTU, Modbus TCP, and DCON	Modbus RTU, Modbus TCP, and I-8K/I-87K on slot
3rd Party Driver Support	No	Driver develops tool to add plug-in 3rd party driver

eLogger is a free and easy HMI development platform designed by ICPDAS. eLogger can not only be used to design local and Web Server HMI but can also achieve remote PAC control via a Web browser on PC or smartphone. With Win-GRAF, it is easy to create a professional monitoring application without any advanced programming background.

Website: https://www.icpdas.com/en/product/guide+Software+eLogger+eLogger

eLogger Features:





- **Support Modbus RTU/ASCII and Modbus TCP Protocols**
- **Support Real-time Data Trend**

  The maximum of five lines in one plot.
- **Support Local HMI:**

  Support a variety of HMI elements and a maximum of 32 pages.
- **Support Web Server HMI:**

  Support administrator login.
- **Support Account Management (for Local HMI):**

  3-level operating management: Admin, Power User, User.
- **Support Remote Maintenance:**

  eLogger is not only supported to upload a program or web pages and execute (or stop) the program through the Internet.
- **Support Data Logging**

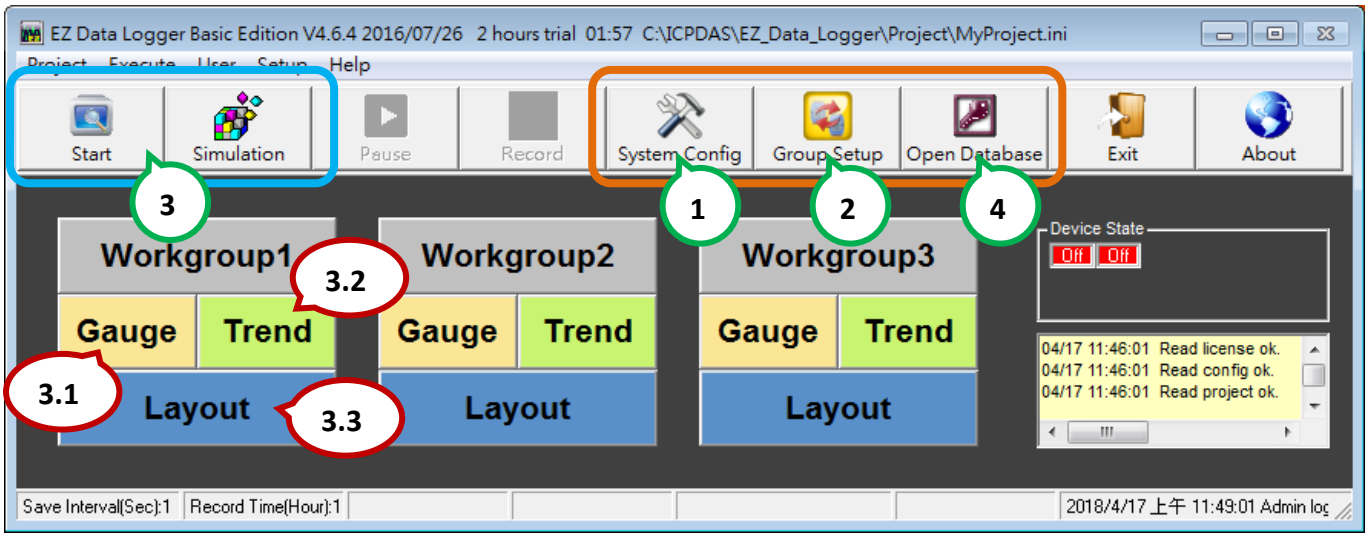
  Local database: Support CSV files.

  Remote database: Support Microsoft SQL Server 2005 or later.

EZ Data Logger is a small data logger software. It supports DCON and Modbus RTU/ASCII protocols, and users can quickly and easily build a data logger software without any programming skill.

To download the software and the user manual (.chm) on the website.
https://www.icpdas.com/en/product/guide+Software+EZ__Data__Logger+EZ__Data__Logger

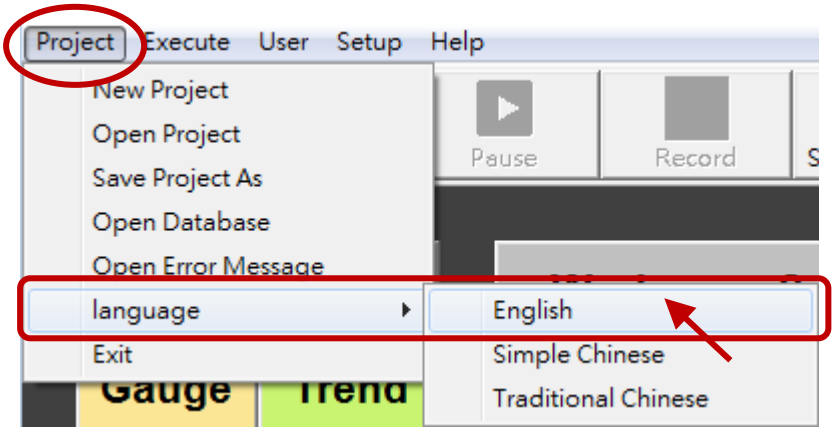
Install and run EZ Data Logger. The software provides four major steps and will be described in the following content.



Step 1: System Configuration
Step 2: Workgroup setting
Step 3: Start to Run (or Simulation)
- 3.1 Gauge
- 3.2 Trend Line
- 3.3 Layout

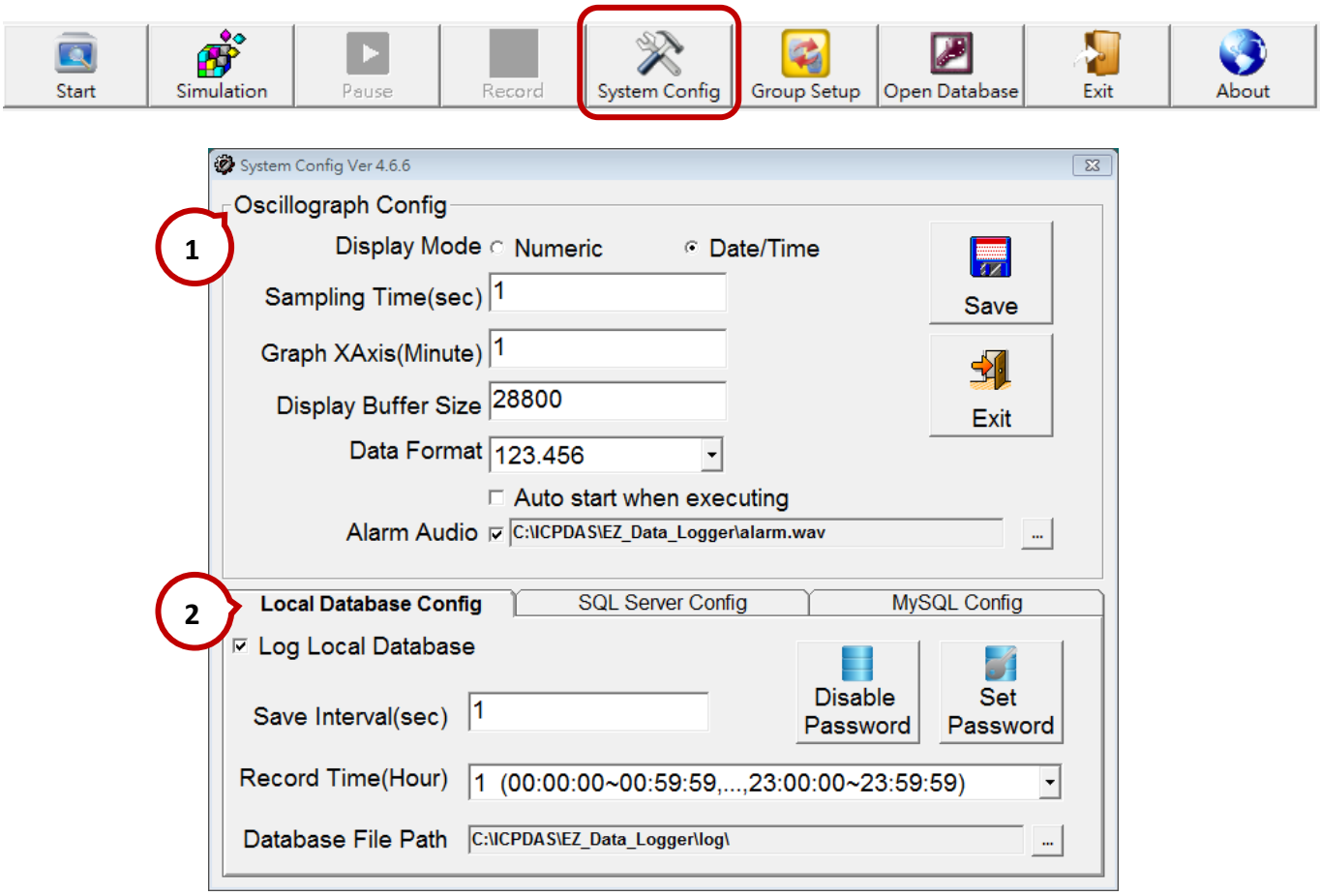
Step 4: Open Database

First, choose a preferred display language from the Project menu for the software.



5.1 System Configuration

The "System Config" window provides the settings for the trend line and the database.



Oscillograph Config	
Display Mode	Numeric: Log data by list. The minimum sampling time is 0.01 sec. Date/Time: Log data by time. The minimum sampling time is 1 sec.
Sampling Time (sec.)	The time interval for scanning IO channels, updating database, and display values.
Graph X Axis (Minute)	The span of the trend display.
Display Buffer Size	Maximum points of each trend line.
Data Format	The format of the display values.
Auto start when executing	Check the box to automatically log data when the program is launched.
Local Database Config	
Save Interval (sec.)	Time interval of saving data in a database (a multiple of sampling time).
Record Time (Hour)	EZ Data Logger will create a new database when the record time is up.
Database File Path	Auto renamed by Datayyyymmdd_hh.mdb when record time is up (yyyy: year mm: month dd: day hh: hour). The database will be saved in the path.

5.2 Workgroup Setting

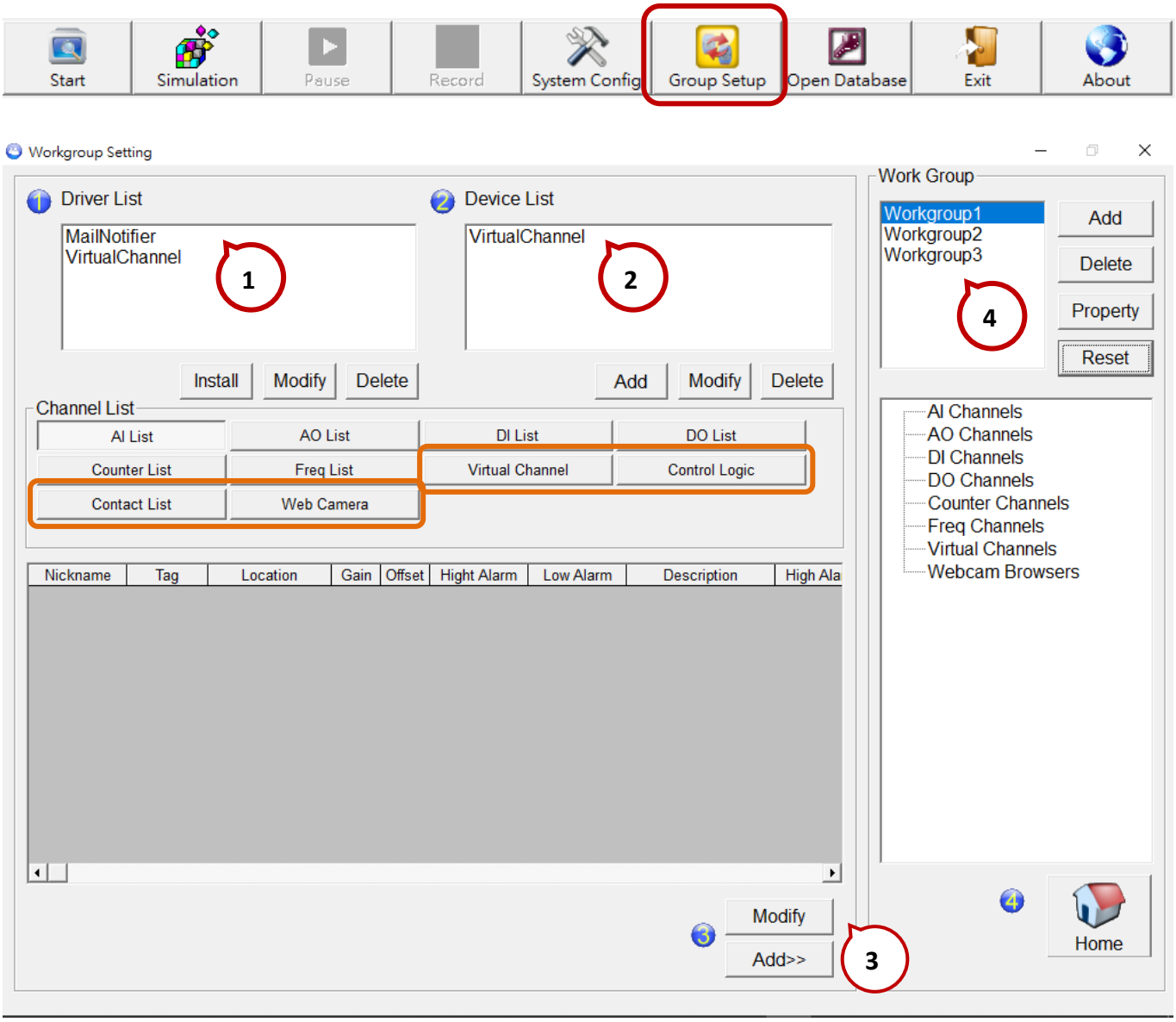
The "Workgroup Setting" window can be divided into four major steps.

Step 1: Install the Driver

Step 2: Add a Device

Step 3: Modify the Channel Settings or Add a Workgroup

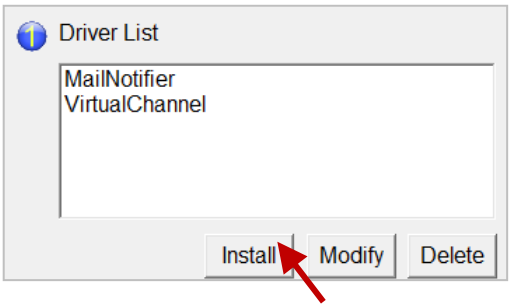
Step 4: Configure the Workgroup Settings



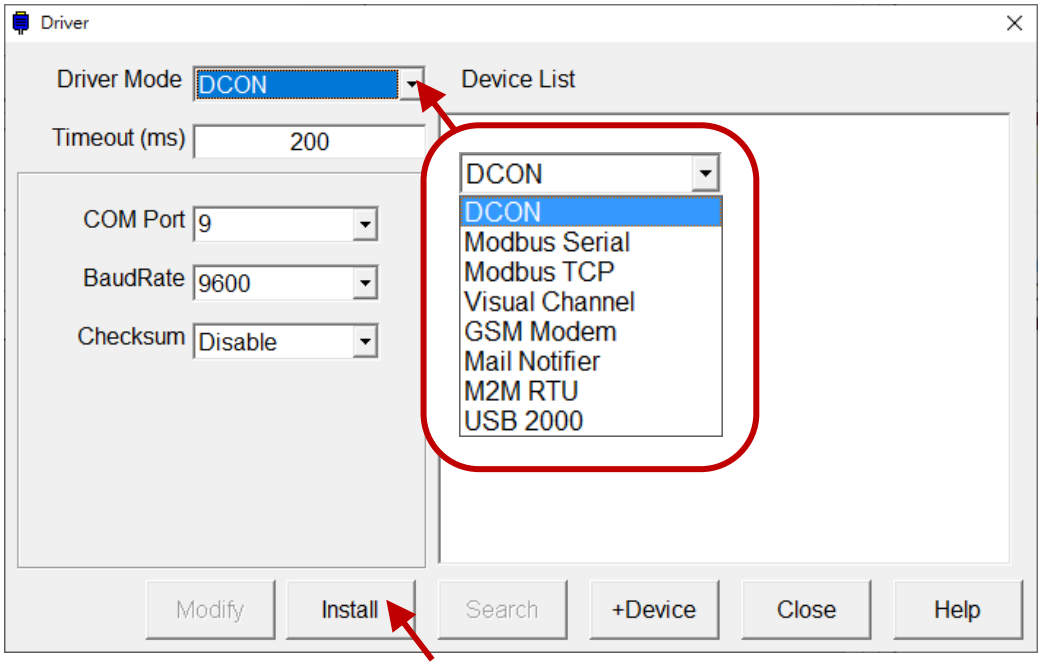
Note: For more information about the Virtual Channel, Control Logic, Web Camera, Mail Notifier, and User Level, refer to EZ Data Logger User Manual (.chm).

5.2.1 Install the Driver

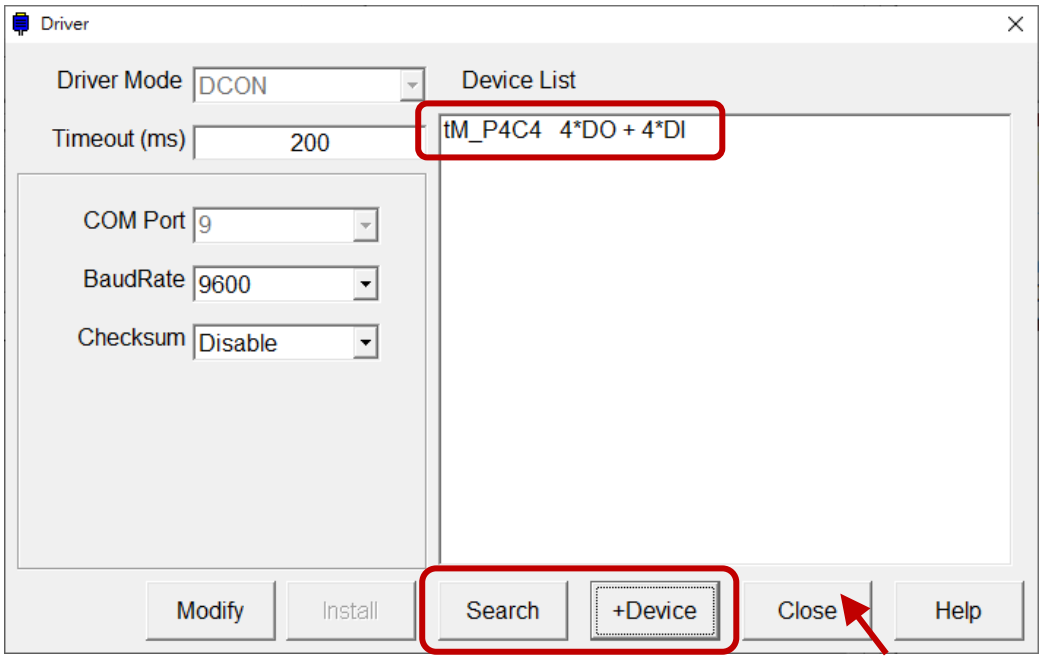
1. Click the “Install” button on the Driver List.



2. Configure the Driver Mode, Timeout (ms), COM Port, Baud Rate, and Checksum to meet the needs of equipment, and then click the “Install” button.

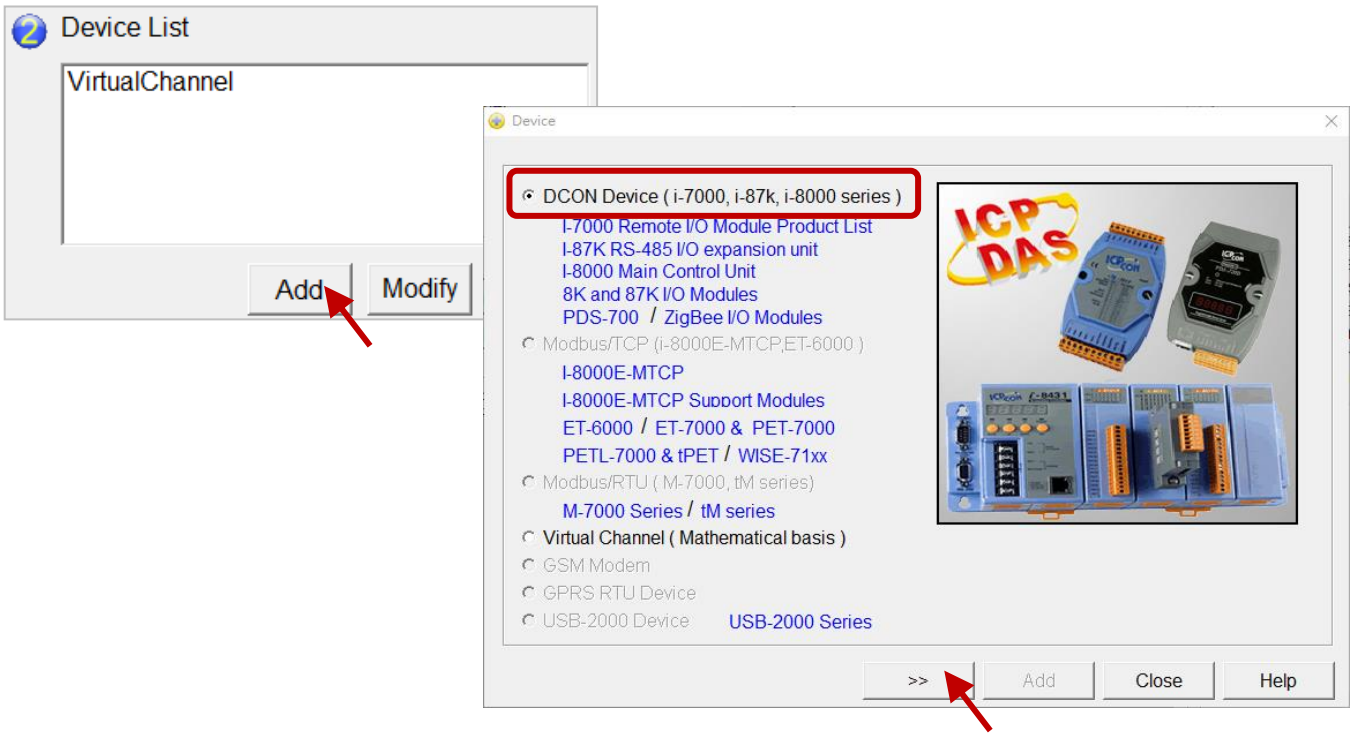


Also, simply click the “+Device” button to add a device (reference: Section 5.2.2 – Install the Device). Alternatively, click the “Search” button to search the connected device, and then click the “Close” button to close the window.

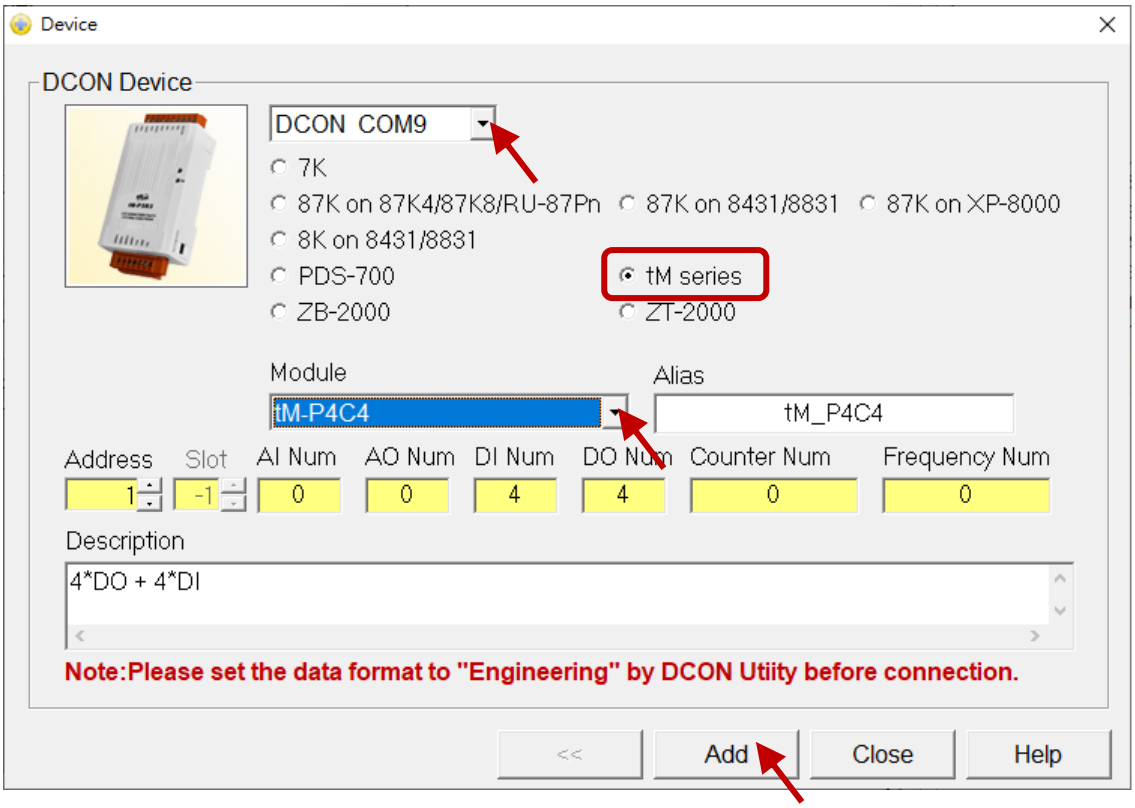


5.2.2 Add a Device

1. Click the “Add” button on the Device List, and click the " >> " button in the Device window.



2. Choose a communication setting (e.g., DCON COM9), select the name of the tM series module also add an alias, and click the Add button.

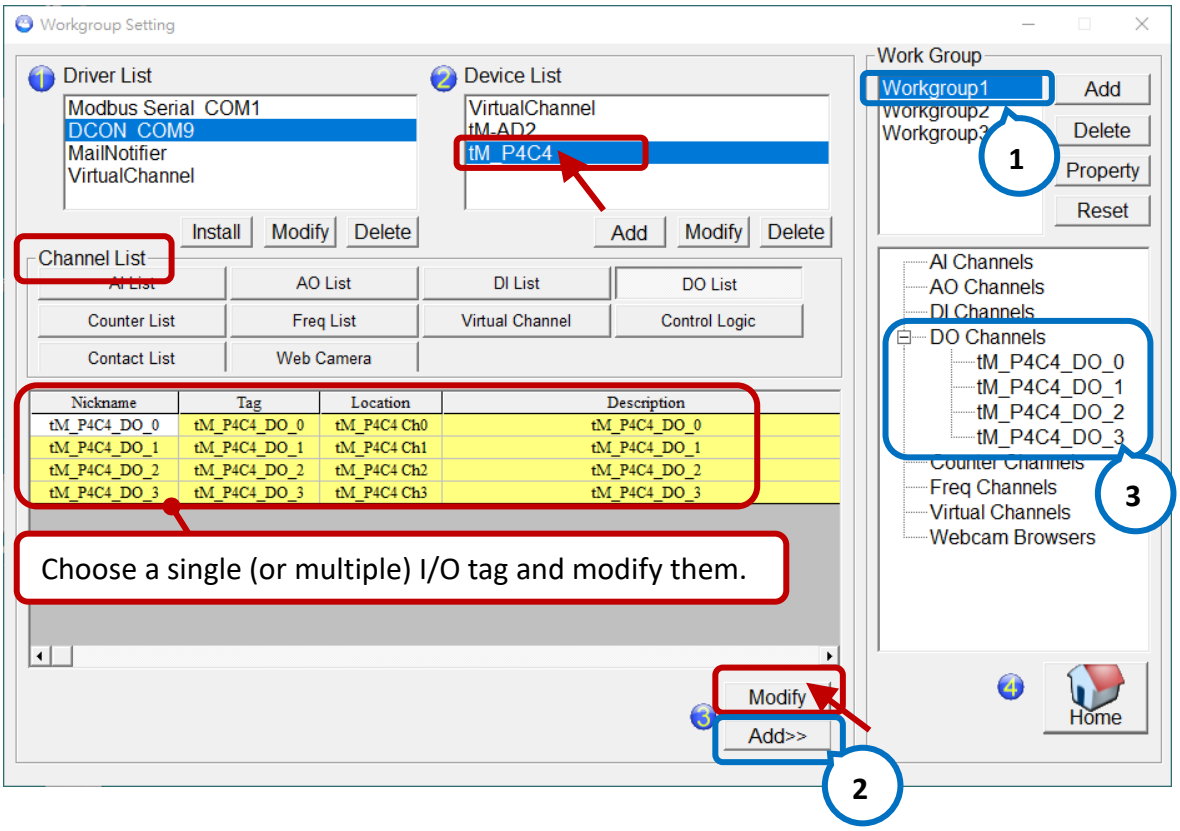


5.2.3 Modify the Channel Settings or Add a Workgroup

After clicking the module name (e.g., tM_P4C4) in the Device List, all I/O tags in the Channel List will be selected. To add them to the Work Group, specify the number of the workgroup (e.g., Workgroup1) and click the “Add >>” button.

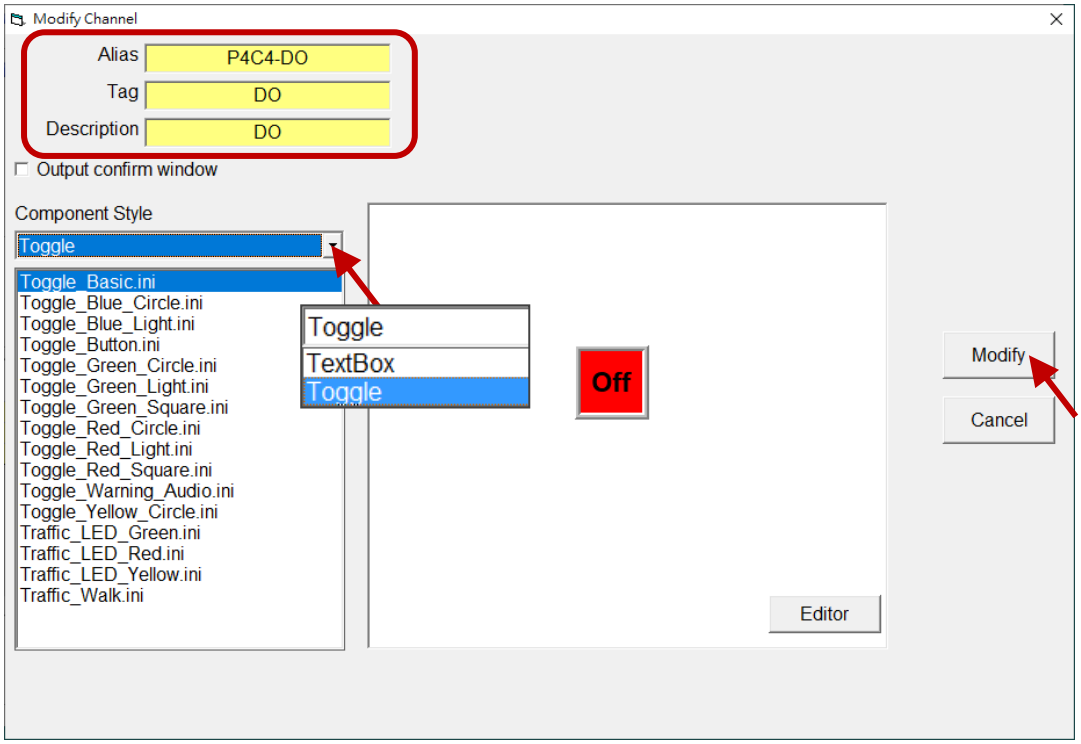
Modify the Channel Settings

1. Choose a single (or multiple) I/O tag for the specified tM module and click the Modify button.



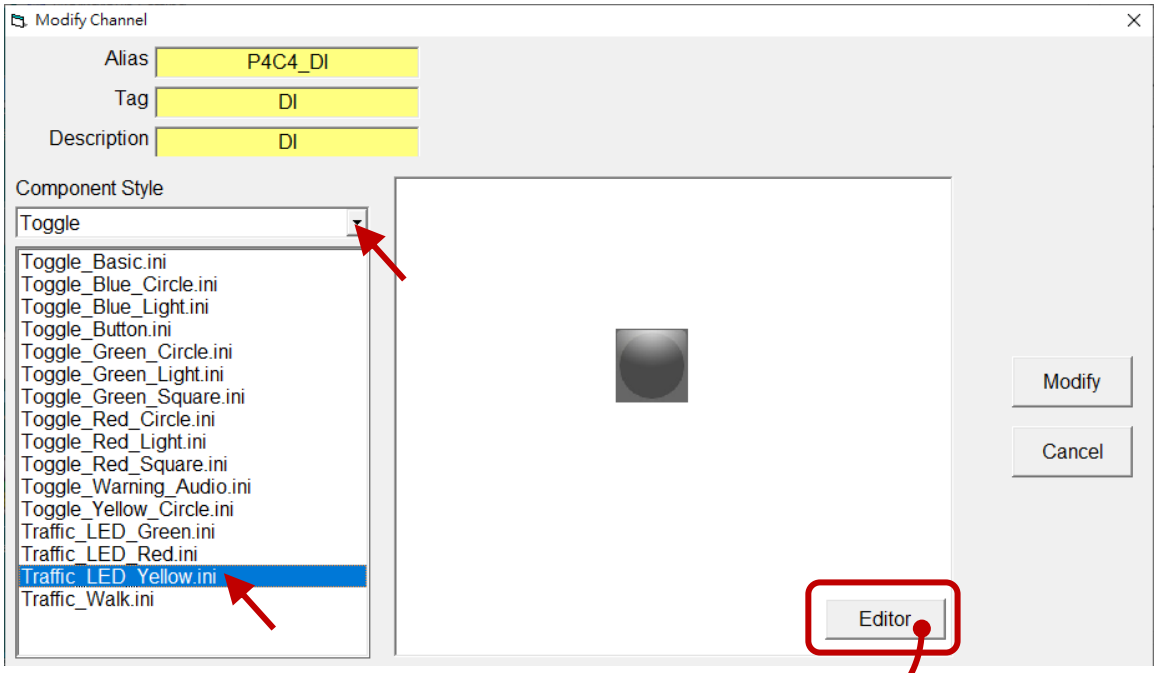
2. Change to the needed settings in the Modify Channel window, and then click the Modify button.

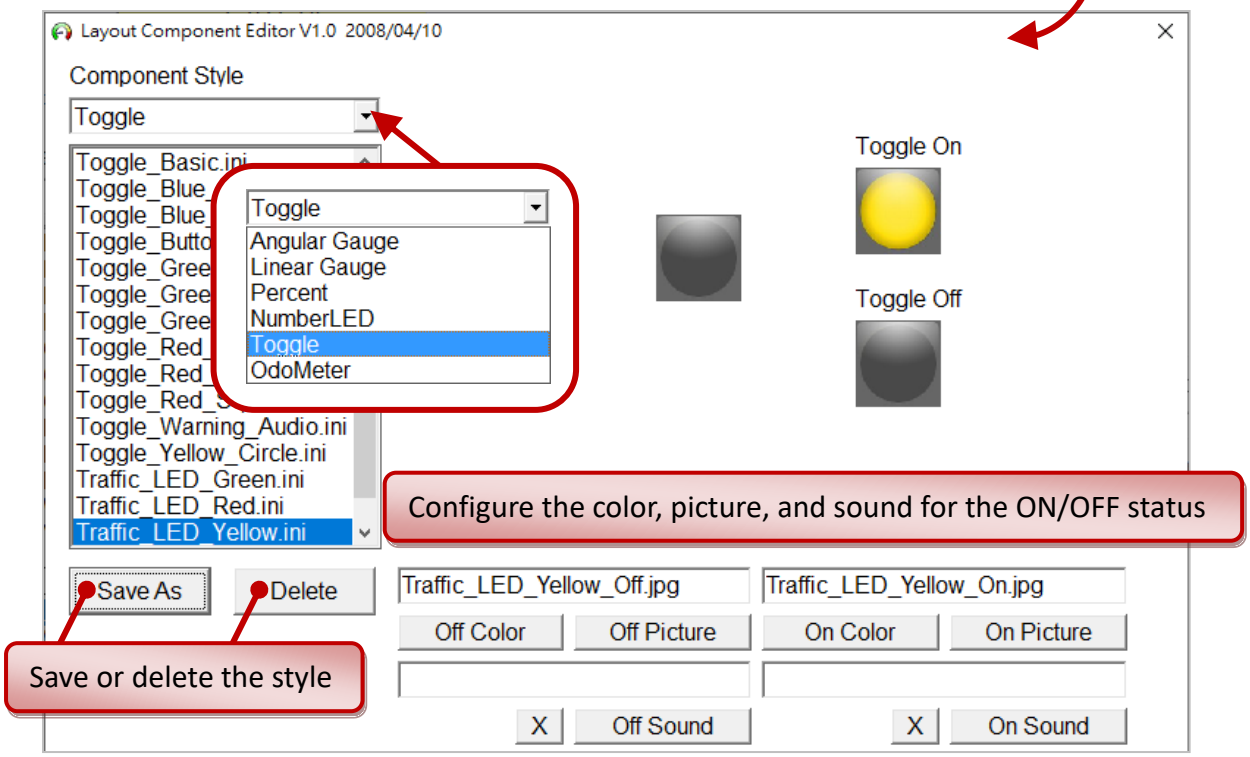
DO channels:



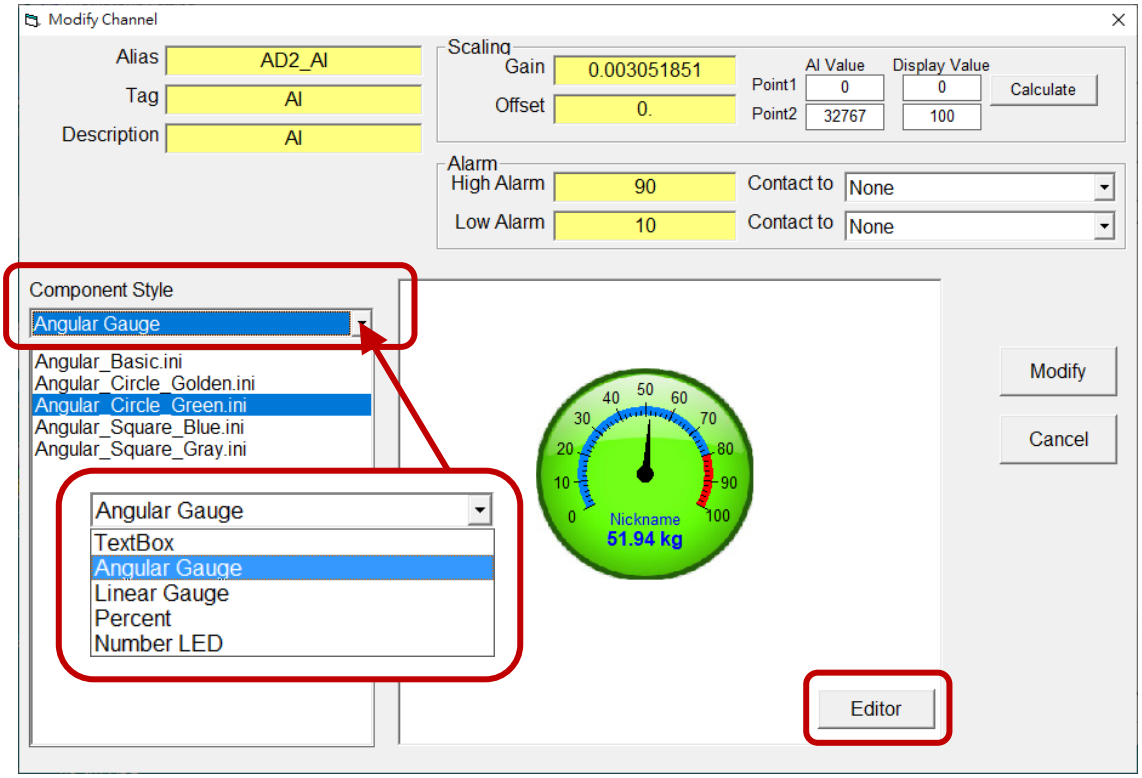
DI, DO channel	
Alias	The name that easy to identify I/O tag
Tag	Tag name that can be used for the Virtual Channel or Control Logic function. Note that the name must begin with the alphabet from a to z and the first character cannot be a number.
Description	Used to display the hint for the object
Output confirm window	Check the box to display a confirmation window before outputting a value
Component Style	Select the style of an object to be displayed on the Layout screen
Editor	Click the button for modifying the style of an object

DI channels:

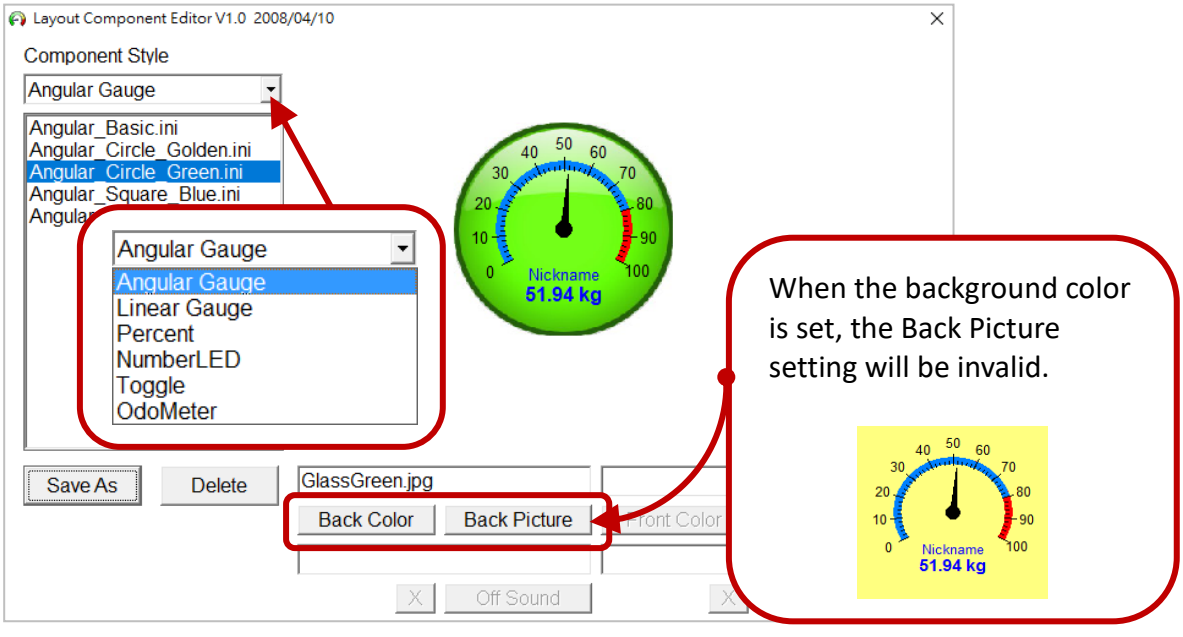




AI channels:



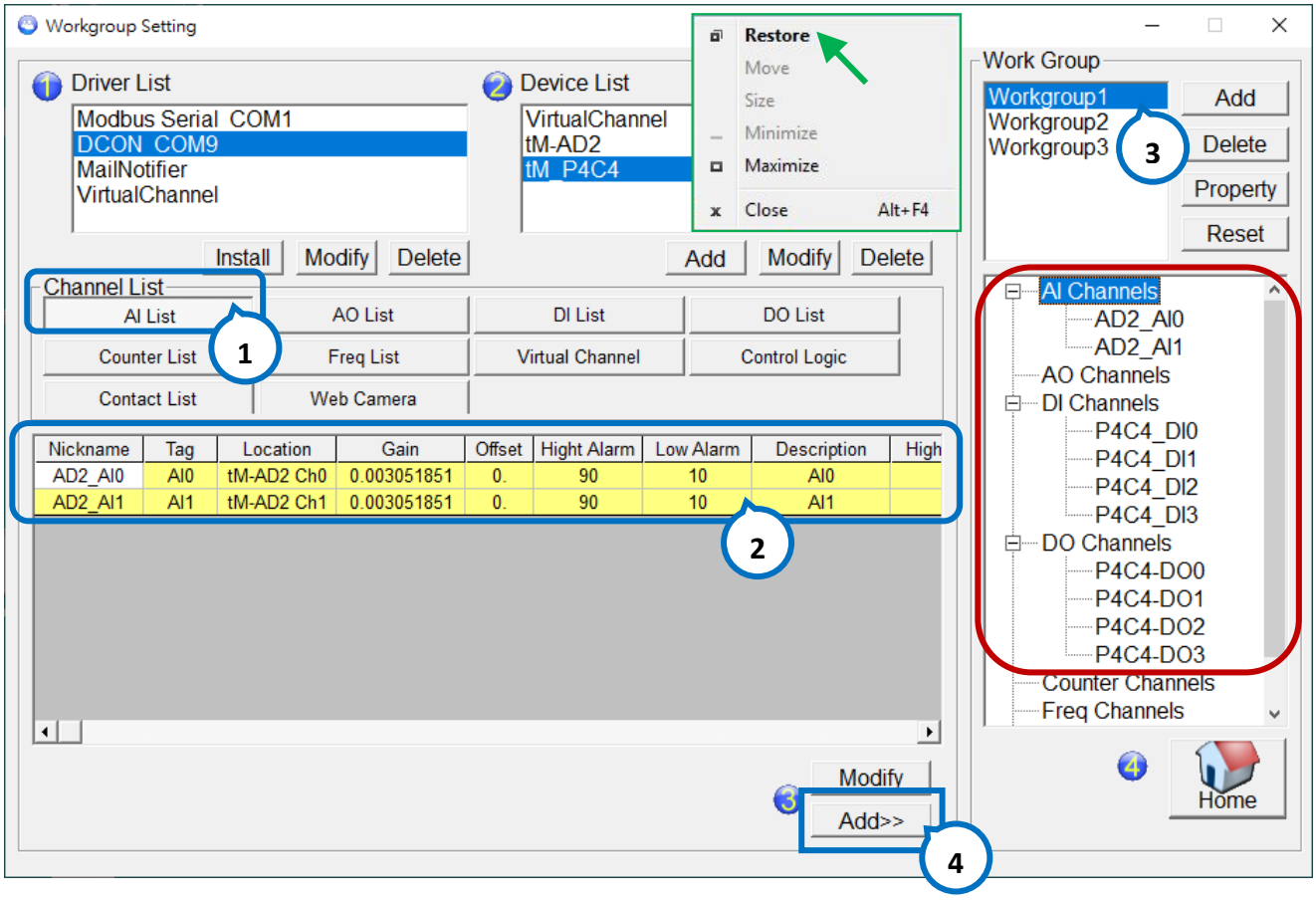
Alias	The name that easy to identify I/O tag
Tag	Tag name that can be used for the Virtual Channel or Control Logic function. Note that the name must begin with the alphabet from a to z and the first character cannot be a number.
Description	Used to display the hint for the object
Scaling	Enter AI Values and Display Values, and click the Calculate button to calculate both the Gain value and Offset value.
Alarm	When the value is out of the High/Low Alarm value, a warning light comes on and plays sounds until it is closed. (Reference: 5.3.1 the Gauge screen)
Editor	Click the button for modifying the style of an object

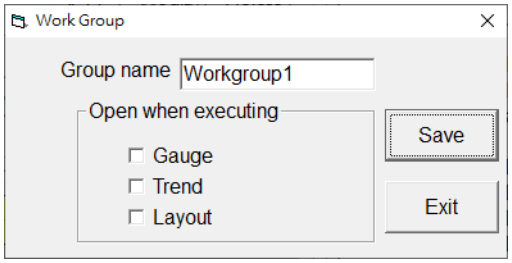
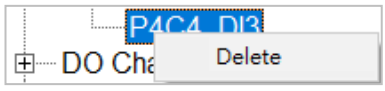


5.2.4 Configure the Workgroup Settings

As described in Section 5.2.3, users can choose multiple I/O tags and add them to the specified workgroup. Each of the workgroups can add a maximum of 32 I/O tags.

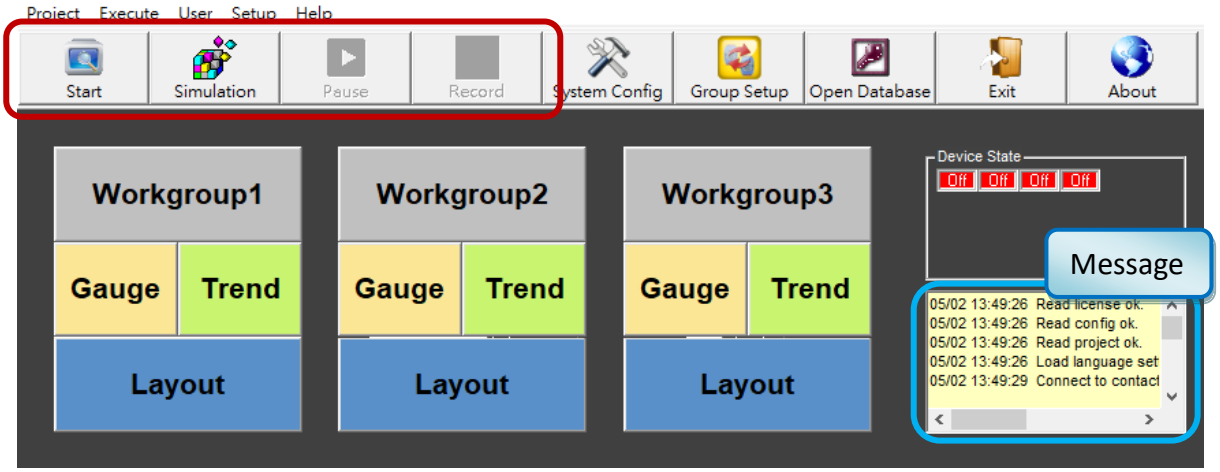
Note: If you cannot reduce the size of the window, simply right-click on the title bar and click Restore.



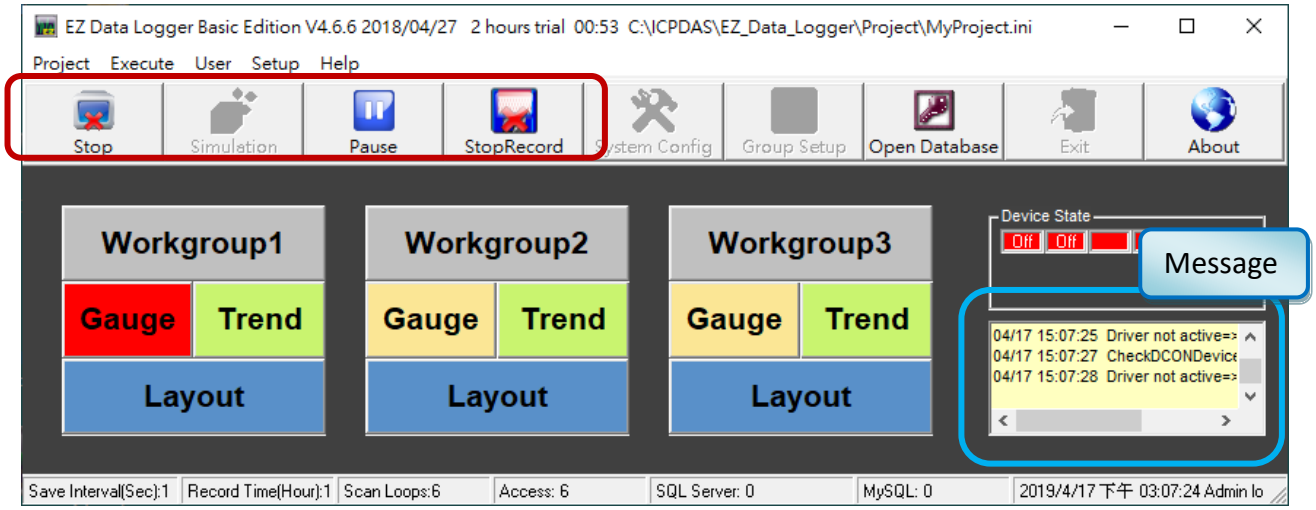
Work Group	
Add	Create a workgroup
Delete	Delete the specified workgroup
Property	<u>Group name</u>: Set a name that is easy to identify <u>Open when executing</u>: If I/O tags are added, check the Gauge, Trend, or Layout box to display the specified screen when the program is launched. 
Reset	Remove all I/O tags
Channels	To delete a single I/O tag, right-click on the tag name and click Delete. 
Home	Go back to the main window

5.3 Start to Run (or Simulation)

After configuring I/O channels and Workgroups, simply start to log data or execute the simulation.

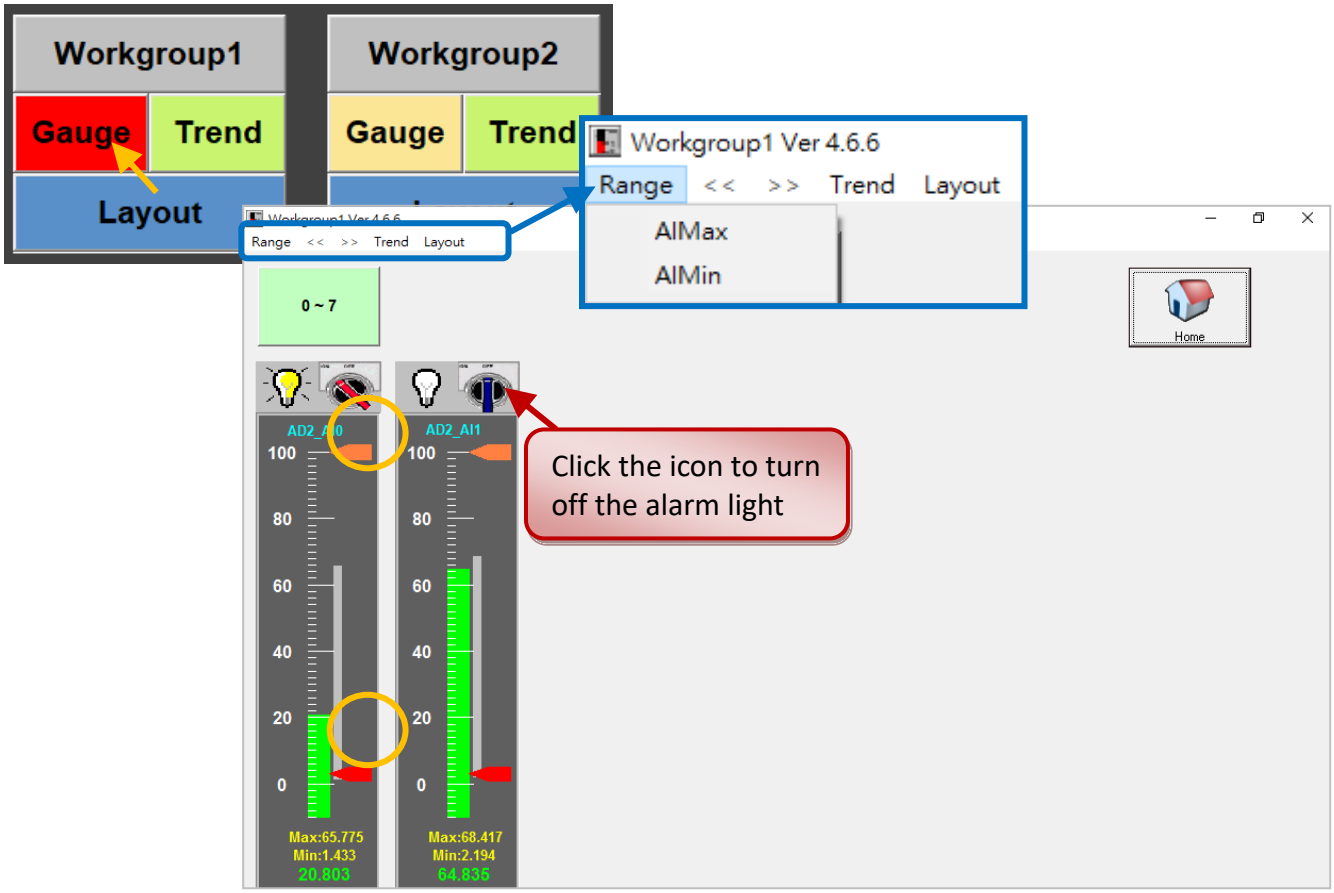


Start/Stop	Start or Stop logging and recording data
Simulation/Stop	Start or Stop simulating and recording data
Pause/Continue	Pause or Continue the dynamic trend display
Record/Stop Record	Start or Stop recording data to the database
Message	Display messages for the data logging



5.3.1 Gauge

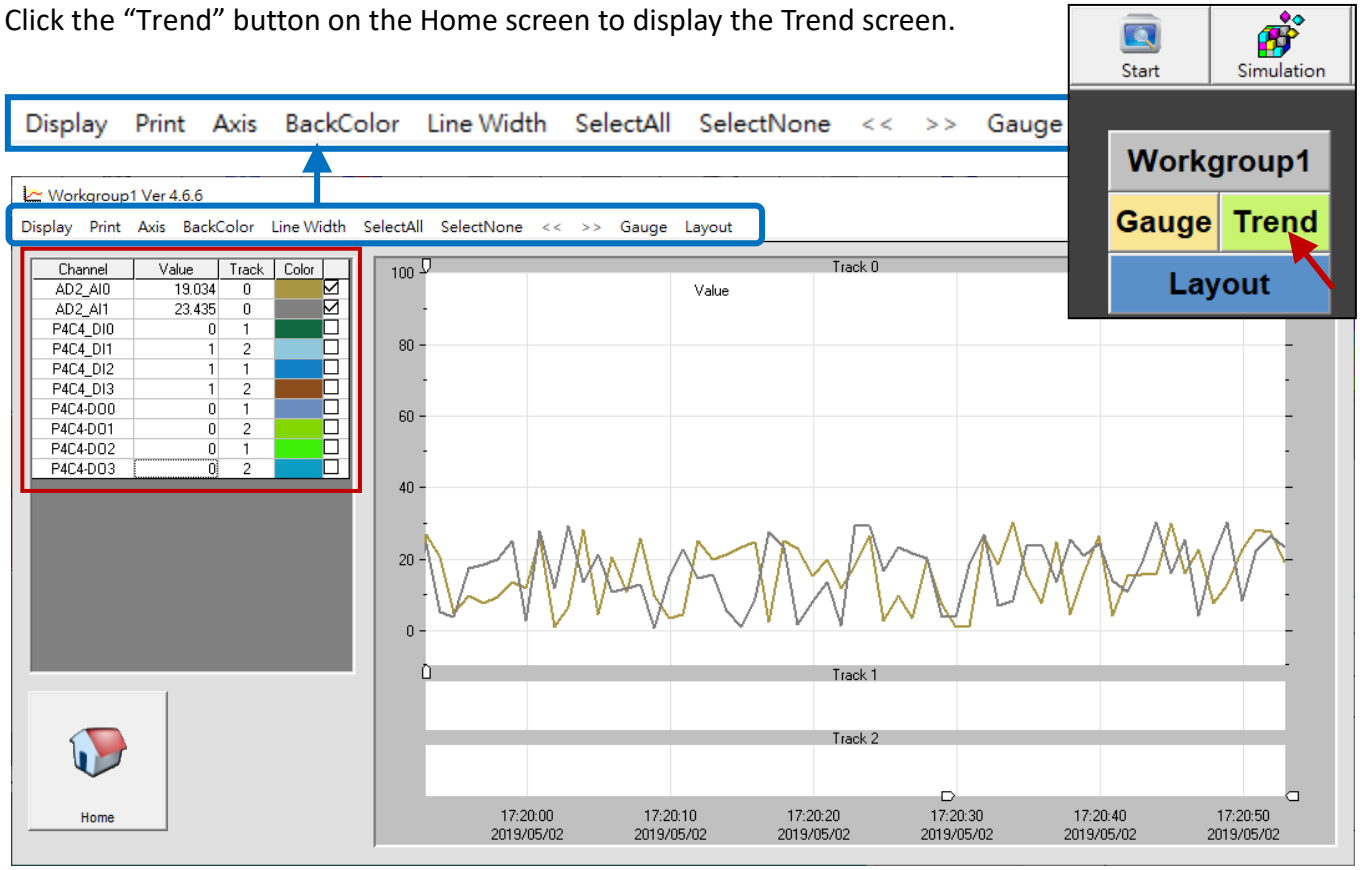
Click the “Gauge” button on the Home screen to display the Gauge screen. If the logged or simulated AI value is exceeding the high/low alarm value, an alarm light will be turned on with an audio warning until the light is turned off.

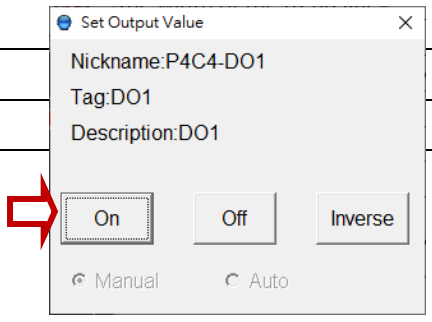


Range	Used to set the maximum/minimum of analog input value if the switch is set to OFF. The orange arrow stands for the high alarm value (e.g., 90) The red arrow stands for the low alarm value (e.g., 10)
<<	Click to switch to the previous workgroup if exists
>>	Click to switch to the next workgroup if exists
Trend	Click to switch to the Trend screen
Layout	Click to switch to the Layout screen
0 ~ 7 8 ~ 15	If the number of channels is more than 8, click the button to switch the screen

5.3.2 Trend Line

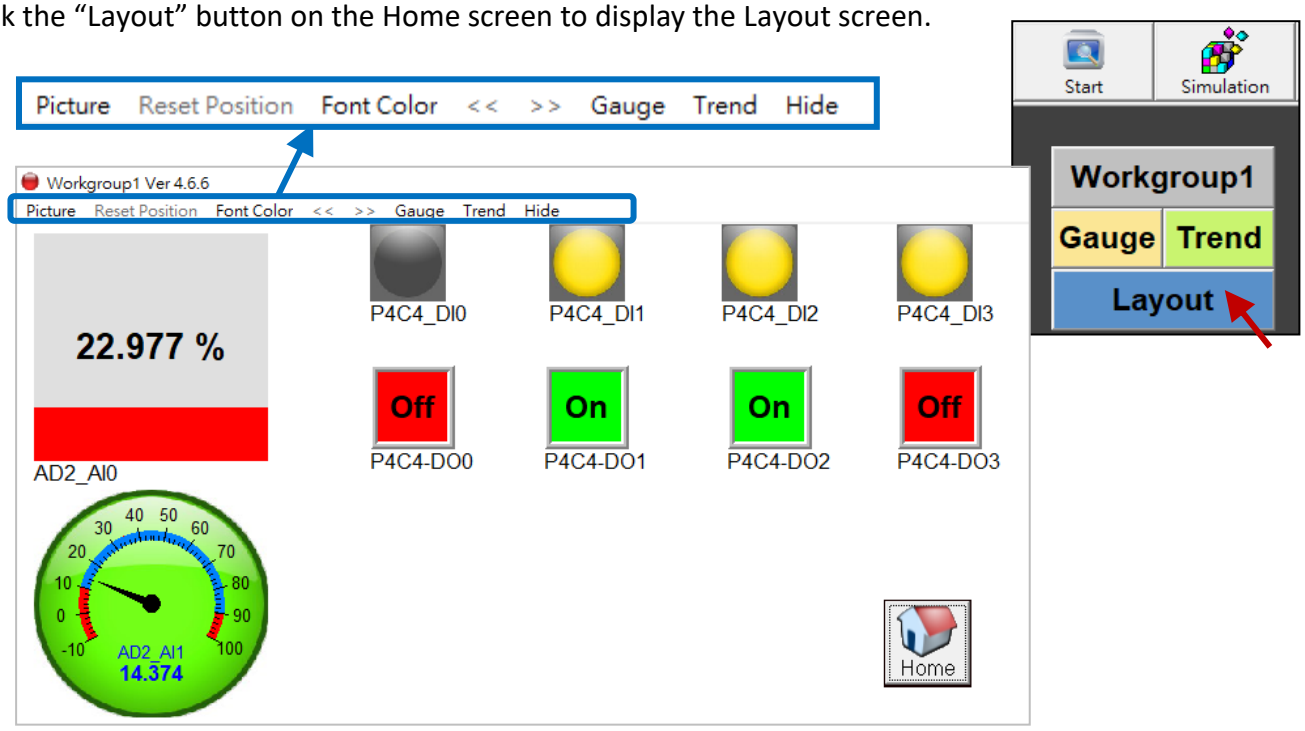
Click the “Trend” button on the Home screen to display the Trend screen.

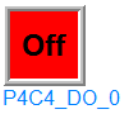


Display	Pause or continuous the dynamic trend
Print	Print the trend includes the group name and file time
Axis	Specify the maximum and minimum values on the Y-axis and the range of time in minutes on the X-axis
BackColor	Specify the background color of the trend
Line Width	Specify the width of the trend line
Select All/ Select None	Display or hide all trend lines
<< / >>	Switch to the previous or the next workgroup
Gauge	Switch to the “Gauge” screen
Layout	Switch to the “Layout” screen
Channel	The name of the channel
Value	When executing data logging (or the simulation), click any AO/DO channel on the list to set the output value 
Track	When executing data logging (or the simulation), click any DI/DO channel to change the display track (1 or 2).
Color	Specify the color of the trend line. Note that the color cannot be changed when executing data logging (or the simulation)

5.3.3 Layout

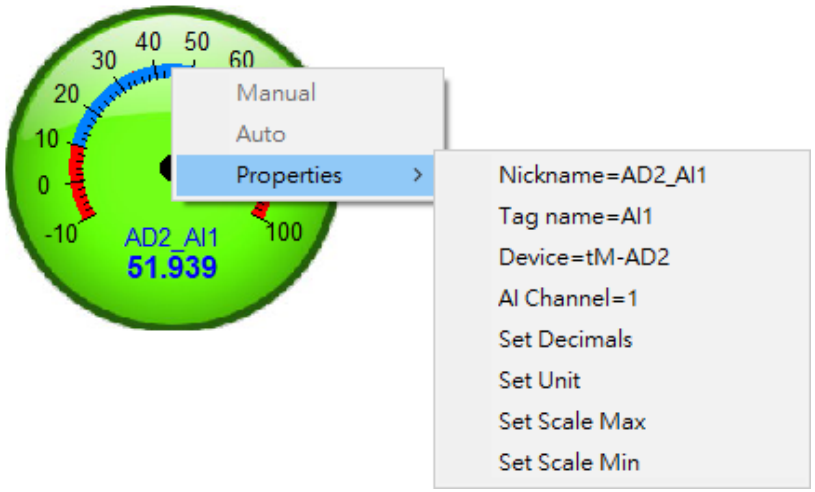
Click the “Layout” button on the Home screen to display the Layout screen.



Picture	Set or cancel the background picture, the jpg or bmp format is supported.
Reset Position	Reset the position for all objects on the screen
Font Color	Set the color of the tag name 
<< / >>	Switch to the previous or the next workgroup
Gauge	Switch to the “Gauge” screen
Trend	Switch to the “Layout” screen
Hide	To hide the current screen

Right-click menu:
On the Layout screen, the right-click menu of the object provides some options that can be used to manually or automatically output value or set properties.

Note: The “Manual” and “Auto” options are used for logical control, refer to EZ Data Logger User Manual (.chm) for more information. When the control logic is active (reference: Section 5.2 - Workgroup Setting - Control Logic), EZ Data Logger will automatically output AO or DO value according to the logic. Also, select "Manual" to manually change the value.



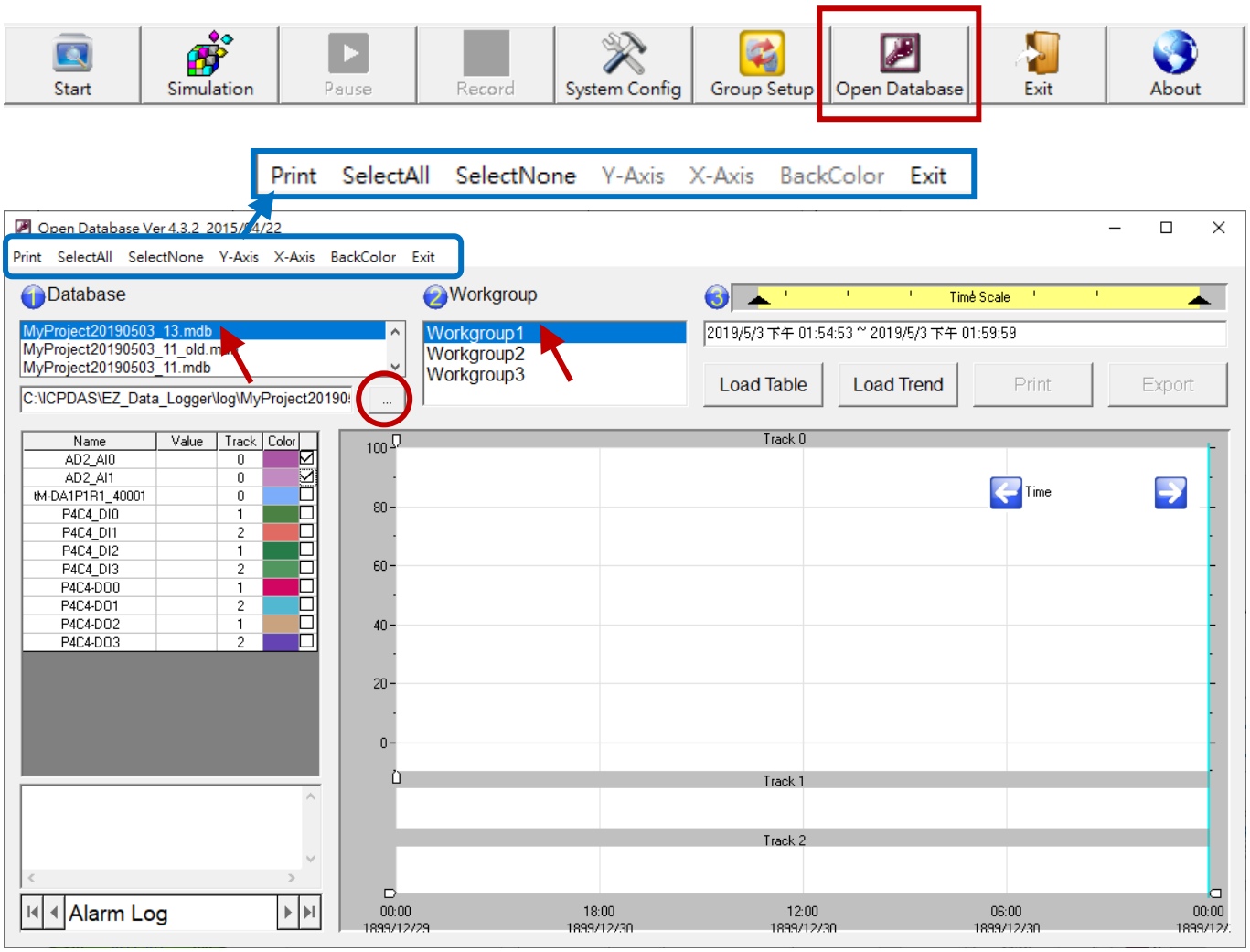
For example, the value = 123.456)

Decimals	Unit	Display
0	℃	123 ℃
1	℉	123.5 ℉
2	kg	123.46 kg
3	V	123.456 V

Properties	
Nickname	The specified alias in the Channel list (Reference: Section 5.2.3)
Tag name	The specified tag name in the Channel list (Reference: Section 5.2.3)
Device	The specified name of a module in the Device list (Reference: Section 5.2.2)
I/O Channel = n	Display the number of the I/O channel
Set Decimals	Specify the display decimal point, it can be 0 to 3
Set Unit	Specify the display unit, e.g., ℃
Set Scale Max	Specify the display maximum scale value, e.g., 100
Set Scale Min	Specify the display minimum scale value, e.g., -10

5.4 Open Database

In the Open Database window, it allows opening a log file (.mdb) and export data into an Excel or CSV file.



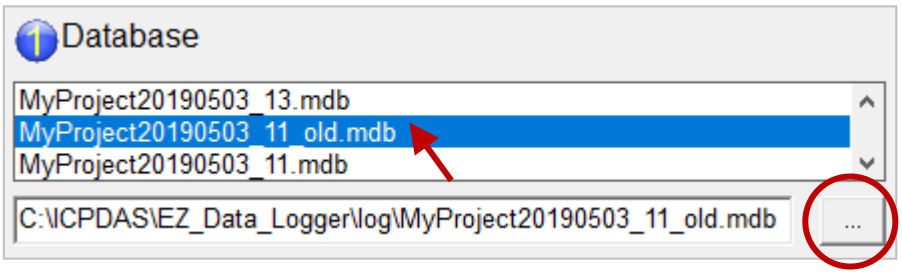
The menu bar provides functions to print or display data for each I/O channel and configure the style of a trend. Follow the steps listed on the next page to load the data table or trends.

Print	Print the data table or trends
Select All / Select None	Check or uncheck all box to display or hide the logged data of all channels
Y-Axis	Specify the maximum (or minimum) display value on Y-axis
X-Axis	Specify the display time scale on X-axis in minutes
BackColor	Specify the background color of the trend
Exit	Close the window

Note: When logging (or simulating) data, the settings of Y-Axis, X-Axis, and BackColor cannot be changed.

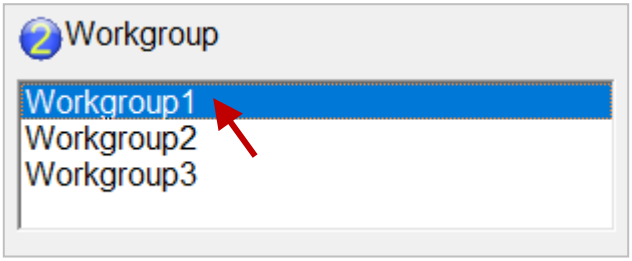
Step 1: Select a database:

Choose a log file in the list or click the browse button to open a file in a specific folder.



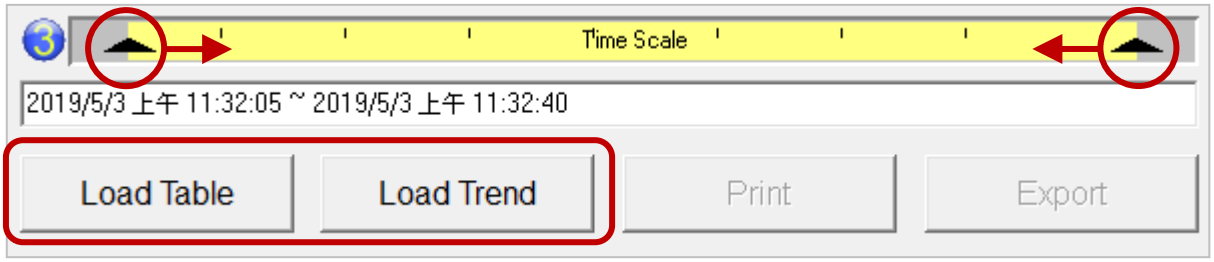
Step 2: Select a workgroup:

Choose a workgroup that contains I/O tags. (Reference: Section 5.2.3)



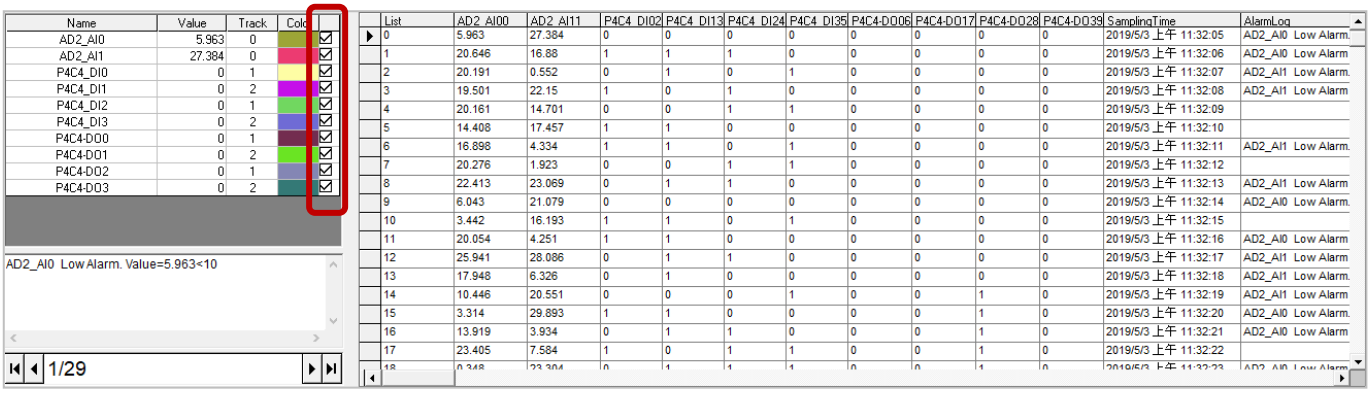
Step 3: Specify a time scale and load the table/trend:

Drag the arrow to the left or right to specify the time scale, and then click the **Load Table** or **Load Trend** button to load a data table or trends.



1) Load Table: Click the "Load Table" button to load a data table.

Note that check (or uncheck) the box for all channels to display or hide data.

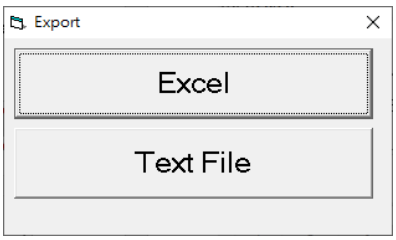


2) Load Trend: Click the "Load Trend" button to load data trends.

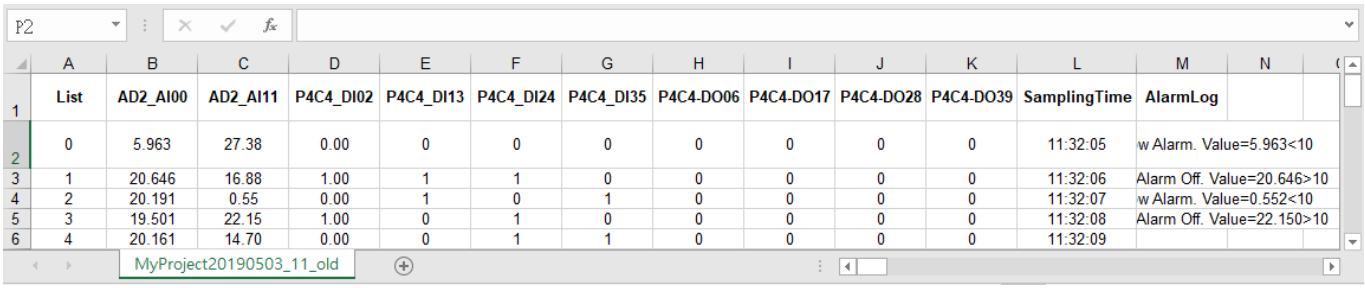
* Check (or uncheck) the box for all channels to display or hide trend lines.
* Click on the trend to display values at the specific time.
* Click the scroll button to display an alarm message.

3) Print: Click the "Print" button to print the trend.

4) Export: Click the "Export" button to save data in an Excel or CSV file.

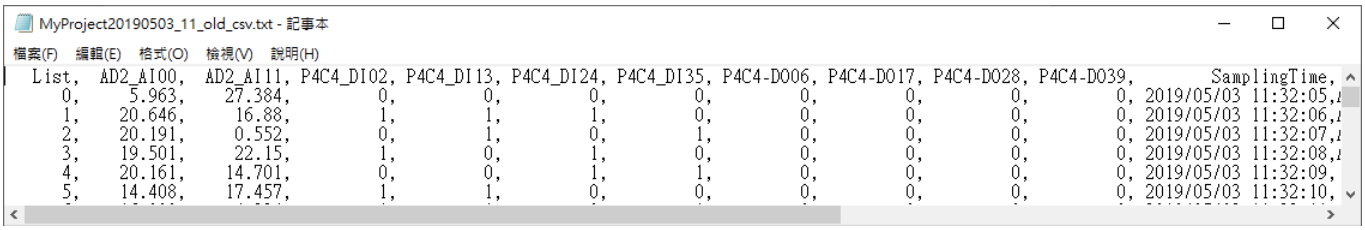


Export to an Excel file:



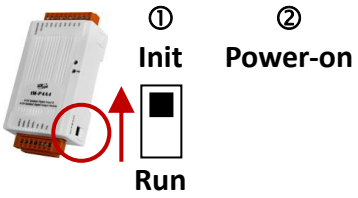
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	List	AD2_AI00	AD2_AI11	P4C4_DI02	P4C4_DI13	P4C4_DI24	P4C4_DI35	P4C4-DO06	P4C4-DO17	P4C4-DO28	P4C4-DO39	SamplingTime	AlarmLog	
2	0	5.963	27.38	0.00	0	0	0	0	0	0	0	11:32:05	w Alarm. Value=5.963<10	
3	1	20.646	16.88	1.00	1	1	0	0	0	0	0	11:32:06	Alarm Off. Value=20.646>10	
4	2	20.191	0.55	0.00	1	0	1	0	0	0	0	11:32:07	w Alarm. Value=0.552<10	
5	3	19.501	22.15	1.00	0	1	0	0	0	0	0	11:32:08	Alarm Off. Value=22.150>10	
6	4	20.161	14.70	0.00	0	1	1	0	0	0	0	11:32:09		

Export to a Text file:



Chapter 6 Software Development Tool

tM series modules support the DCON and Modbus protocols. When working with the DCON protocol, configure or get the module data by using DCON Utility Pro or PACSDK. When working with the Modbus protocol, develop your applications via co-work with software tools and SCADA described in Section 6.3.

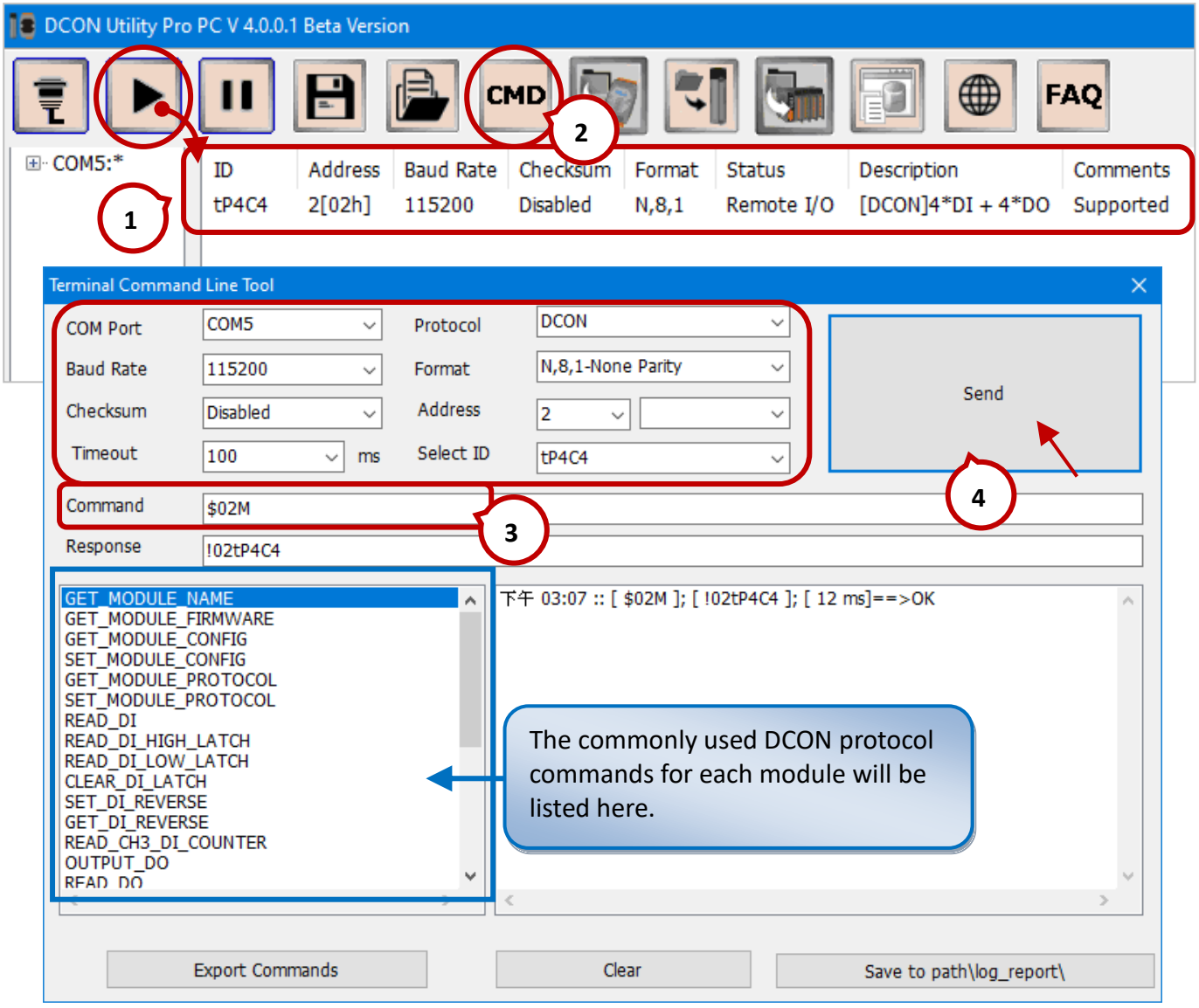


6.1 Using DCON Utility Pro

Step 1: The Protocol setting can be configured only the INIT switch is set.

Step 2: Search for the tM module and set the Protocol to DCON, and then rebooting in Run mode.

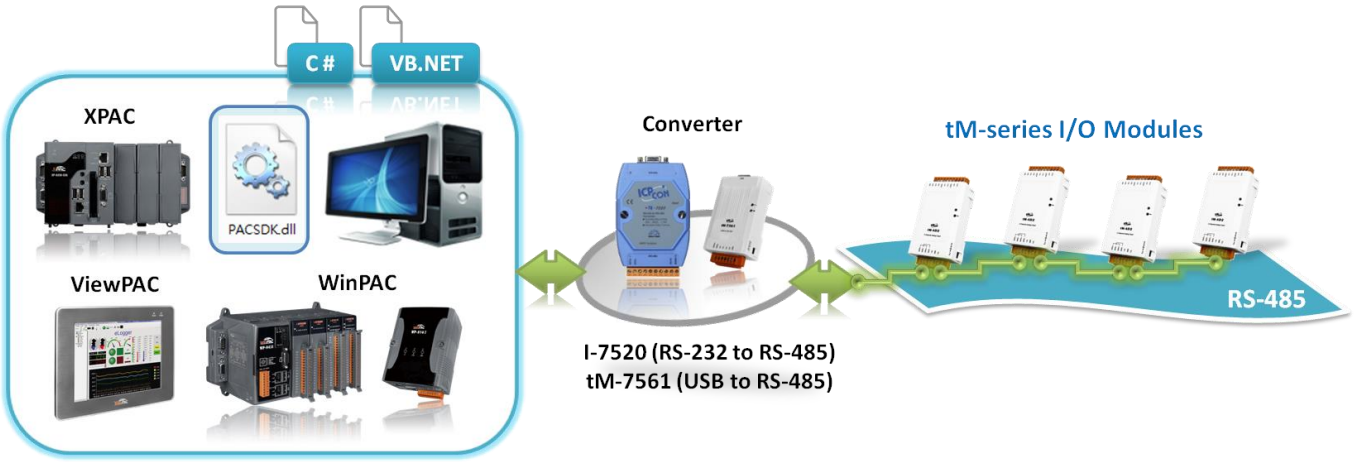
Step 3: Click the **Command Line** button and enter a DCON protocol command in the **Command** field. (Reference: Appendix E DCON Protocol Commands)



6.2 Using PACSDK

PACSDK are software development toolkits that can be used on a PC or PAC (e.g., XPAC, WinPAC, and ViewPAC series). PACSDK includes header files, libraries, documentation, and tools that can be used to develop VC/C#/VB.net applications to access I/O data of tM series module.

Note: PACSDK only supports the DCON protocol.



PACSDK

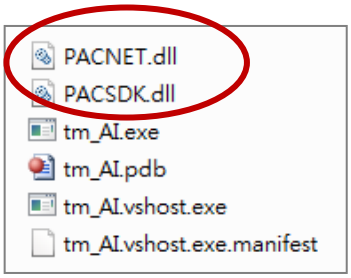
Items		PACSDK	Description
Header file		**PACSDK.h**	Using libraries when developing VC applications on Windows PC.
Library files		**PACSDK.lib**	
DLL files	For C program	**PACSDK.dll**	Loading libraries when executing VC applications on PC or PAC.
	For .NET program (i.e., C#, VB)	**PACNET.dll**	1) Using management libraries when developing .NET applications 2) Load libraries when executing .NET applications on PC or PAC.

Download Files on FTP:

1) PACSDK:

Users can download SDK according to the development platform (i.e., PC or PAC).

When using the following development platform - **PC or XPAC (WES),** copy PACSDK.dll and PACNET.dll to the folder where the applications (.exe file) is located after downloading.



For PC
https://www.icpdas.com/en/product/guide+Software+Development__Tools+PAC__SDK

XPAC, iPPC (WES7)
For XP-8000-WES7, XP-9000-WES7, iPPC (WES7) https://www.icpdas.com/en/download/show.php?num=2540

When using the following development platform **- XPAC (CE6), WinPAC (CE5), or WinPAC/ViewPAC (CE7)**, visit the following webpages to download and install files.

Development Tools: Visual Studio 2008 Professional Edition or earlier

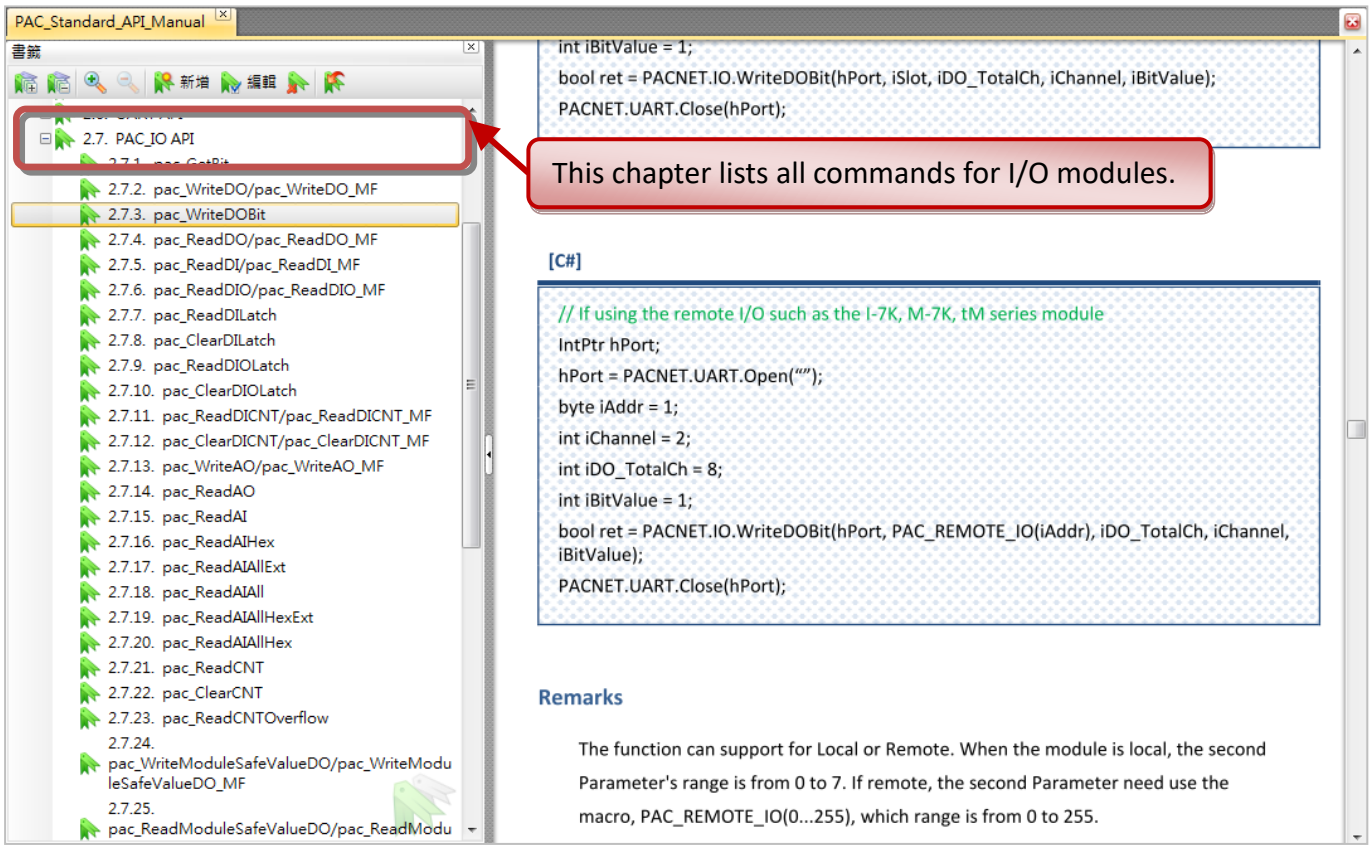
XPAC (CE6)
For XP-8000-CE6 https://www.icpdas.com/en/download/show.php?num=2473
WinPAC (CE5)
WinCE5.0 PACs/ViewPACs without I/O Slot(s) https://www.icpdas.com/en/download/show.php?num=2594 WinCE5.0 PACs/ViewPACs with I/O Slot(s) https://www.icpdas.com/en/download/show.php?num=2593
WinPAC/ViewPAC (CE7)
WinCE7.0 PACs/ViewPACs without I/O Slot(s) https://www.icpdas.com/en/download/show.php?num=2409 WinCE7.0 PACs/ViewPACs with I/O Slot(s) https://www.icpdas.com/en/download/show.php?num=2348

2) PAC API Manual:

The user manual describes the way to install the PACSDK, set up the development environment, and use the **PAC_IO** API functions.
(pac_standard_api_manual_x.x.x.pdf)

Download the user manual according to the development platform at the following path.

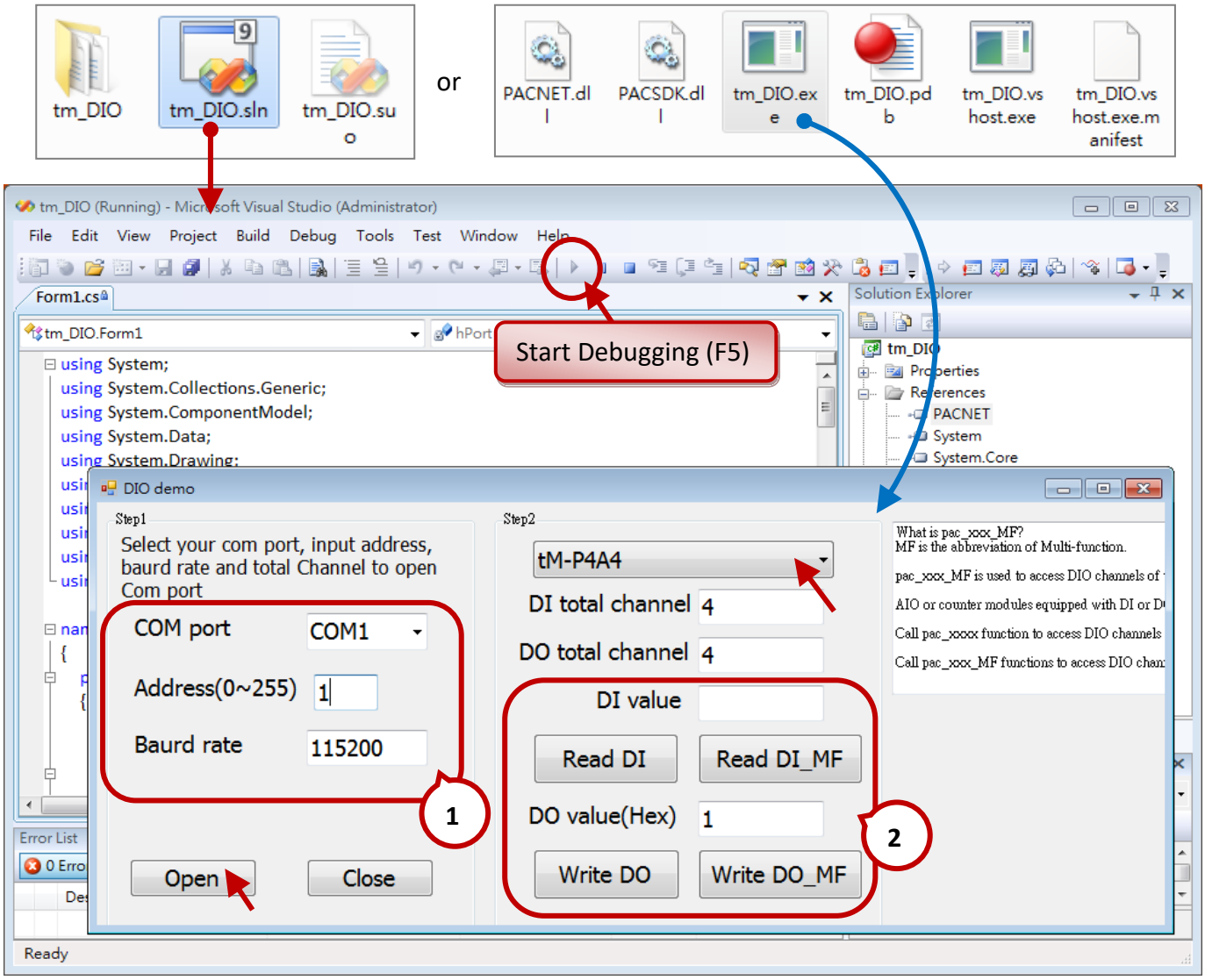
PC
https://www.icpdas.com/en/download/show.php?num=1049
PAC (WES)
https://www.icpdas.com/en/download/show.php?num=2527
PAC (WinCE)
https://www.icpdas.com/en/download/show.php?num=2407



3) C# demo program (for DCON protocol):

www.icpdas.com/en/download/show.php?num=2876

In the case of a DIO demo, refer to Section 2.4 to complete the hardware connection, and double-click the tm_DIO.sln to open the project file, and then click the "Start Debugging" button. Alternatively, double-click the tm_DIO.exe at the "…\tm_DIO\bin\Debug" path.



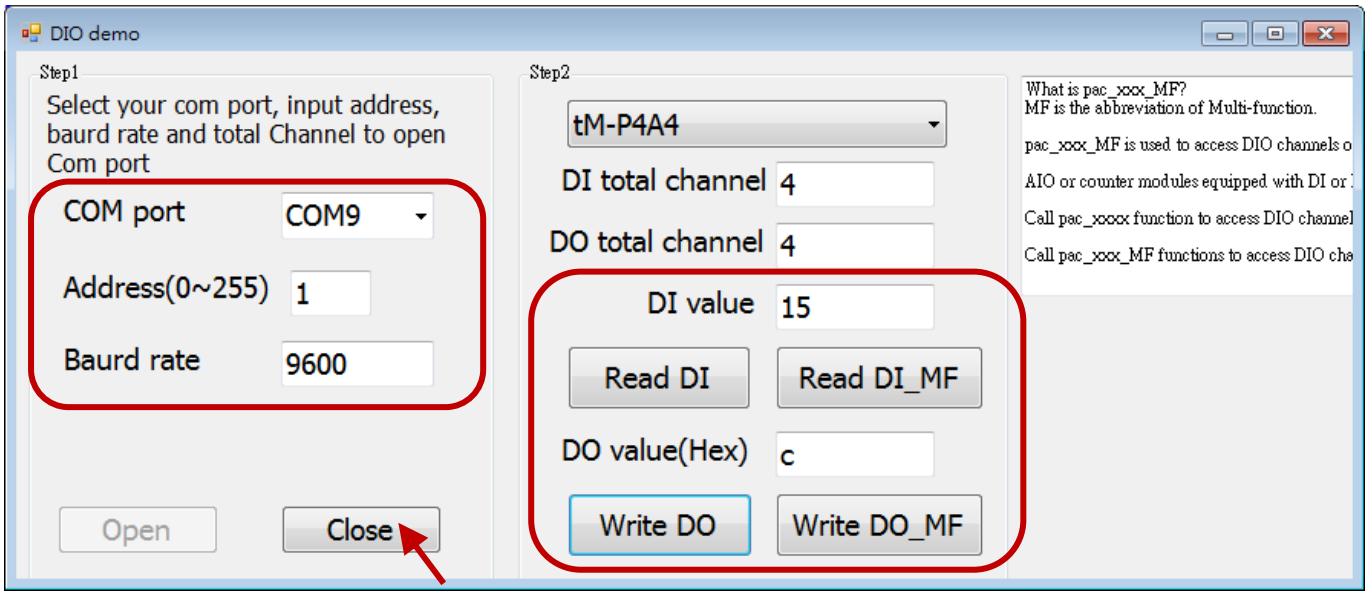
Step 1: Choose the COM Port, enter the address (Slave id) and baud rate for the tM module, and click the Open button to open the COM port. In this example, COM Port = COM9, Address = 1 and Baud rate = 9600. If you do not know the settings of the module, refer to Chapter3 to configure them by using DCON Utility Pro. **Note:** The Protocol setting of the module must be set as DCON.

Step 2: Choose the module name (e.g., tM-P4A4) and then the number of I/O channels will automatically be displayed in the "DI (or DO) total channel" field. Also, click the "Read DI" button to read the DI value, or enter a hexadecimal value (e.g., "C" = 1101) in the "DO value (Hex)" field and click the "Write DO" button to write a DO value. **Note:** When using a multi-function I/O (e.g., tM-AD4P2C2), replaced to click the "Read DI_MF" or "Write DO_MF" button to access data.

After completing the previous steps, the screen will be displayed as below:

DI value = 15 $_{(10)}$ = 1111, indicates the status of DI3, DI2, DI1, and DI0 are ON.

DO value = c $_{(16)}$ = 1101, indicates the status of DI3, DI2, and DI0 is ON, and the status of DI1 is OFF.



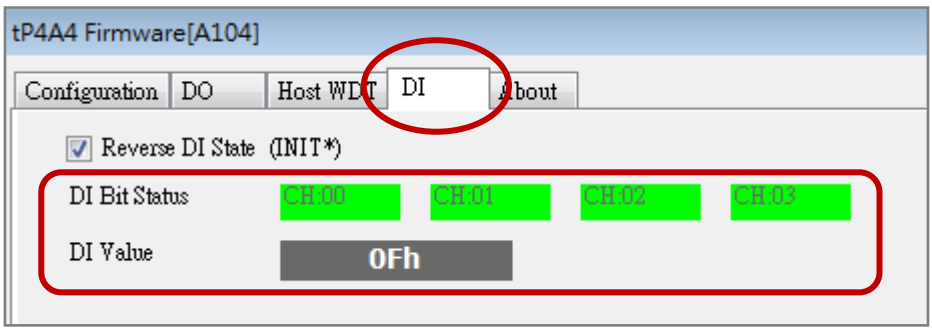
Step 3: Click the "Close" button to release the COM port.

Users can see the change of LED indicators on the module, or observe the DI or DO status by using DCON Utility Pro. Note that the COM port in use for the demo must be released.

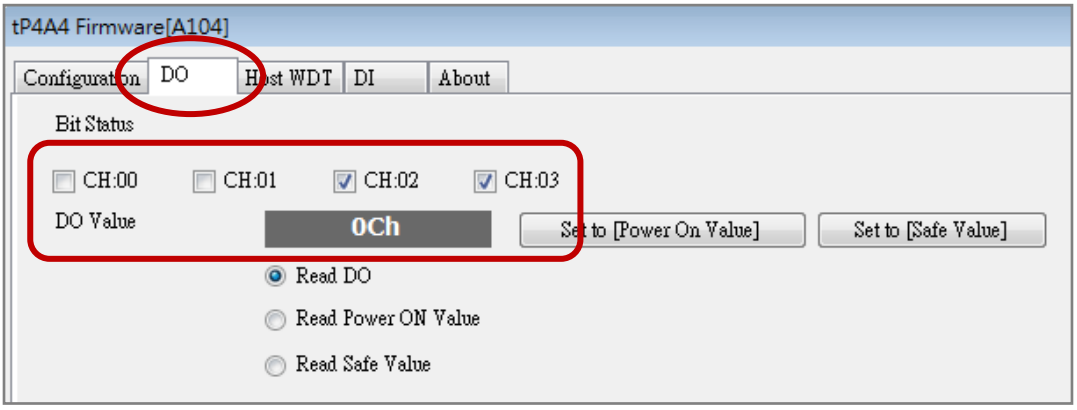
Step 1: Search for the I/O module. (Reference: Section 3.1)

Step 2: Double-click the name of the module to open the configuration page, and then click the DI or DO tab to see the results.

[The DI Page]:



[The DO Page]:



6.3 Modbus Development Tools and SCADA

ICP DAS provides a variety of software tools and SCADA that supports Modbus RTU/ASCII protocol, and helpful for users to develop their applications.

6.3.1 Software Tool – OPC DA Server and UA Series Product

NAPOPC DA Server is a free **OPC DA Server** for ICP DAS products that can be used to integrate data from ICP DAS remote I/O module, PAC I/O, and other equipment. Also, NAPOPC DA Server allows the SCADA/HMI/Database software that supports OPC DA Client, to access I/O data to achieve the process control and manufacturing automation applications.

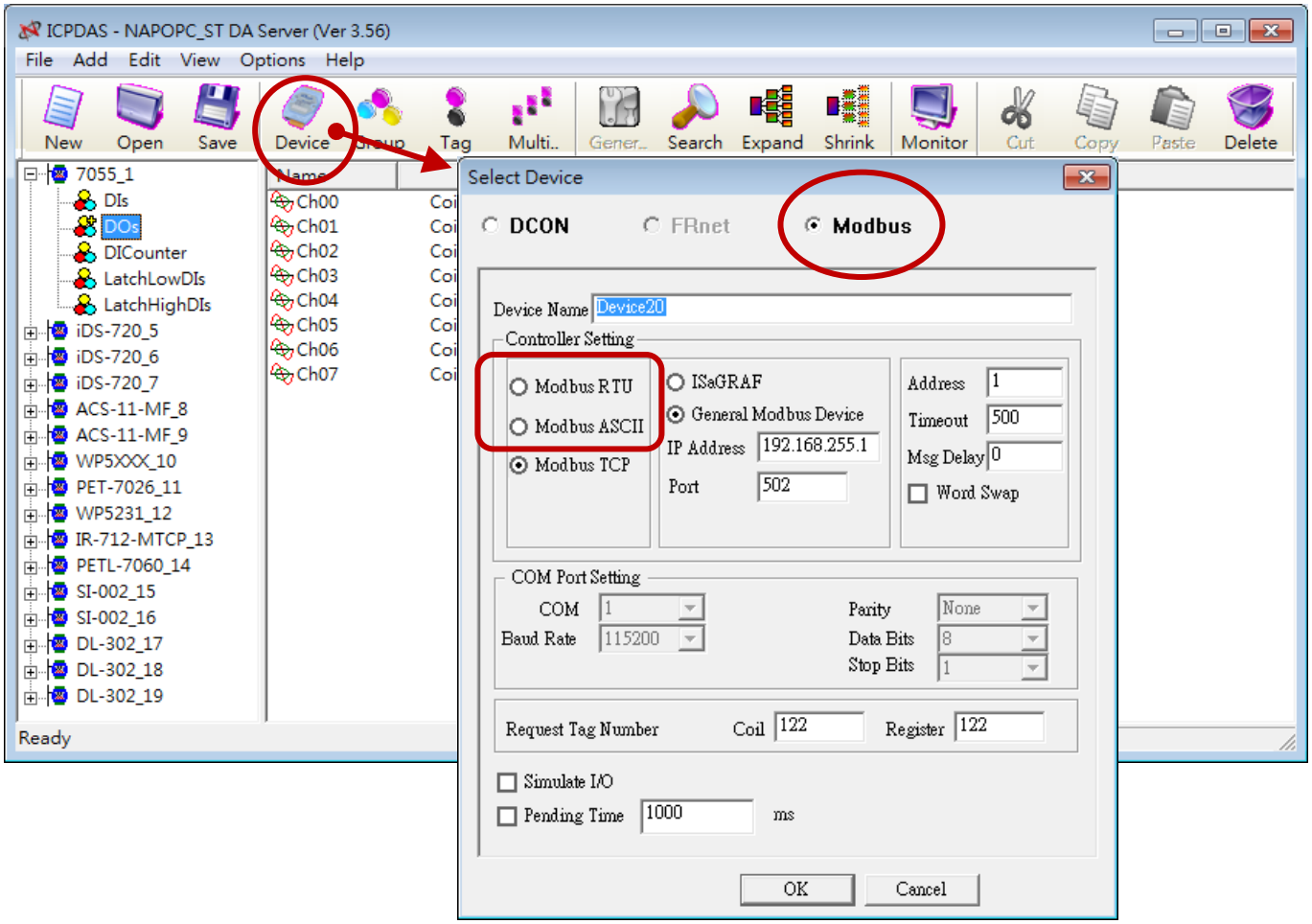
Also, users can choose an optional license of the Modbus protocol for the NAPOPC_ST DA Server (PC) to work with the third-party Modbus equipment.

Website / Data Sheet/ DM:

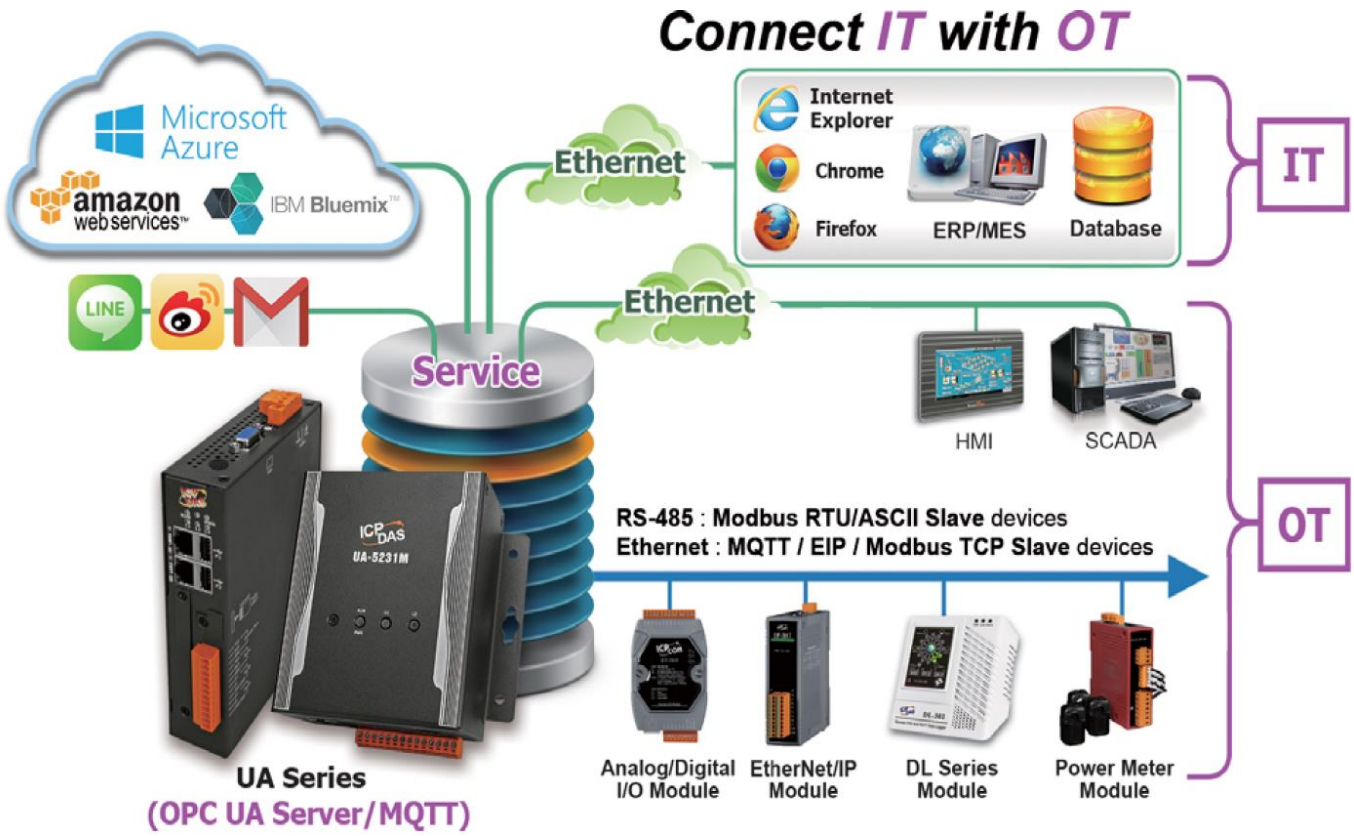
https://opc.icpdas.com/opcda.htm

https://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/napopcsvr/data_sheet/napopc_st_data_sheet_e.pdf

https://www.icpdas.com/root/support/catalog/pdf/Brochure/NAPOPC/NAPOPC-DM-e.pdf



UA Series IIoT Communication Server

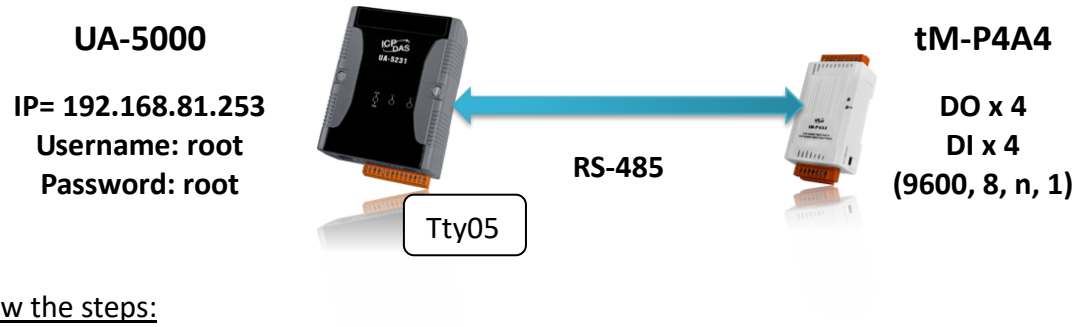


UA series is a series of IIoT (Industrial IoT) Communication Server for integrating the system and devices of IT and OT. UA series feature the IIoT Gateway function that allows users to access remote I/O modules and controllers via Modbus TCP/ RTU/ASCII, MQTT, and EtherNet/IP communication protocols. IIoT gateway function can also convert these I/O data to conform to OPC UA or MQTT protocols for implementing in the applications of MES, ERP, SCADA, or Cloud service. Besides, UA series feature the Data Logger function that allows users to write I/O data directly into the remote database and save it to the local file (.CSV) as the historical records.

UA series supports uploading I/O data to the IoT Cloud platform such as Amazon AWS and Windows Azure. It can also connect to the Cloud logic service platform “IFTTT” which supports many web APPs for users to receive first-hand notification messages through the most commonly used mobile APPs when an event is triggered. UA series enhance the networking and interoperability between IT and OT. Through UA series, users can easily deploy Industrial IoT.

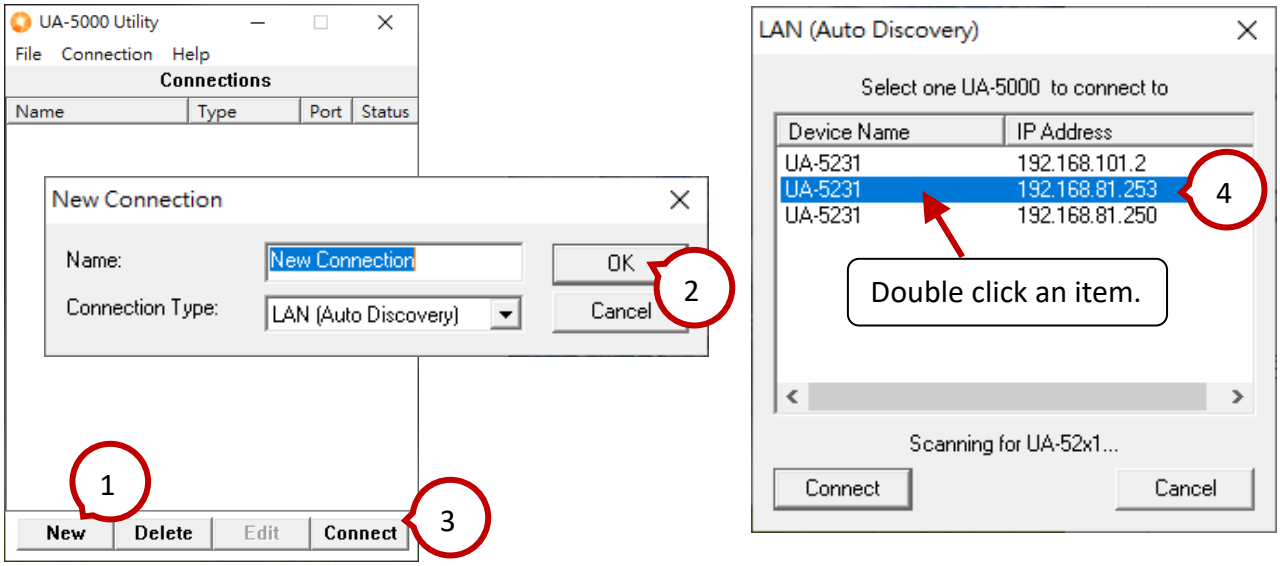
Visit the ICP DAS UA Series website for more information.
https://www.icpdas.com/en/product/guide+IIoT+Controller_Server+Communication__Server

In the example, using a UA series to connect to a tM-P4A4 module through an RS-485 port (i.e., Tty05) to read/write Modbus RTU I/O data. Also, the Modbus RTU to OPC UA data conversion function is added.

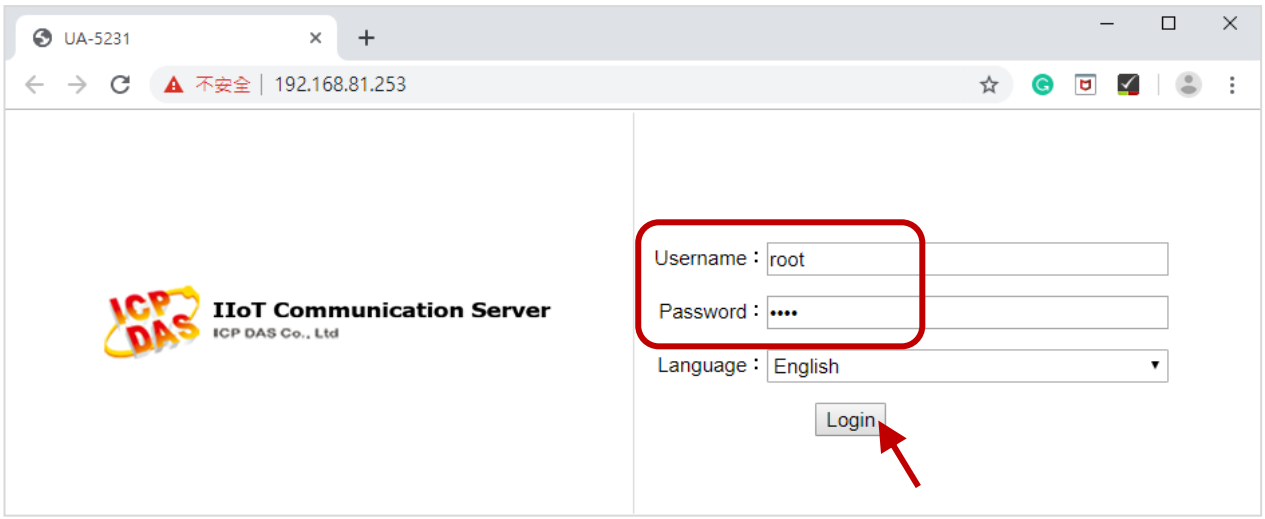


Follow the steps:

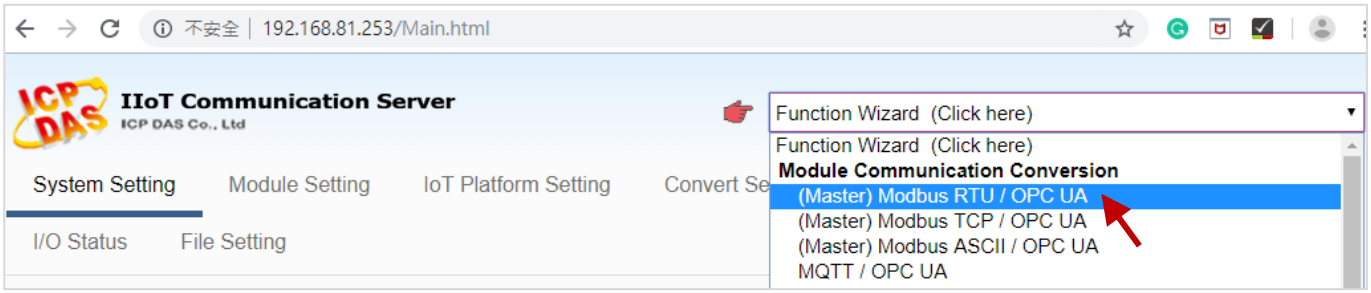
1. Log in to the configuration web page of a UA-5000 series.
   If the IP address of UA-5000 series is unknown, downloading the UA-Series Utility and searching the UA device on the web.
   https://www.icpdas.com/en/download/index.php?nation=US&kind1=&model=&kw=ua-



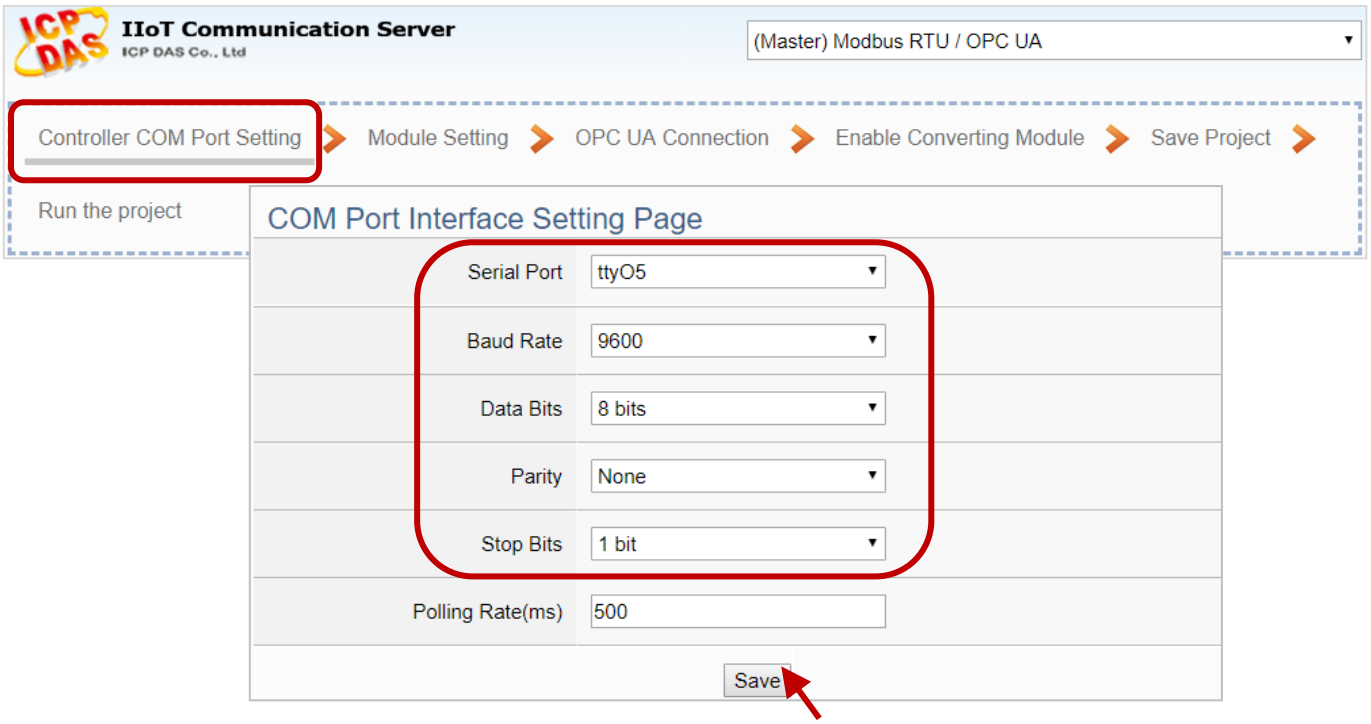
Double-click a UA-5000 list item, and the web browser will automatically be displayed. Enter the Username and Password (root) and click the Login button to log into the configuration page.



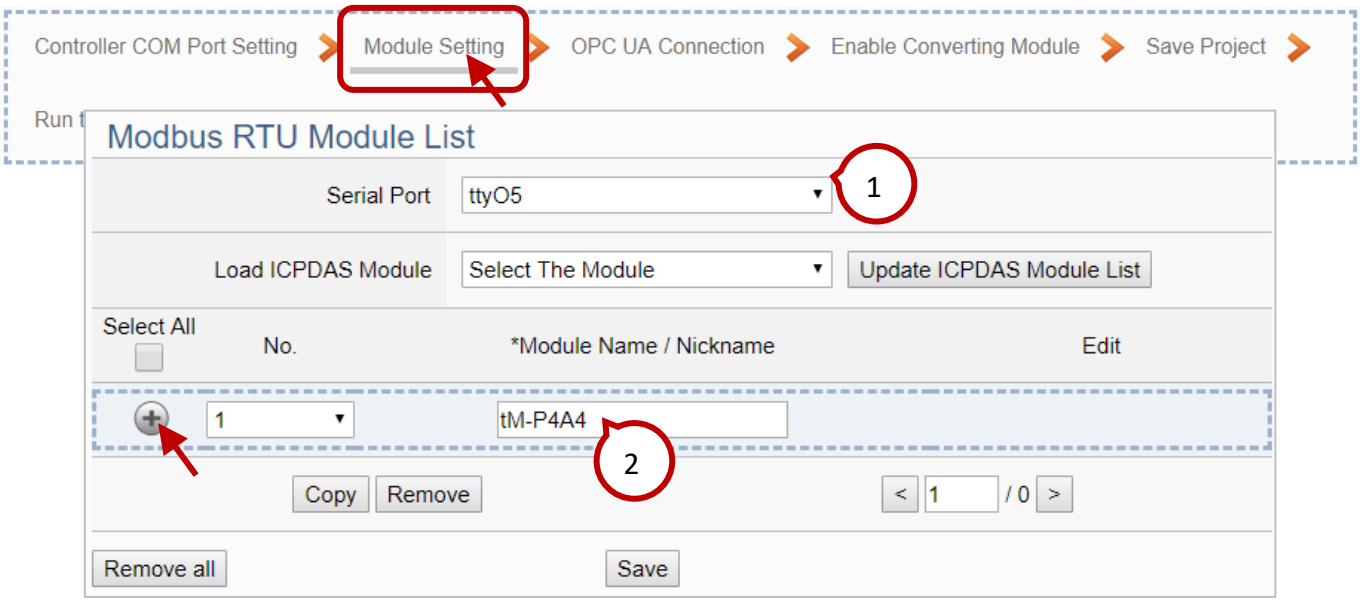
2. Choose “(Master) Modbus RTU/OPC UA” to add the Modbus RTU to the OPC UA conversion function.



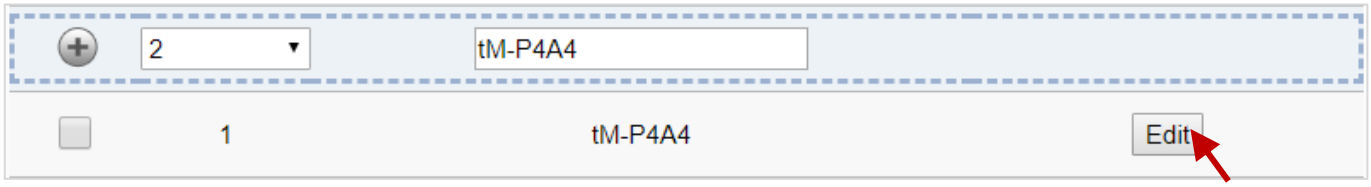
3. Set the COM Port (e.g., Tty05) and communication parameters (9600, 8, n, 1) for the UA-5231, and then click Save.



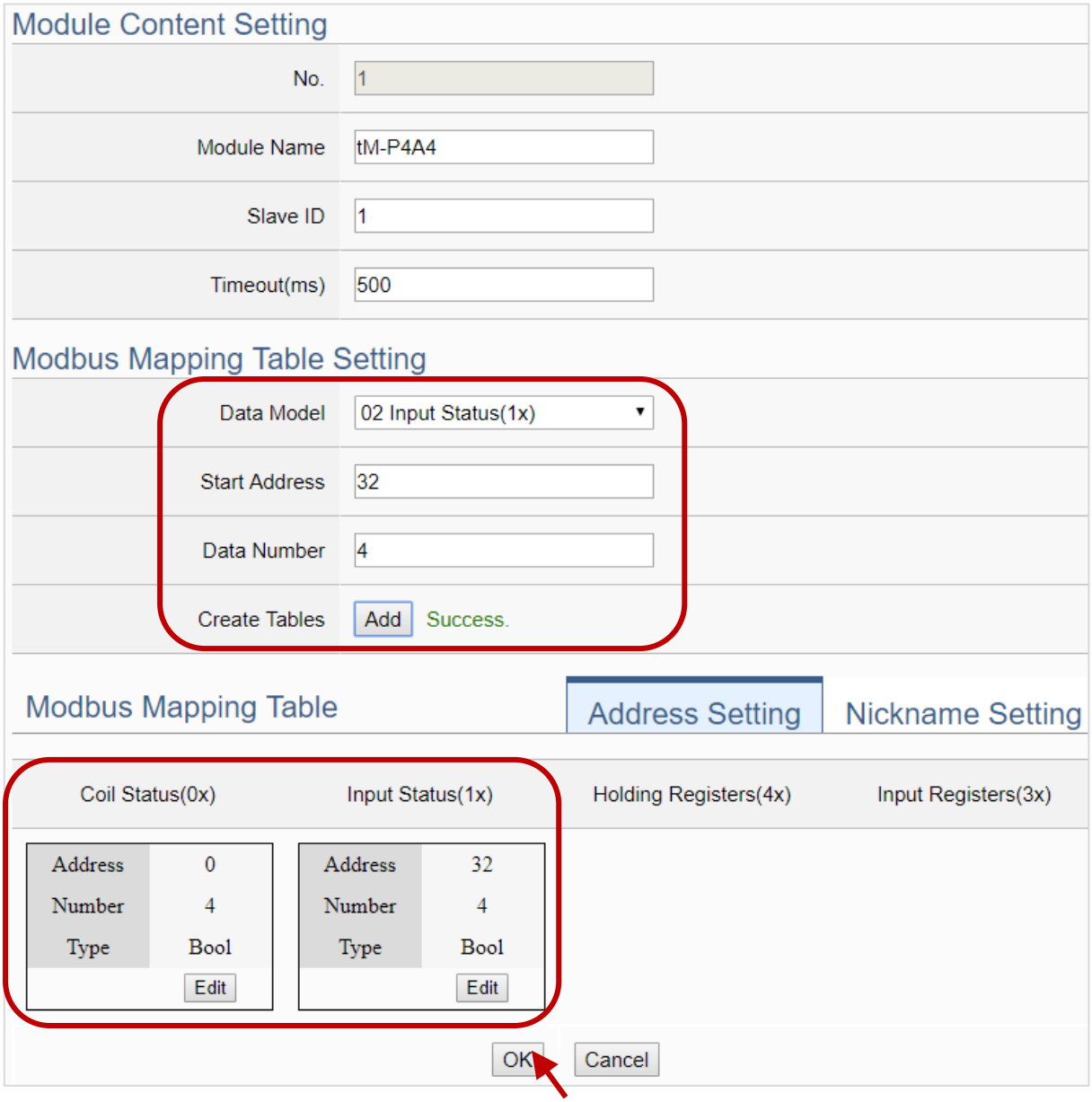
4. Click “Module Setting” to set the COM Port and enter a module name, and then click the “+” button to add the module setting.



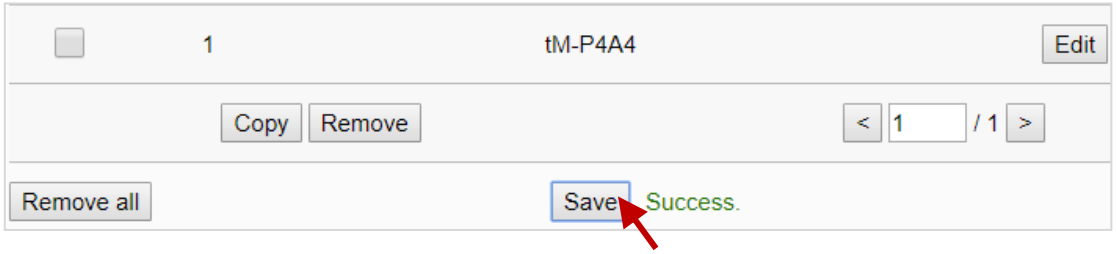
Next, click the Edit button to display the configuration page.



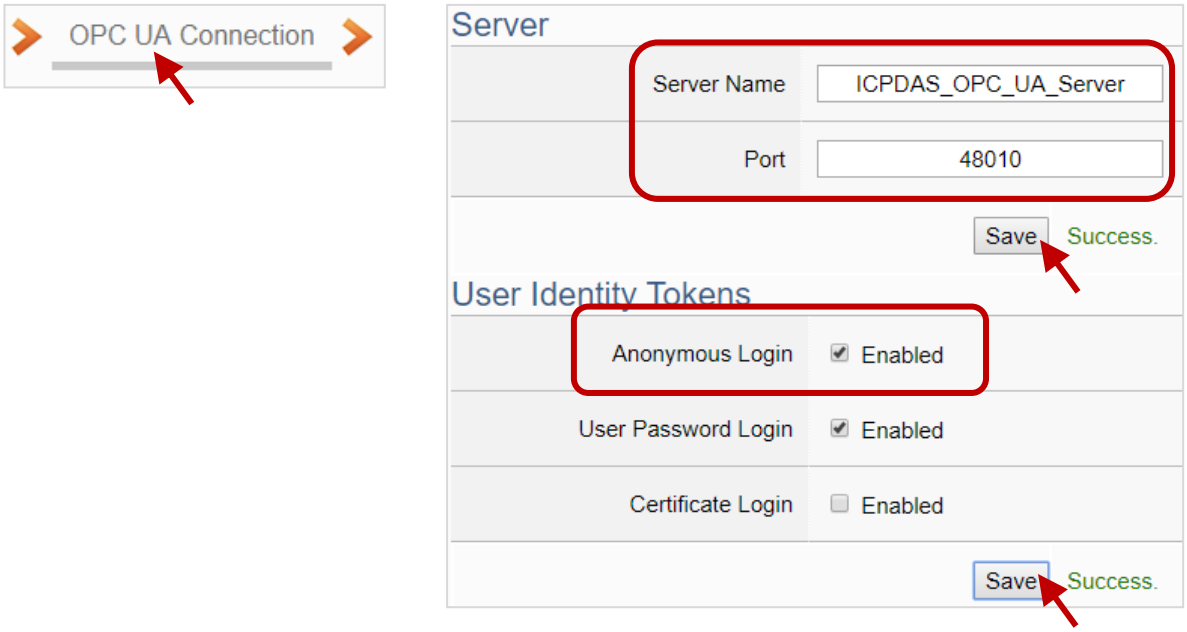
5. The tM-P4A4 module (4 DI and 4DO) is used in this example, refer to Appendix C to look up the Modbus address. After configuring the Data Model, Start Address, and Data Number settings, click the Add button to add the Modbus Mapping Table and click OK.



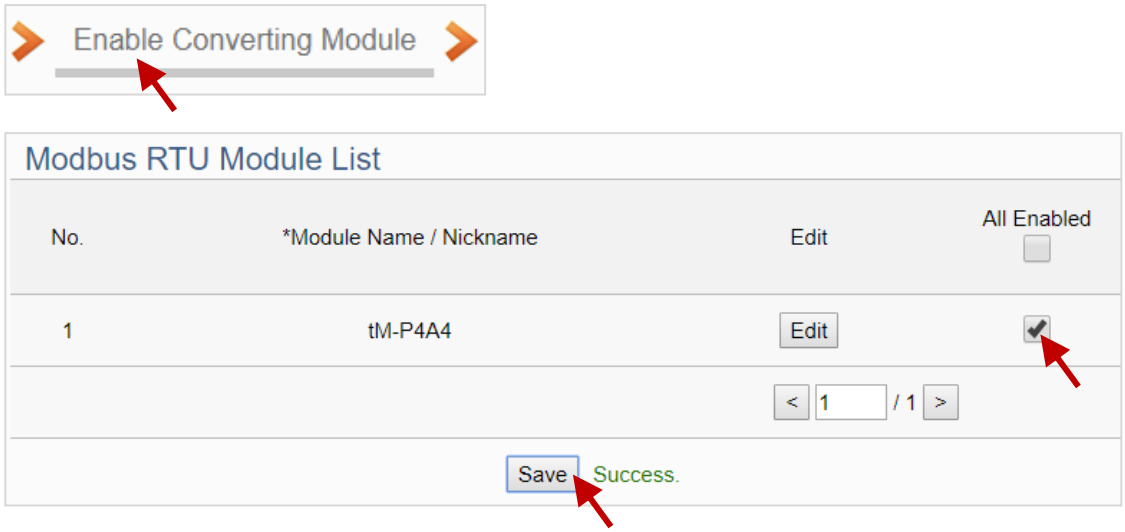
Then, it will go back to the Modbus RTU Master List page. Click Save to save the settings.



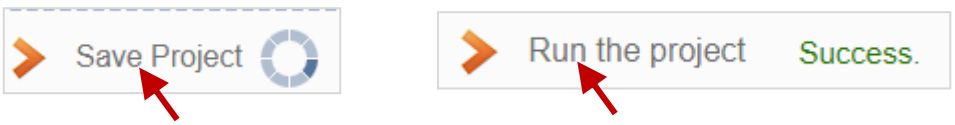
6. Click **OPC UA Connection** to display the Local Server page. By default, the Anonymous Login function is enabled. Just click both two Save buttons.



7. Click **Enable Converting Module** and check the Module item to enable the data conversion function, and then click Save.



8. Click **Save Project** to save the project and click **Run the Project** to update and execute the new project. Now, UA-5231 can conduct the data conversion for the tM module.



Visit the website for more information about UA applications.
https://www.icpdas.com/en/product/guide+IIoT+Controller_Server+Communication__Server

6.3.2 Software Tool – nModbus

nModbus is a C# 3.0 implementation of the Modbus protocol. It is developed and maintained voluntarily and provided free of charge. ICP DAS verified and improved the DLL based on the official released NModbus_net-2.0_1.11.0.0-source.zip.

Programmers can use the DLL released by ICP DAS to develop a Modbus application for regular Windows-based PCs or WinCE based devices.

Download the DLL, nModbus API manual, and demos on the nModbus website:
https://www.icpdas.com/en/product/guide+Software+Development__Tools+Modbus__Tool

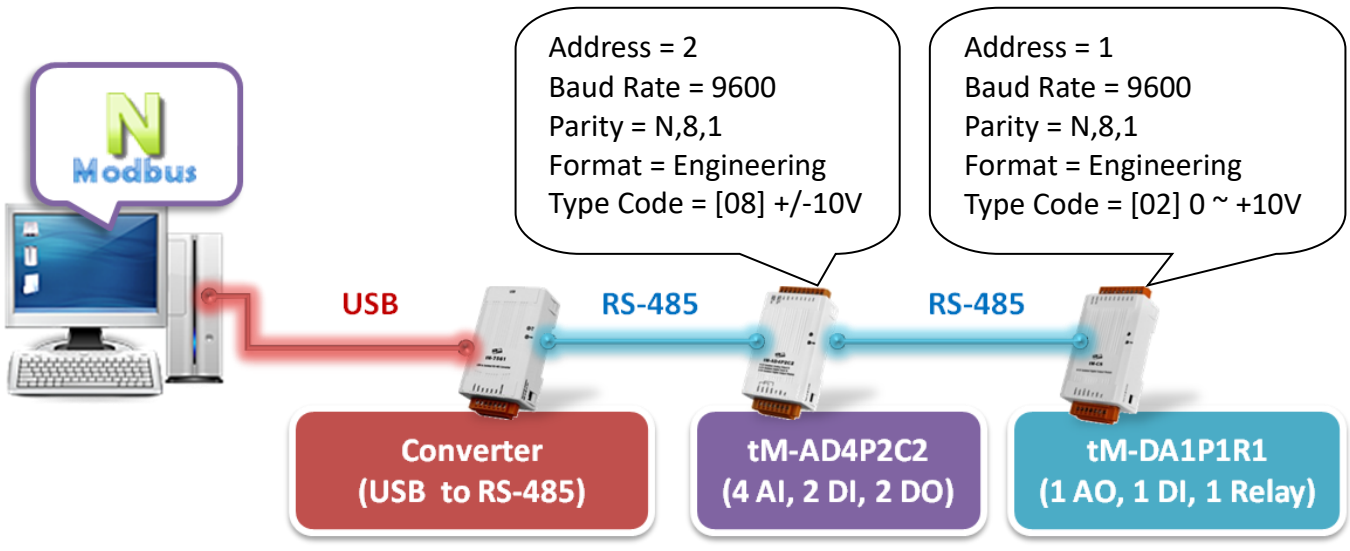
nModbus API Manual:
https://www.icpdas.com/en/download/show.php?num=1024

Demo:
https://www.icpdas.com/en/download/show.php?num=1025

- **PC 版:**
  - **DLL**: log4net.dll, nmodbuspc.dll
- **WinCE 版:**
  - **DLL**: cabc.dll, fc19.dll, nmodbusce.dll

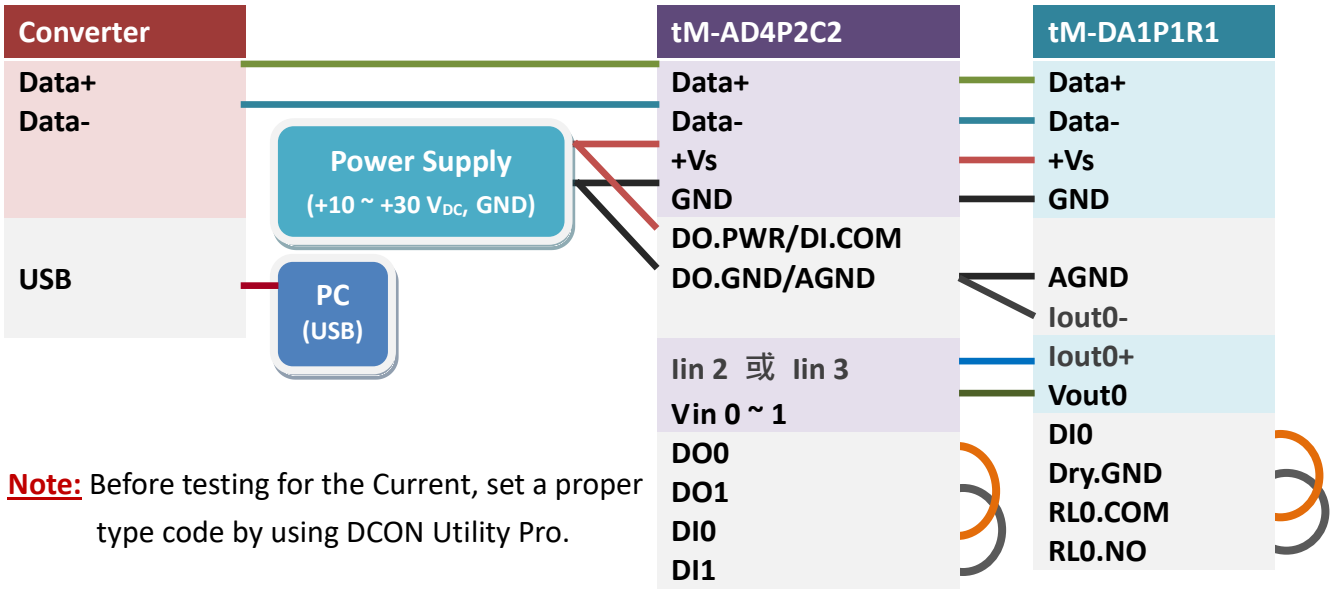
Download the Modbus RTU Master test demo that supports tM series module.
www.icpdas.com/en/download/show.php?num=2876 (C# Demo for Modbus Protocol)

Before using the module, configure them by using DCON Utility Pro. (Reference: Chapter3)



Hardware Wiring for this example:

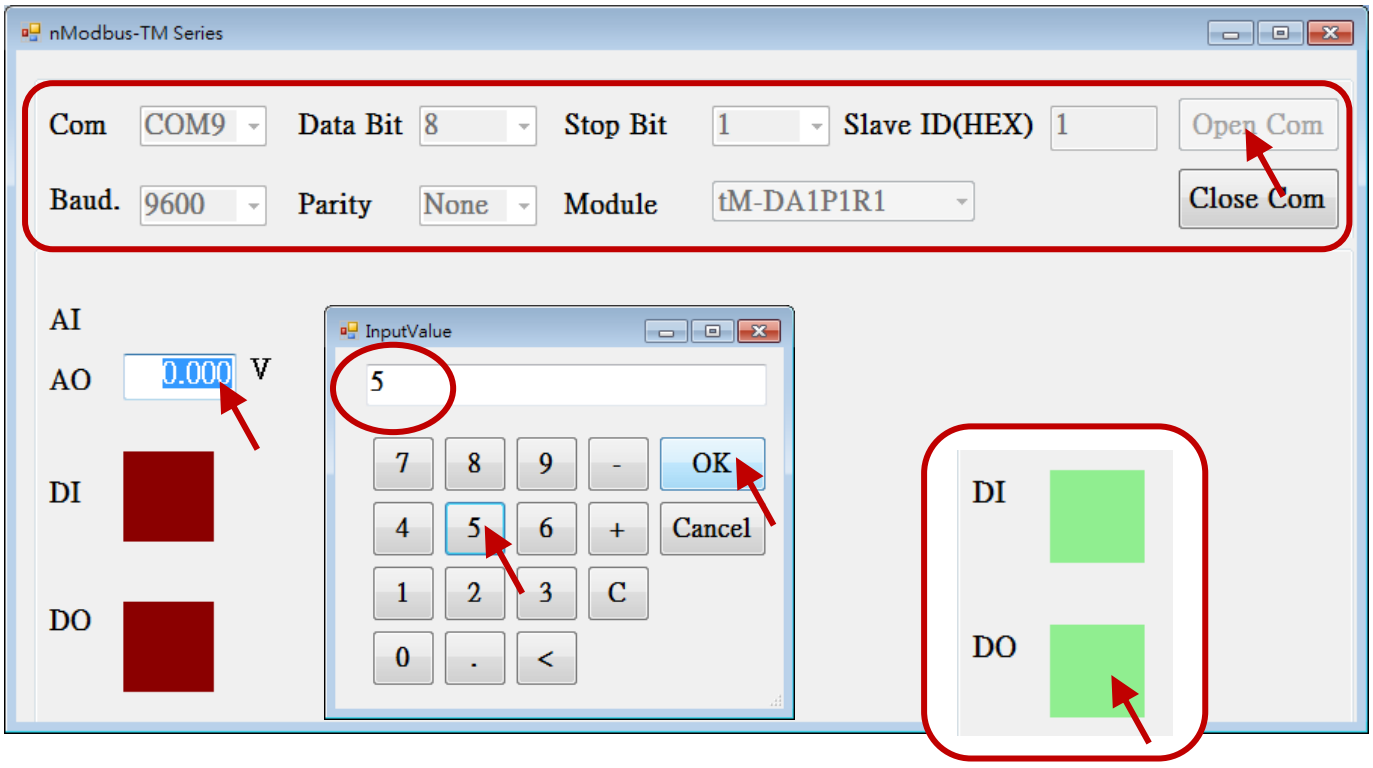
In this case, using tM-AD4P2C2 and tM-DA1P1R1 modules to test the DI/DO/AI/AO status. First, connect all devices properly as follows, refer to Section 2.3 to know the way of the hardware wiring.



Note: Before testing for the Current, set a proper type code by using DCON Utility Pro.

Double click “modbusrtu_master_tm.exe” in the path “…\modbusrtu_master_tm\bin\Debug” on PC. Next, enter the communication parameters of a module, and click the “Open Com” button to connect.

Enter an AO value for the tM-DA1P1R1, and see the change of AI0 and AI1 values for the tM-AD4P2C2. Also, click the DO button and see the change of DI status.



The current AI0 and AI1 values of the tM-AD4P2C2 are 4.957V. Next, click the DO button to change the DI status.

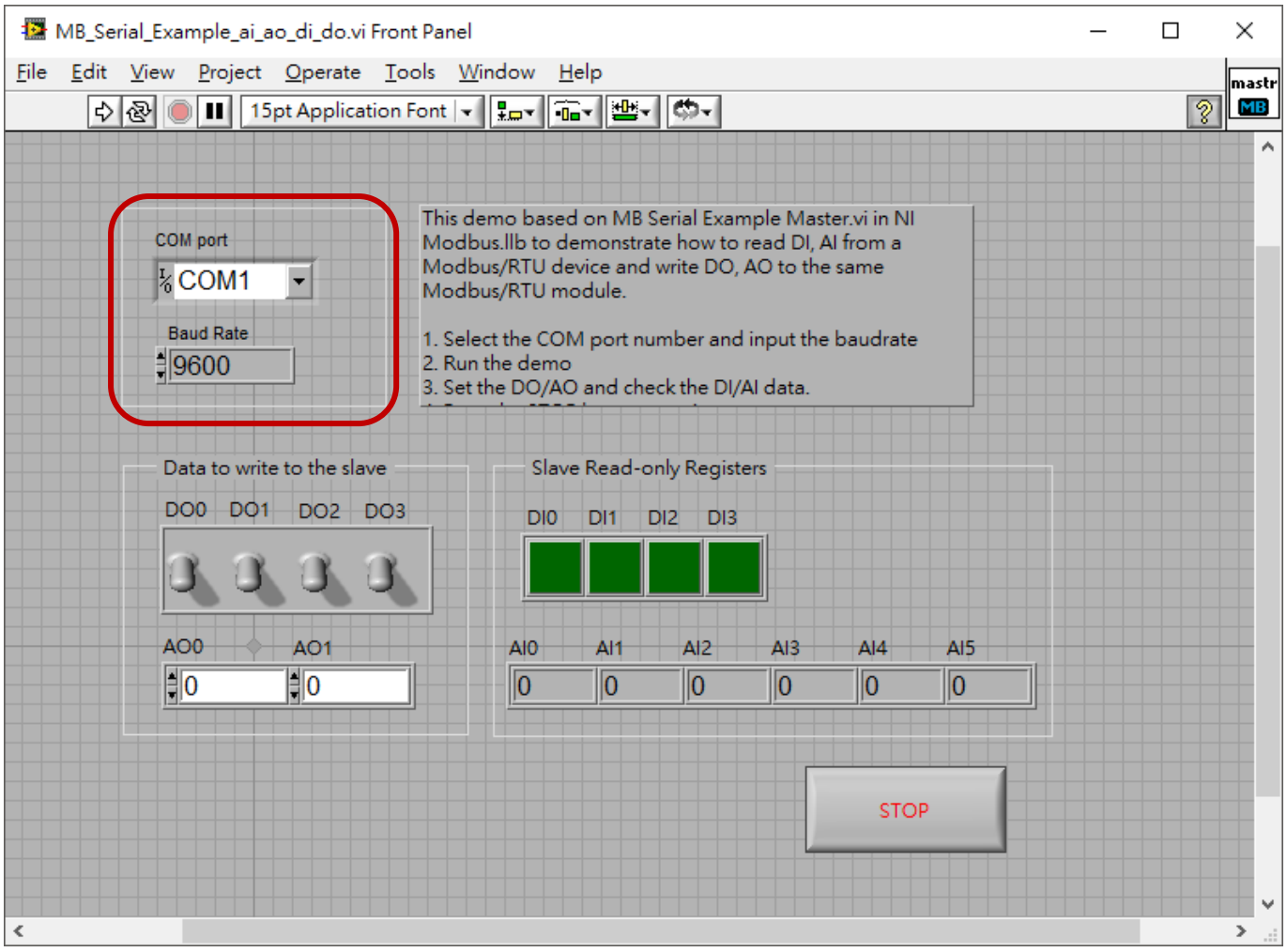
6.3.3 Software Tool – LabVIEW

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is a system-design platform and development environment for a visual programming language from National Instruments. LabVIEW provides an easy-to-use graphical interface and supports a variety of hardware drivers and software analysis tools that help users to speed up the amount of time to develop applications. LabVIEW has been widely used for the test, measurement, and automated control in various laboratories or industries.

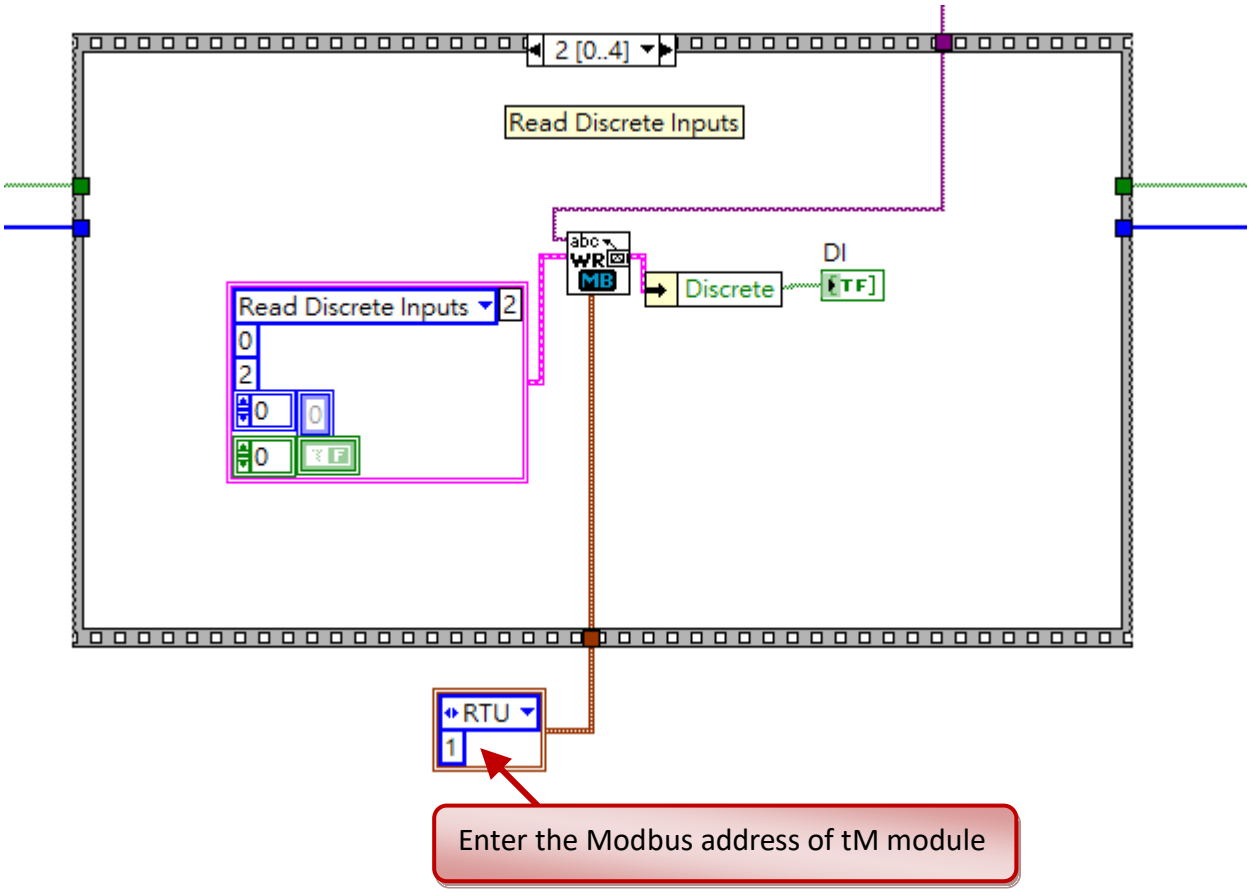
Download the Modbus RTU demo programs at the link, and then connect the tM module.
https://www.icpdas.com/en/product/guide+Software+Development__Tools+LabVIEW__Tools

Note: Before communicating with the tM module, refer to Chapter3: Software tool - DCON Utility Pro to configure the module, and then confirm the COM port number and both the Modbus address and baud rate settings of the module.

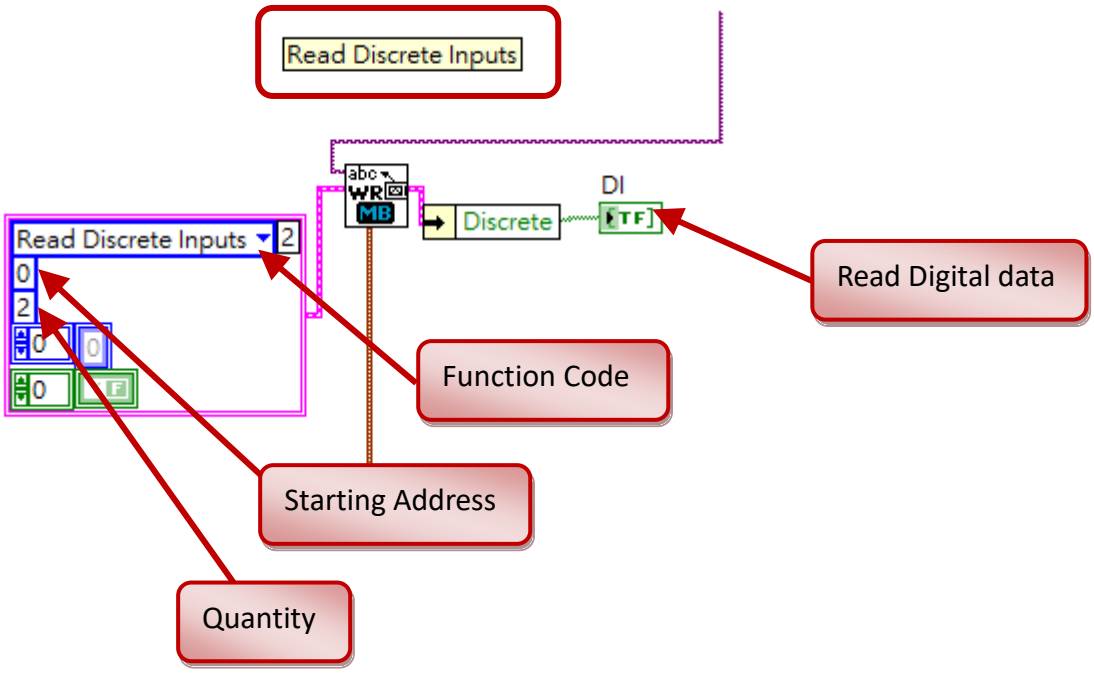
Step 1: Download the ModbusRTU.zip and unzip the file.
Step 2: Choose a demo program for the I/O module that is connected with PC (e.g., MB_Serial_Example_ai_ao_di_do.vi), and set the proper COM port number and baud rate.



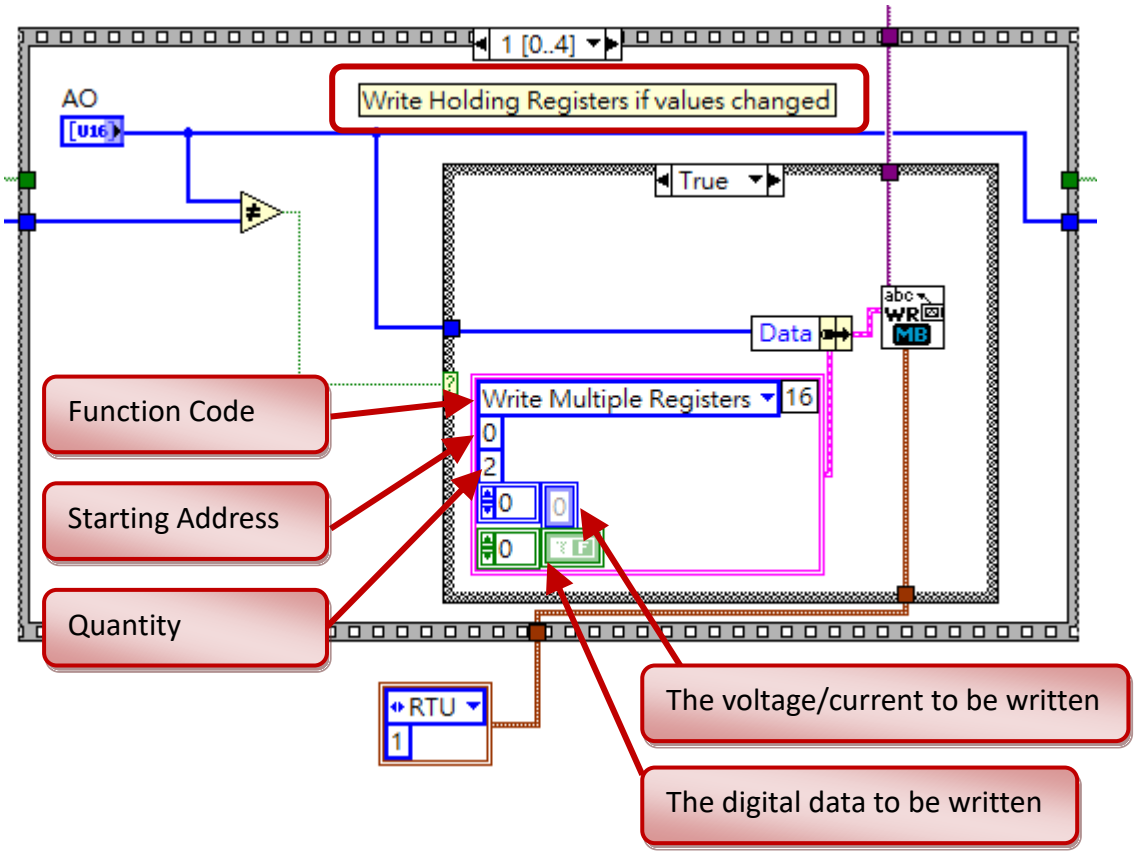
Step 3: Enter the Modbus address of the module.



Step 4: To read data from the module, set the function code, starting address, and the length of data.



Step 5: To write data to the module, set the function code, starting address, the length of data, and the data to be written.



Description of Modbus Function Codes

Function Code	Description		I/O Address
01	Read Coils	(Read D O data)	0xxxx
02	Read Discrete Inputs	(Read DI data)	1xxxx
03	Read Holding Registers	(Read AO data)	4xxxx
04	Read Input Registers	(Read AI data)	3xxxx
05	Write Single Coil	(Write single DO data)	0xxxx
06	Write Single Register	(Write single AO data)	4xxxx
15	Write Multiple Coils	(Write multiple DO data)	0xxxx
16	Write Multiple Registers	(Write multiple AO data)	4xxxx

6.3.4 Software Tool – InduSoft

InduSoft Web Studio is a powerful, integrated collection of automation tools that includes all the building blocks needed to develop modern Human Machine Interfaces (HMI), Supervisory Control and Data Acquisition (SCADA) systems, and embedded instrumentation and control applications.

InduSoft Web Studio supports all Windows runtime platforms, ranging from 10 / 8 / 7 (32-bit/64-bit), Windows XP/ Vista, and Windows Server Editions, along with built-in support for local or remote (web) based visualization. InduSoft also conforms to industry standards such as Microsoft .NET, OPC, DDE, ODBC, XML, and ActiveX.

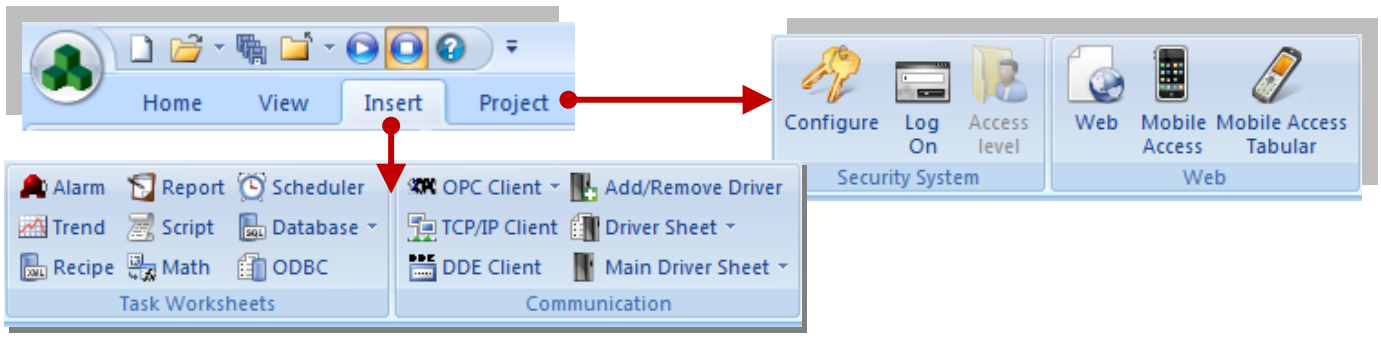
Notice:

ICP DAS Co., Ltd. has released InduSoft v8.1 license for Hardkey. Visit the website for the new features.

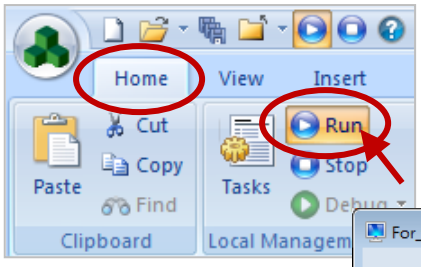
InduSoft website:
https://www.icpdas.com/en/product/guide+Software+InduSoft+InduSoft

Software Features:

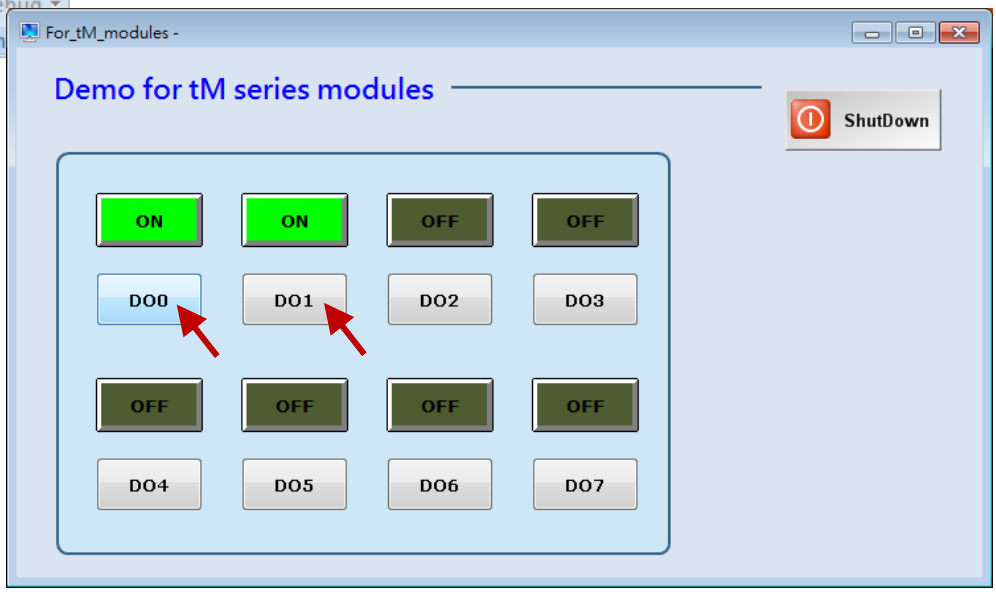
- ✧ Graphics and Animation design tools
- ✧ Connect to any SQL database (MS SQL, MySQL, Sybase, Oracle), MS Access, or Excel, and ERP/MES systems
- ✧ Alarms, Events, Trends, Recipes, and Reports management tools
- ✧ Provides over 240 native communication drivers
- ✧ Supports web server, database, and overall system redundancy
- ✧ Two powerful scripting languages are supported; built-in InduSoft functions and standard VBScript.
- ✧ Remotely view screens as web pages using IE browser or InduSoft Secure Viewer
- ✧ Compare any configuration file or merge changes from multiple developers.
- ✧ Monitoring and interacting with process values on any browser (e.g., iOS Safari, Google Chrome, etc.) that supports HTML5.



Description of the demo program:



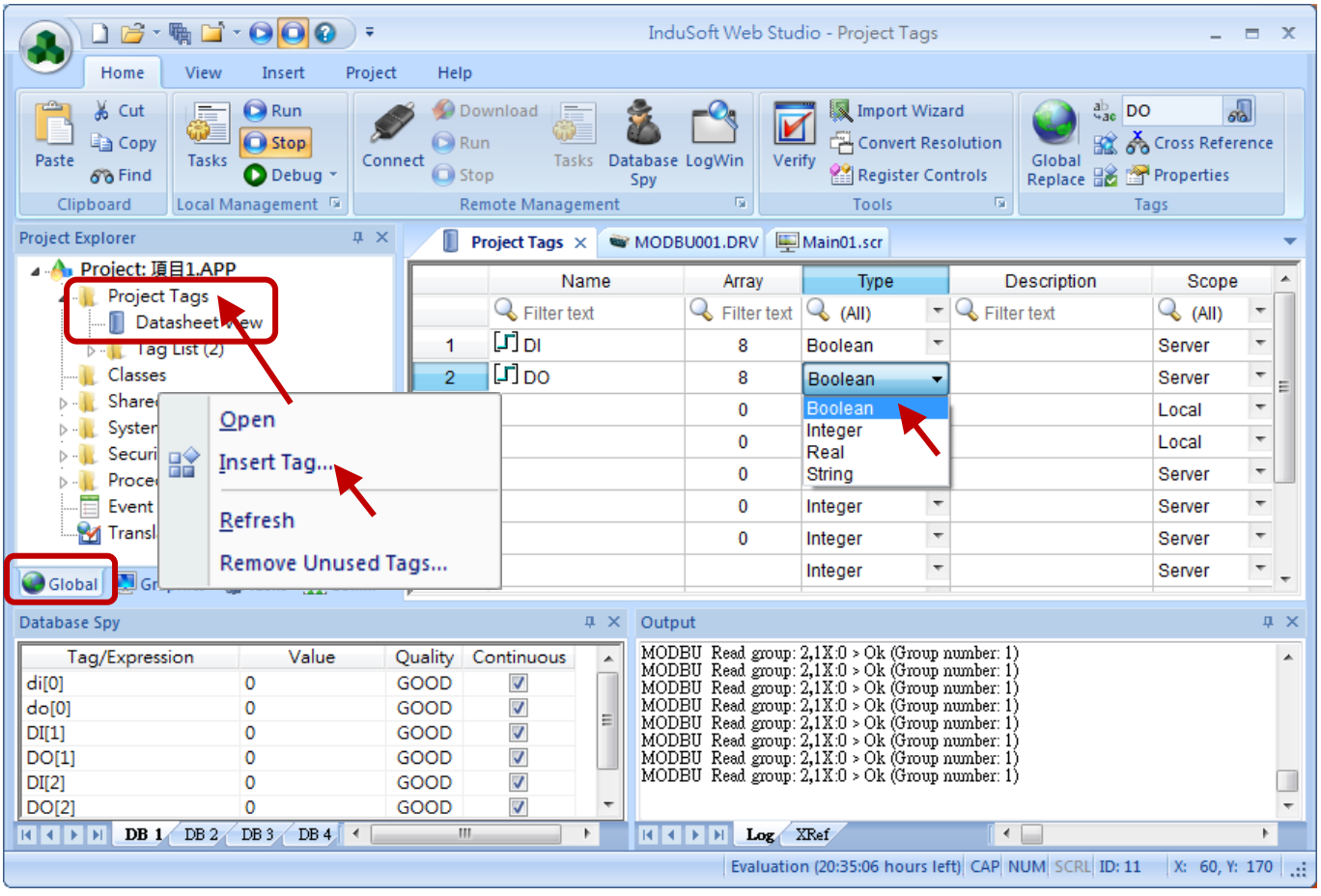
The demo program is used to read/write data from/to tM-P8 (8 DI) and tM-C8 (8 DO) modules through the Modbus RTU protocol. Refer to the InduSoft user manual for more details about the setting. First, click Run (F5) from the Home menu to execute the project.



Follow the steps below:

1. Add variables

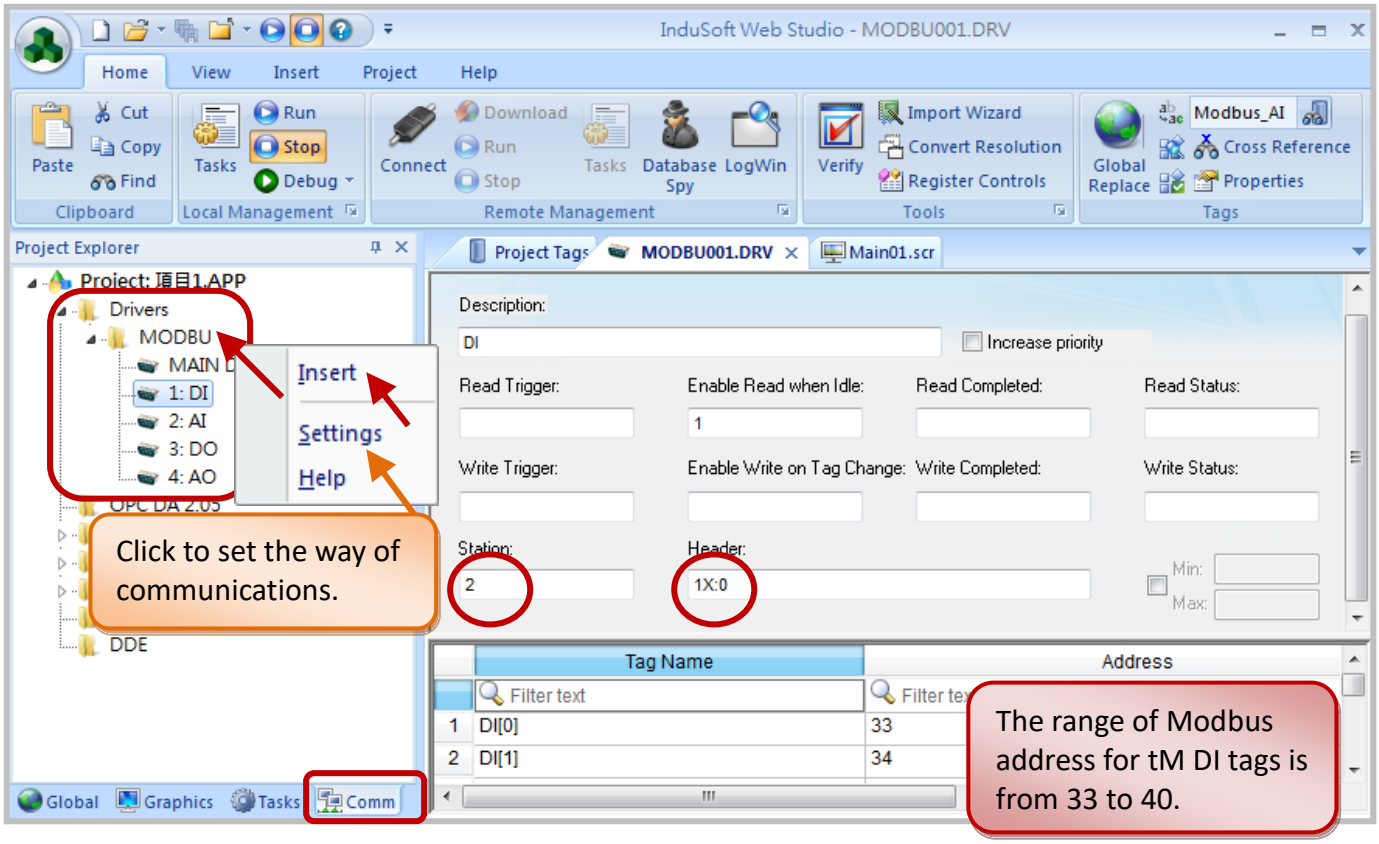
On Global tab, right-click on Project Tags and click Insert Tag to add the variable.



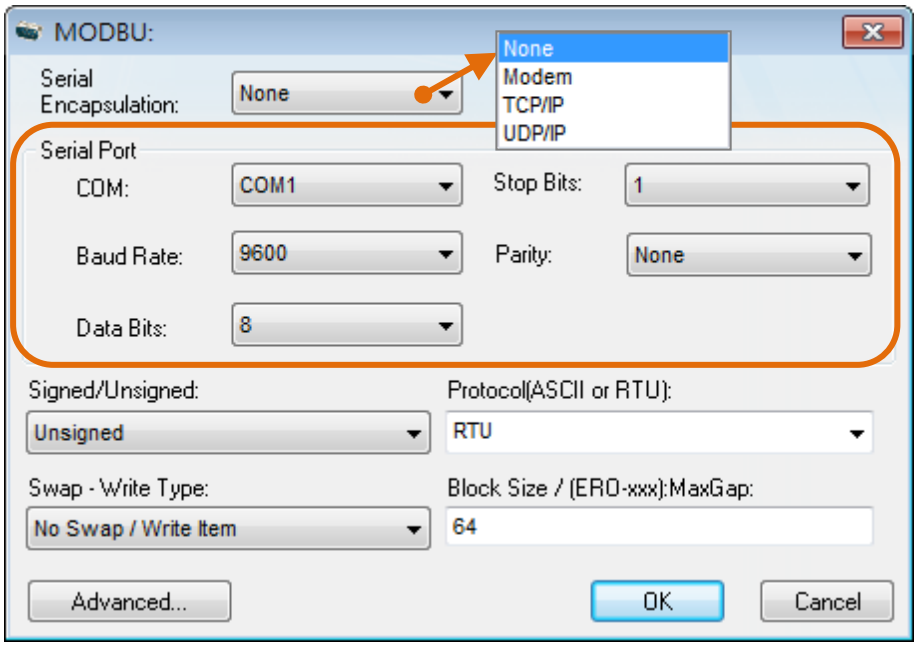
2. Configure the variable and the method of the communication

On the **Comm** tab, right-click MODBU and click Insert to configure the variable. In this example, the Net-ID of tM-P8 (DI) is “2”, and the Header is set to “1x: 0”; the Net-ID of tM-C8 (DO) is 3, and the Header is set to “0x: 0”.

Note: Using DCON Utility Pro to set communication parameters of the tM module before using them. In this example, Protocol = Modbus RTU, Address = “2” (for tM-P8) or “3” (for tM- C8), Baud Rate = 9600, Parity = N,8,1, and Checksum = Disable.

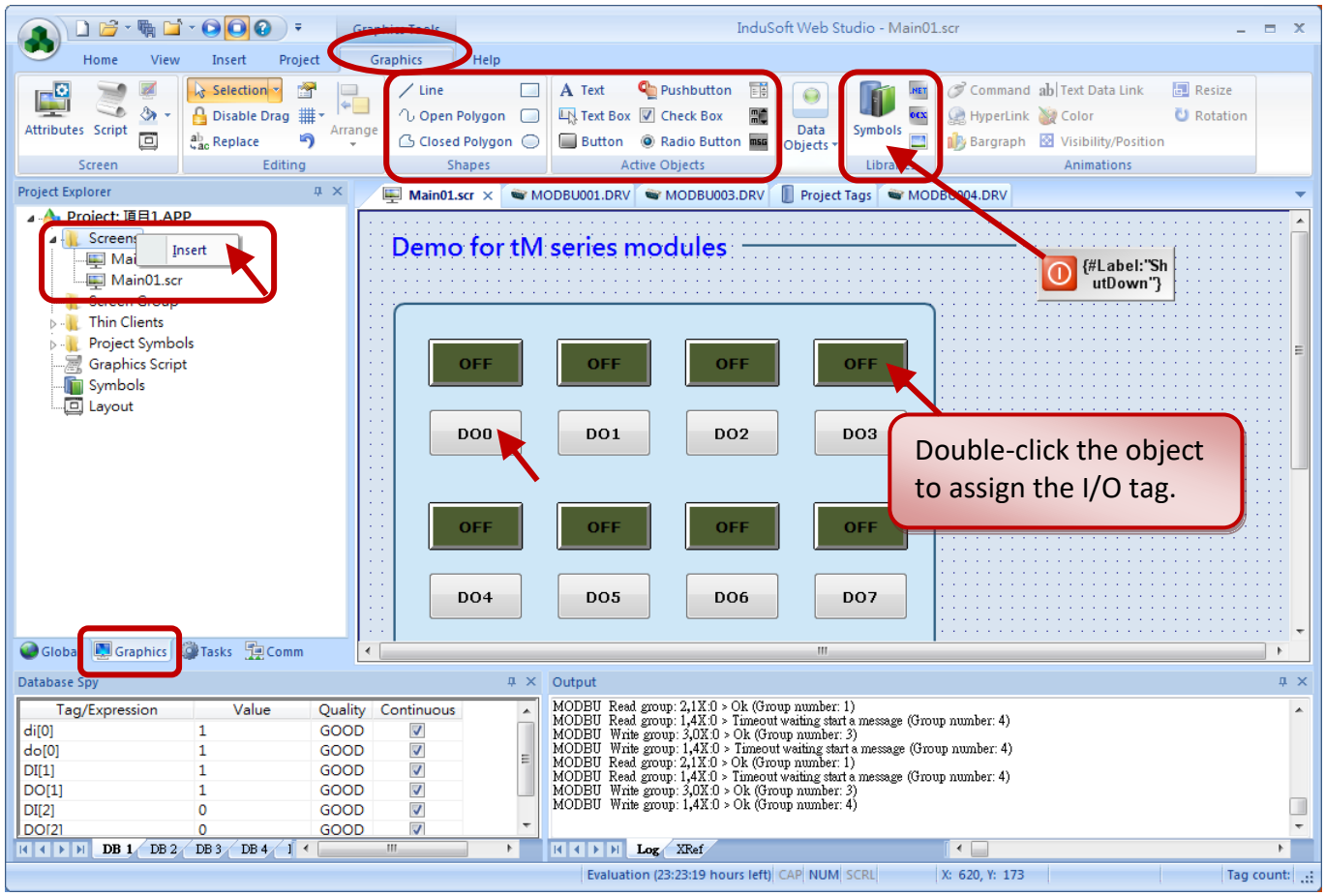


Also, click Settings to specify the communication method for InduSoft to communicate with the module. In this case, using COM1.

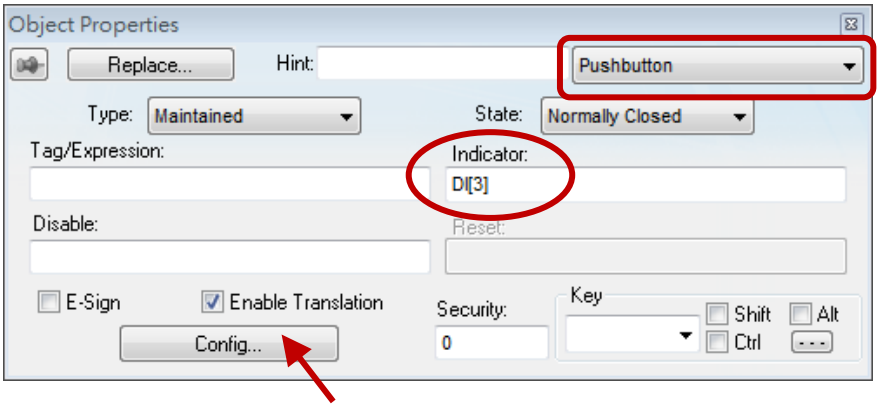


3. Design an HMI page and assign variables

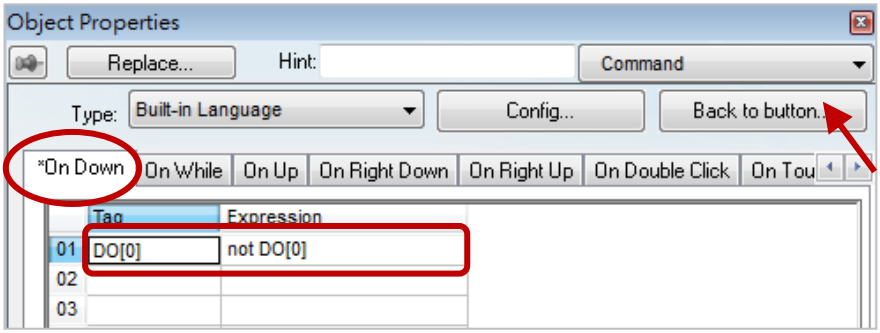
On the "Graphics" tab, right-click Screen and click Insert to add a screen. Using objects listed on the Graphics menu to design an HMI page, and double-click the object to assign the I/O tag.



Pushbutton – DI



Button – DO



6.4 Linux Modbus Development Tool

ICP DAS LinPAC family is embedded with a flexible and open-source Linux system which can be used to control tM modules via DCON or Modbus protocols. This section illustrates three kinds of Modbus software tools for users to develop various applications.

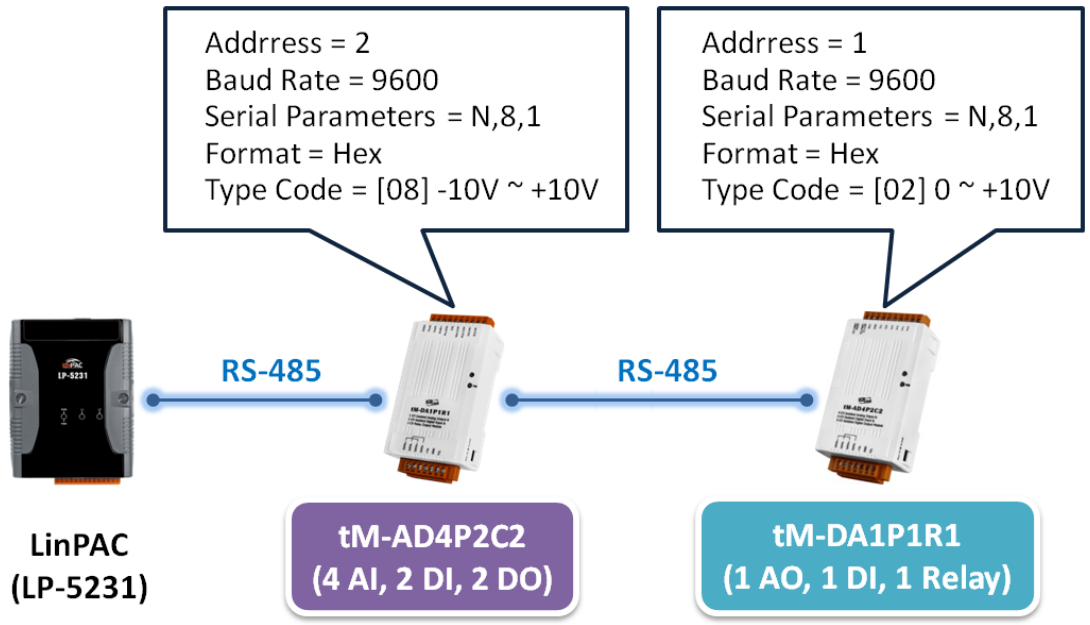
6.4.1 Modbus tool for Linux

The LinPAC series is the Linux-based PAC and support DCON and Modbus protocols. The user can visit the website for more information and download the LinPAC SDK.
https://www.icpdas.com/en/product/guide+Software+Development__Tools+Modbus__Tool#2844

The following table lists the file path of the Modbus tool for AM335x SDK:

OS platform	File path of SDK
Windows	C: \cygwin\LinPAC_AM335x_SDK\examples**xvboard** or C: \cygwin\LinPAC_AM335x_SDK\examples**modbus**
Linux	root@LinuxPC-ICPDAS: /icpdas/linpac_am335x_sdk/i8k/examples/**xvboard** or root@LinuxPC-ICPDAS: /icpdas/linpac_am335x_sdk/i8k/examples/**modbus**

In this case, using the PAC (LP-5231) and two modules (tM-DA1P1R1 and tM-AD4P2C2) for testing. The parameters for the two modules will be set as follows:



First, set the Net-ID of the slave device, follow the steps below:

```
root@LP-5231:~# getmodbus 2 9600 1 4 484 1 100
1
root@LP-5231:~# setmodbus 2 9600 1 16 484 1 2 100
root@LP-5231:~# getmodbus 2 9600 2 4 484 1 100
2
```

Step 1: Use the command to **query** the Net-ID of the tM-AD4P2C2 module.

Command:

getModbus <comport> <baudrate> <NetID> <command> <address> <count> <timeout(ms)>

```
#   getModbus 2 9600 1 4 484 1 100
```

Step 2: Use the command to **modify** the Net-ID of the tM-AD4P2C2 module.

Command:

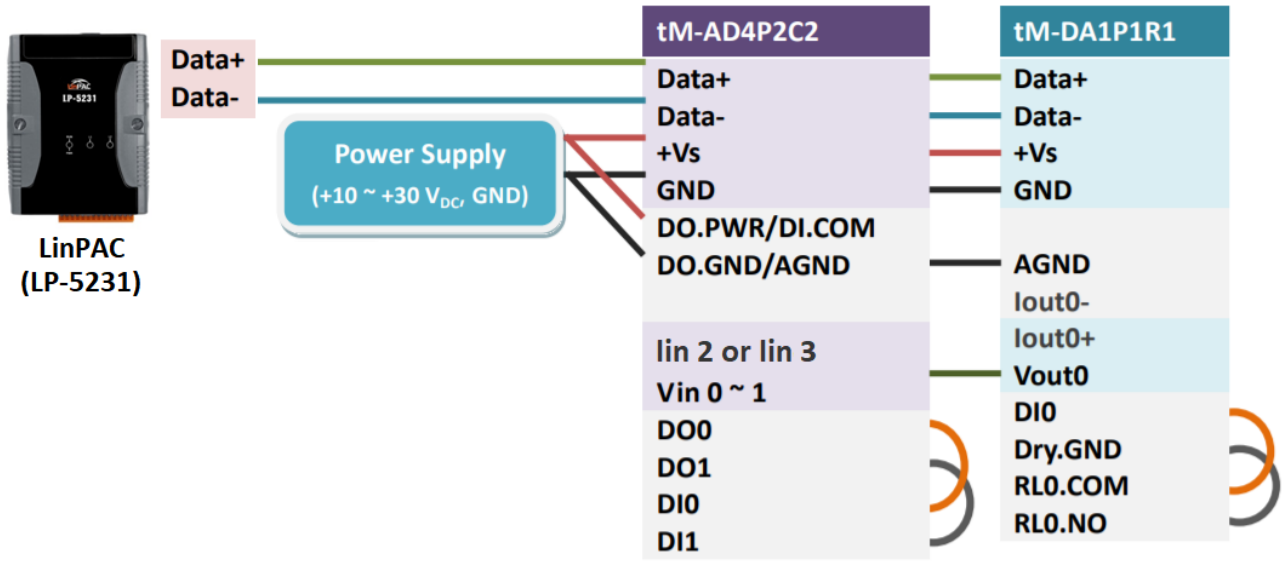
setModbus <comport> <baudrate> <NetID> <command> <address> <count> <value> <timeout(ms)>

```
#   setModbus 2 9600 1 16 484 1 2 100          // Set the NetID as 2
```

Note:

1. Refer to Appendix C - Modbus Register Mapping (Base 1) to set the address of the device.
2. The base address for tM series module is **0** (i.e., Base 0). For example, the Modbus registers 40485 is used to read/write module address (i.e., Net-ID). In this case, you should use the address 484 to get or set the Net-ID.

Hardware Wiring for this example:



❒ Test the AI/AO channel

Wiring the AO channel of tM-DA1P1R1 to the AI channel of tM-AD4P2C2, and the setModbus.c and getModbus.c programs can be used to test AI/AO. Follow the steps below:

```
root@LP-5231:~# setmodbus 2 9600 1 16 32 1 65535 100
root@LP-5231:~# getmodbus 2 9600 1 3 32 1 100
65535
root@LP-5231:~# getmodbus 2 9600 2 4 0 1 100
32767
```

Step 1: Use the command to **set** the AO value of the tM-DA1P1R1 module.

```
#   setModbus 2 9600 1 16 32 1 65535 100      //Output 10 V
```

Step 2: Use the command to **read** the AO value of the tM-DA1P1R1 module.

```
#   getModbus 2 9600 1 3 32 1 100
```

Step 3: Use the command to **read** the AI value of the tM-AD4P2C2 module.

```
#   getModbus 2 9600 2 4 0 1 100
```

❒ Test the DI/DO channel

Wiring the DO channel of tM-DA1P1R1 to its DI channel, and the setModbus.c and getModbus.c programs can be used to test DI/DO. Follow the steps below:

```
root@LP-5231:~# setmodbus 2 9600 1 15 0 1 1 100
wCount=1 iCount=8 iIndex=0
root@LP-5231:~# getmodbus 2 9600 1 1 0 1 100
1
root@LP-5231:~# getmodbus 2 9600 1 2 32 1 100
1
```

Step 1: Use the command to **set** the DO status of the tM-DA1P1R1 module.

```
#   setModbus 2 9600 1 15 0 1 1 100      //Set the status of DO channel to "ON"
```

Step 2: Use the command to **read** the DO status of the tM-DA1P1R1 module.

```
#   getModbus 2 9600 1 1 0 1 100
```

Step 3: Use the command to **read** the DI status of the tM-DA1P1R1 module.

```
#   getModbus 2 9600 1 2 32 1 100
```

Wiring the DO channel of tM-AD4P2C2 to its DI channel, and the setModbus.c and getModbus.c demo programs can be used to test DI/DO. Follow the steps below:

```
root@LP-5231:~# setmodbus 2 9600 2 15 0 1 1 100
wCount=1 iCount=8 iIndex=0
root@LP-5231:~# getmodbus 2 9600 2 1 0 1 100
1
root@LP-5231:~# getmodbus 2 9600 2 2 32 1 100
1
```

Step 1: Use the command to **set** the DO status of the tM-AD4P2C2 module.

```
#   setModbus 2 9600 2 15 0 1 1 100      //Set the status of DO channel to "ON"
```

Step 2: Use the command to **read** the DO status of the tM-AD4P2C2 module.

```
#   getModbus 2 9600 2 1 0 1 100
```

Step 3: Use the command to **read** the DI status of the tM-AD4P2C2 module.

```
#   getModbus 2 9600 2 2 32 1 100
```

6.4.2 Using LinPAC and the Python Application

The LinPAC series supports the Python programming language. The user can find the Modbus tool for testing tM series module from the official website of Python. In this example, the LP-5231 module is connected to the tM-DA1P1R1 module and the modbus-tk tool is used for accessing Modbus registers.

For more information about the modbus-tk tool, visit https://github.com/ljean/modbus-tk.
Follow these steps to install the software and test the module:

Step 1: Use the command to check if the version of Python is 2.5 or later.

```
#   python --version
```

```
root@LP-5231:~# python --version
Python 2.7.3
root@LP-5231:~#
```

Step 2: Use the command to install pyserial module.

```
#   pip install pyserial
```

Step 3: Use the command to download the modbus-tk package.

```
#   wget https: //github.com/ljean/modbus-tk/archive/master.zip
```

Step 4: Use the command to unzip the modbus-tk package.

```
#   unzip master.zip
```

Step 5: Use these commands to install the modbus-tk tool.

```
#   cd modbus-tk-master
#   python setup.py build
#   python setup.py install
```

Step 6: Use the command to check if "pyserial" and "modbus-tk" have been installed successfully.

```
#   pip list
```

```
root@LP-5231:~/modbus-tk-master# pip list
Package    Version
---------- ------------
distribute 0.6.24dev-r0
modbus-tk  0.5.8
pip        18.1
pyserial   3.4
setuptools 0.6rc11
```

Step 7: Find the "rtumaster_example.py" demo program provided by modbus-tk.

```
root@LP-5231:~# cd modbus-tk-master/examples/
root@LP-5231:~/modbus-tk-master/examples# ls
modbus_system_monitor.py  rtumaster_example.py  tcpmaster_example.py
mysimu.py                 rtuslave_example.py   tcpslave_example.py
root@LP-5231:~/modbus-tk-master/examples#
```

Step 8: Modify the parameters of the rtumaster_example.py demo program.

```
import modbus_tk
import modbus_tk.defines as cst
from modbus_tk import modbus_rtu
#PORT = 1
PORT = '/dev/ttyO2'
def main():
    """main"""
    logger = modbus_tk.utils.create_logger("console")
    try:
        #Connect to the slave
        master = modbus_rtu.RtuMaster(
          serial.Serial(port=PORT, baudrate=9600, bytesize=8, parity='N', stopbits=1, xonxoff=0)
        )
        master.set_timeout(5.0)
        master.set_verbose(True)
        logger.info("connected")
        logger.info(master.execute(1, cst.READ_HOLDING_REGISTERS, 32, 1))
        #
        #
        #
        #
```

The communication port

The communication parameters

The Modbus command
master.execute(NetID, function code, address, count)

Note: The base address for tM series module is '0' (Base 0).

Step 9: Execute the demo program to read the AO value of the tM-DA1P1R1 module. The results will be displayed as illustrated in the figure below.

```
root@LP-5231:~/modbus-tk-master/examples# python rtumaster_example.py
2018-12-10 17:47:25,575 INFO   modbus_rtu.__init__     MainThread   RtuMaster /dev/ttyO2 is opened
2018-12-10 17:47:25,578 INFO   rtumaster_example.main  MainThread   connected
2018-12-10 17:47:25,580 DEBUG  modbus.execute  MainThread     -> 1-3-0-32-0-1-133-192
2018-12-10 17:47:25,606 DEBUG  modbus.execute  MainThread     <- 1-3-2-255-255-185-244
2018-12-10 17:47:25,607 INFO   rtumaster_example.main  MainThread    (65535,)
```

6.4.3 Using LinPAC and the Perl Application

The LinPAC series supports the Perl programming language. For testing tM series module, users can find the Modbus tool from the official website of Perl. In this example, the LP-5231 module is connected to the tM-DA1P1R1 module and the Device-Modbus-RTU tool is used for accessing Modbus registers. Follow the steps to install the software and test the module:

Step 1: Download and unzip the Device-Modbus-RTU package (Device-Modbus-RTU-0.022.tar.gz) from the website https: //metacpan.org/release/Device-Modbus-RTU

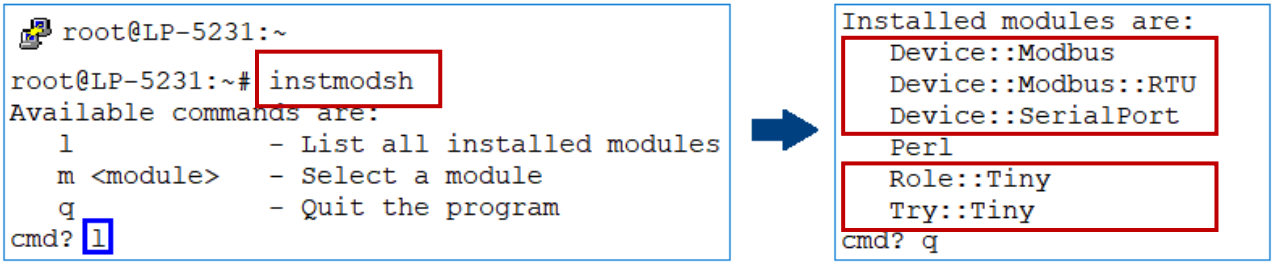
Step 2: Use the command to install the dependent module for Device-Modbus-RTU.

```
#   sudo cpan Role: : Tiny Try: : Tiny Device: : SerialPort Device: : Modbus
```

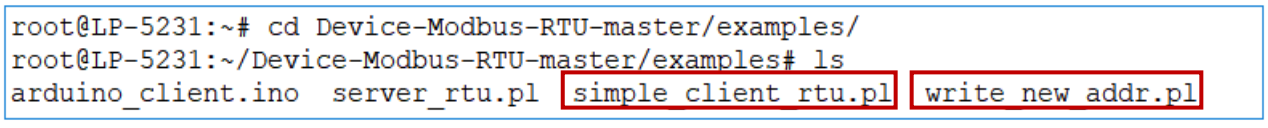
Step 3: Use the command to install the Device-Modbus-RTU tool.

```
#   cd Device-Modbus-RTU-0.022
#   perl Makefile.PL
#   make
#   make test
#   make install
```

Step 4: Use the 'instModsh' command to check if the Perl module has been installed successfully.



Step 5: Find the "write_new_addr.pl" and "simple_client_rtu.pl" demo programs provided by Device-Modbus-RTU.



Step 6: Modify the parameters of demo programs.

❒ Modify the COM port setting in the “write_new_addr.pl” and “simple_client_rtu.pl” scripts.

```
my $client = Device: : Modbus: : RTU: : Client->new(
    port     => '/dev/ttyO2',     // The number of COM port
    baudrate => 9600,             // Bits per second
    parity   => 'none',           // Parity check
);
```

❒ Modify the Modbus command setting in the "write_new_addr.pl" script to set the AO value.

```
my $req = $client->write_single_register(        // Modbus function code
    unit     => 1,                // The NetID of Slave device
    address  => 32,               // The channel address
    value    => 65535             // Set the AO vale
);
```

❒ Modify the Modbus command setting in the “simple_client_rtu.pl” script to read the AO value.

```
my $req = $client->read_holding_registers(       // Modbus function code
    unit     => 1,                // The NetID of Slave device
    address  => 32,               // The channel address
    quantity => 1,                // The number of channels to read
);
```

The "write_new_addr.pl" demo program:

```
#! /usr/bin/env perl
use Device::Modbus;
use Device::Modbus::RTU::Client;
use Data::Dumper;
use strict;
use warnings;
use v5.10;
my $client = Device::Modbus::RTU::Client->new(
    port     => '/dev/ttyO2',
    baudrate => 9600,
    parity   => 'none',
);
my $req = $client->write_single_register(
    unit     => 1,
    address  => 32,
    value    => 65535
);
say "->" . Dumper $req;
$client->send_request($req);
my $resp = $client->receive_response;
say "<-" . Dumper $resp;
```

The COM port setting

The Modbus command setting

Note: The base address for tM series module is ‘0’ (Base 0).

Step 7: Execute the demo program to control tM series module.

(1) The results of executing the "write_new_addr.pl" program.

```
root@LP-5231:~/Device-Modbus-RTU-master/examples# perl write_new_addr.pl
->$VAR1 = bless( {
                   'unit' => 1,
                   'function' => 'Write Single Register',
                   'value' => 65535,
                   'address' => 32,
                   'code' => 6
                 }, 'Device::Modbus::Request' );
<-$VAR1 = bless( {
                   'unit' => 1,
                   'crc' => 45193,
                   'message' => bless( {
                                        'function' => 'Write Single Register',
                                        'value' => 65535,
                                        'address' => 32,
                                        'code' => 6
                                      }, 'Device::Modbus::Response' )
                 }, 'Device::Modbus::RTU::ADU' );
```

(2) The results of executing the “simple_client_rtu.pl” program.

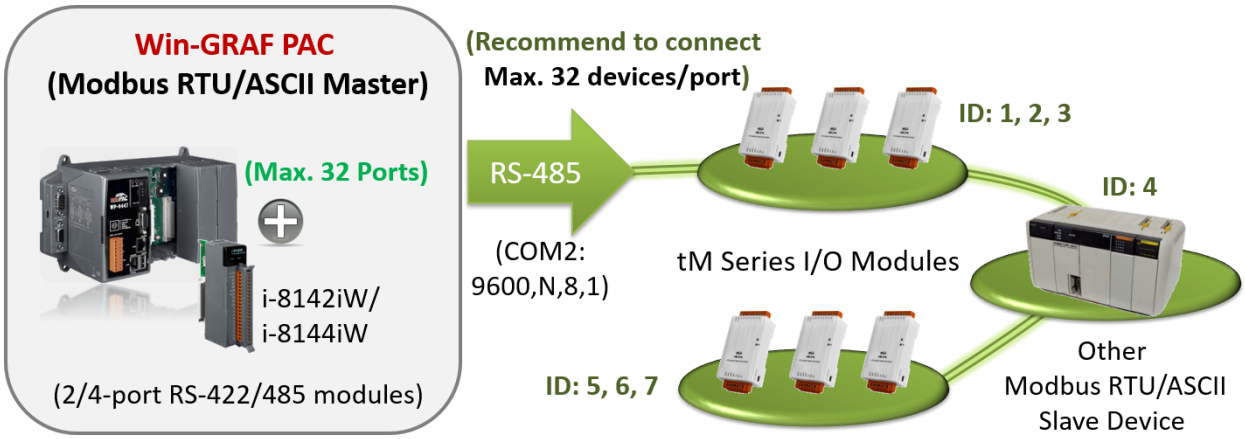
```
root@LP-5231:~/Device-Modbus-RTU-master/examples# perl simple_client_rtu.pl
->$VAR1 = bless( {
                   'unit' => 1,
                   'function' => 'Read Holding Registers',
                   'quantity' => 1,
                   'address' => 32,
                   'code' => 3
                 }, 'Device::Modbus::Request' );
<-$VAR1 = bless( {
                   'unit' => 1,
                   'crc' => 62649,
                   'message' => bless( {
                                        'bytes' => 2,
                                        'function' => 'Read Holding Registers',
                                        'values' => [
                                                      65535
                                                    ],
                                        'code' => 3
                                      }, 'Device::Modbus::Response' )
                 }, 'Device::Modbus::RTU::ADU' );
```

6.5 Using Win-GRAF Development Software

Win-GRAF is a powerful SoftLogic development software and PLC-like SoftLogic package that supports IEC 61131-3 Standard Open PLC Languages running on Windows 7/8 (or later). The Win-GRAF Runtime application can run on any Win-GRAF PAC, such as the WinPAC series WP-5238-CE7, WP-8xx8, and WP-8xx8-CE7, or the touch panel ViewPAC series VP-x208-CE7 and VP-x238-CE7, or the advanced CPU XPAC-CE6 series XP-8x38-CE6.

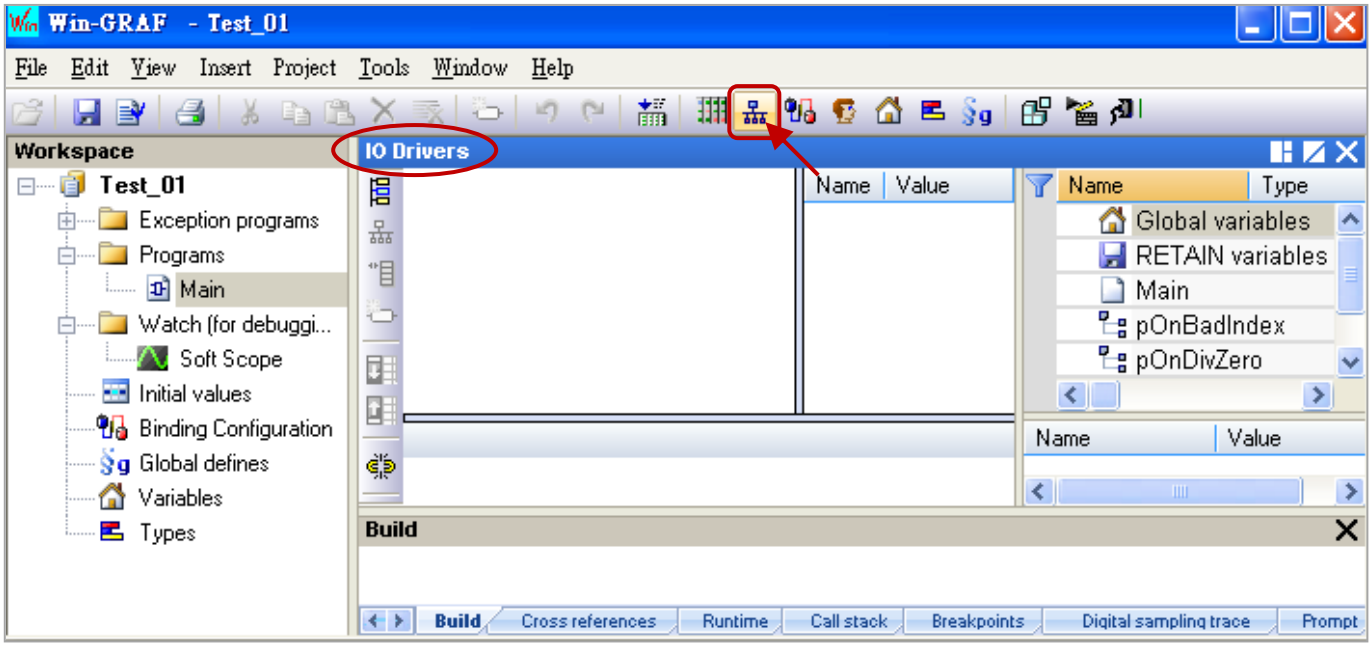
Using the Win-GRAF Workbench with ICP DAS Win-GRAF PACs, the control/monitor systems can easily implement an industrial level of data acquisition and logic control in various industry fields.
Website: https://www.icpdas.com/en/product/guide+Software+Development__Tools+Win-GRAF

Win-GRAF PAC supports Modbus RTU/ASCII/TCP/UDP Master/Slave and DCON protocols. The following example will describe how to enable the Win-GRAF PAC as Modbus RTU/ASCII Master to connect to a tM series module.

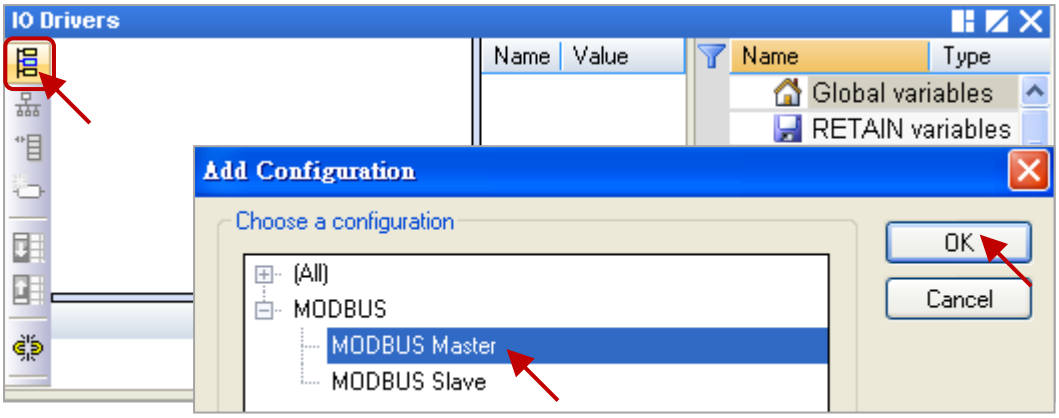


Steps for using Win-GRAF:

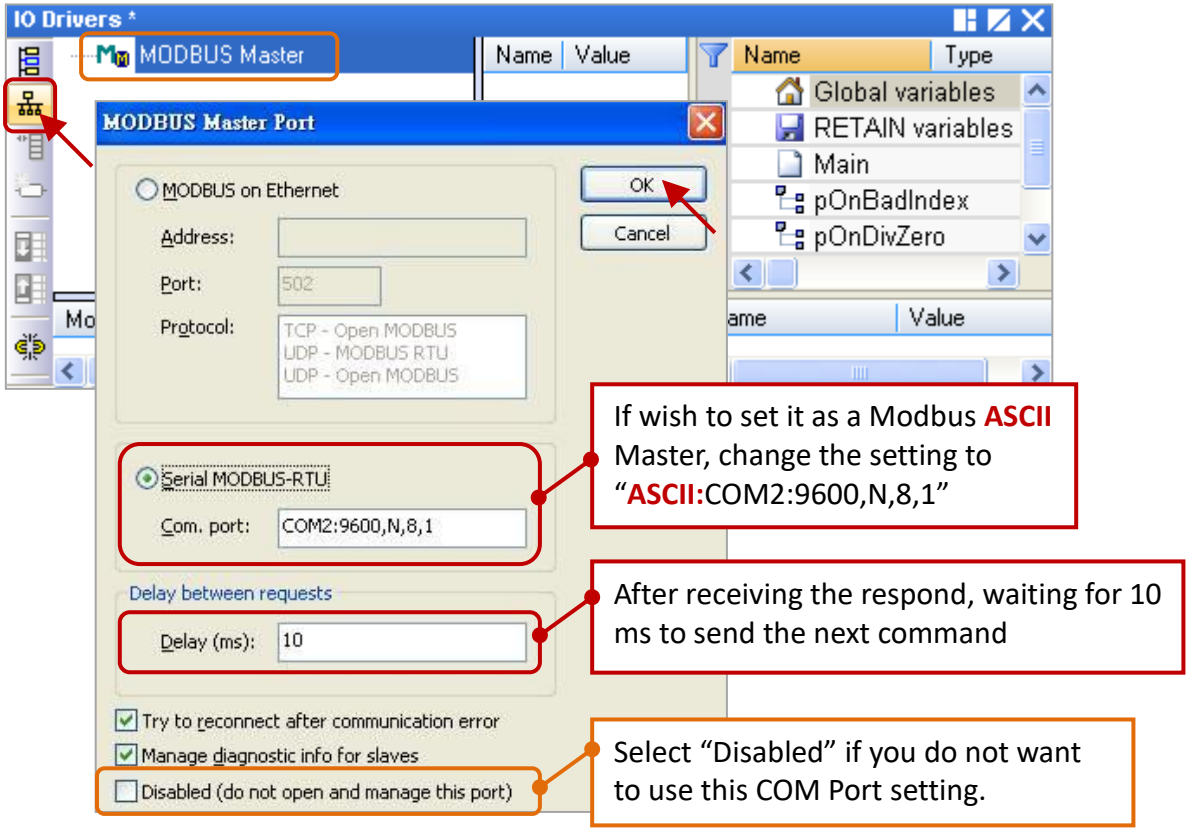
1. Mouse click the “Open Fieldbus Configuration” tool button to open “IO Drivers” window.



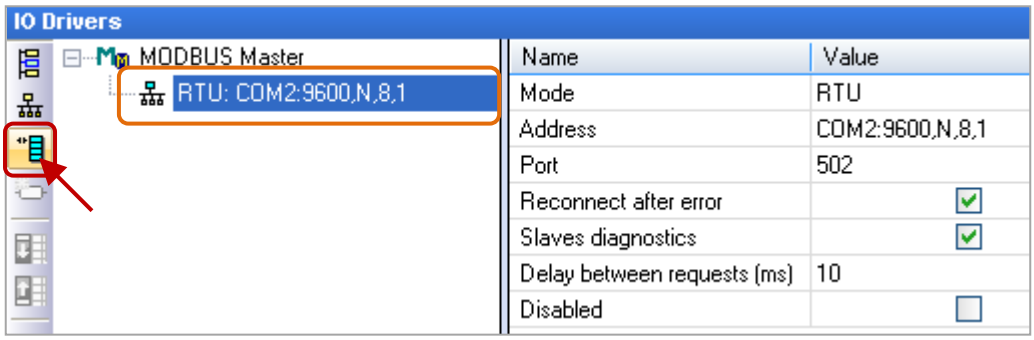
2. Click the “Insert Configuration” button on the left of the “IO Drivers” window, then click the “MODBUS Master” and “OK” to enable the Modbus Master setting.



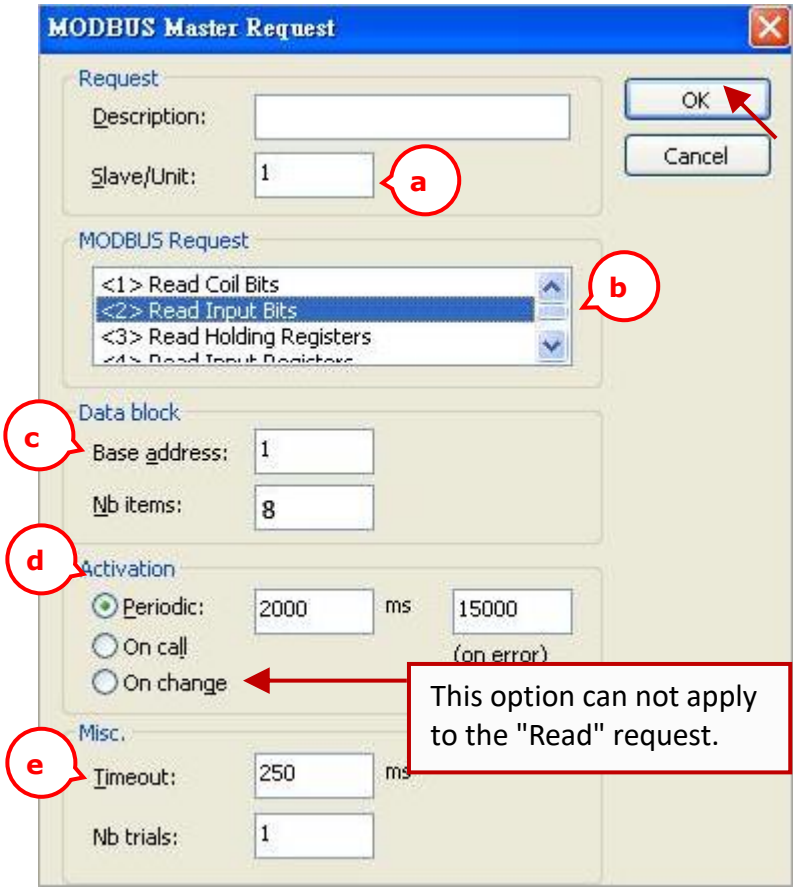
3. Click the “Insert Master/Port” button on the left side to open the setting window. Next, select the “Serial MODBUS-RTU”, set COM Port (e.g., “COM2:9600,N,8,1”) and Delay time (recommended value: 10 ms, it can be 0 to 10000), and then click “OK”.



4. Click the “Insert Slave/Data Block” button on the left side to create a data block.



5. Each data block represents one Modbus Master request, add it according to the application. Configure the following settings in the “MODBUS Master Request” windows and click “OK".



a. Slave/Unit:
Enter the Net-ID of the Slave device. (In this case, the Net-ID is “1”).

b. MODBUS Request:
In this case, it used to read DI data. Select “<2> Read Input Bits” option.

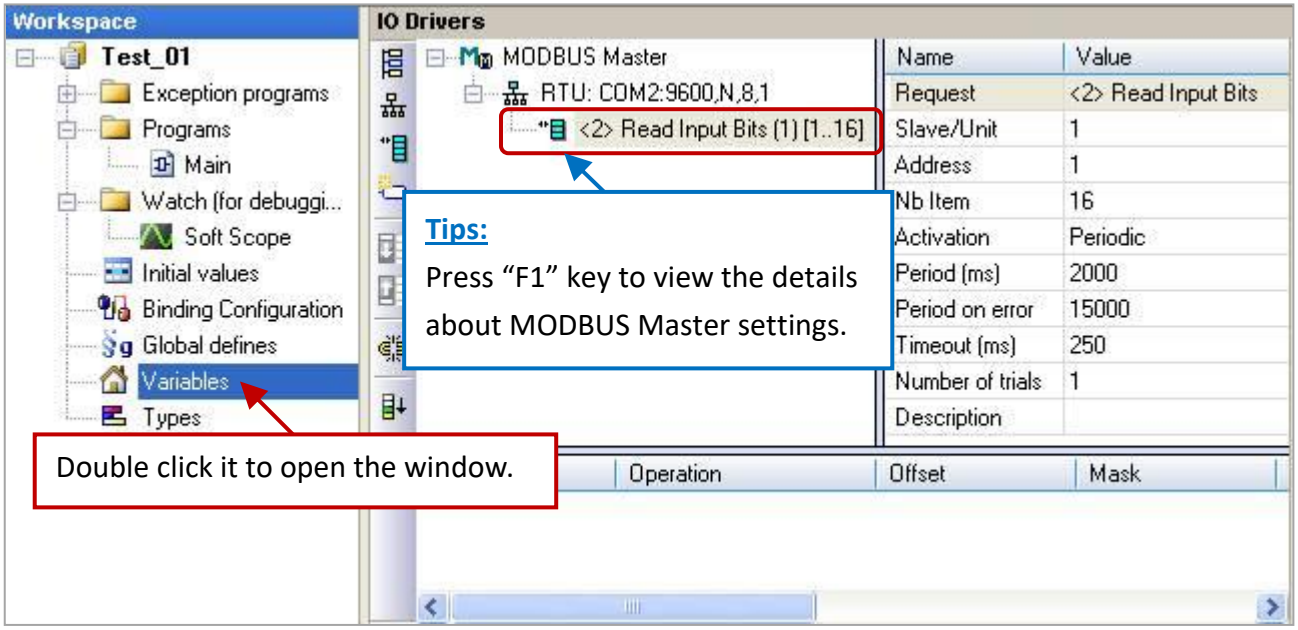
c. Base address:
Start from “1” by default.

Nb items:
The number of DI signals to read. (In this case, the number is “8”).

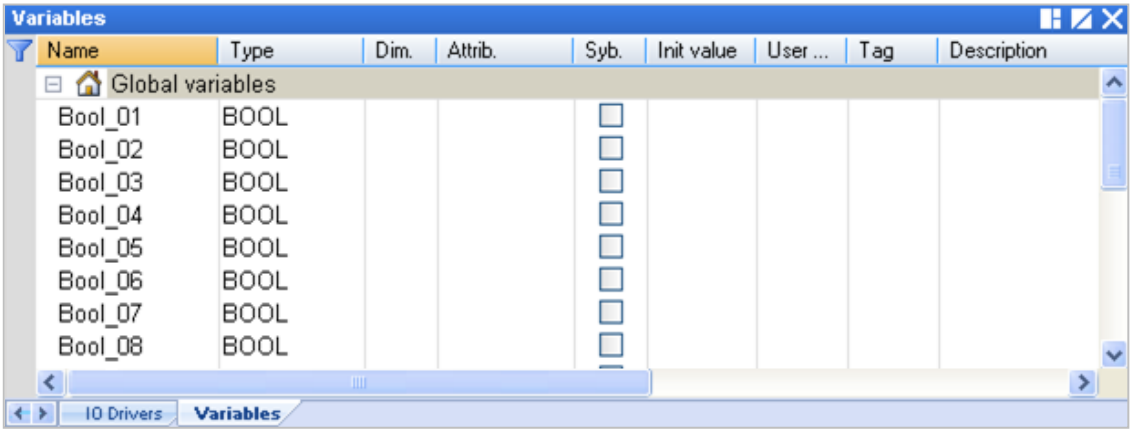
d. Periodic:
Sending the request periodically. (In this case, to send once every two seconds.) “on error” means the next sending time when an exception occurred (e.g., 15 seconds).

e. Timeout: Set a timeout value. When time-out occurred, it will show the defined error code. The recommended value for the Modbus RTU/ASCII device is 200 to 1000 ms. E.g., 250 ms)

6. In the “I/O Drivers” window, a "Read Input Bits" data block is added. Next, open the “Variables” window and then add variables for accessing I/O data.

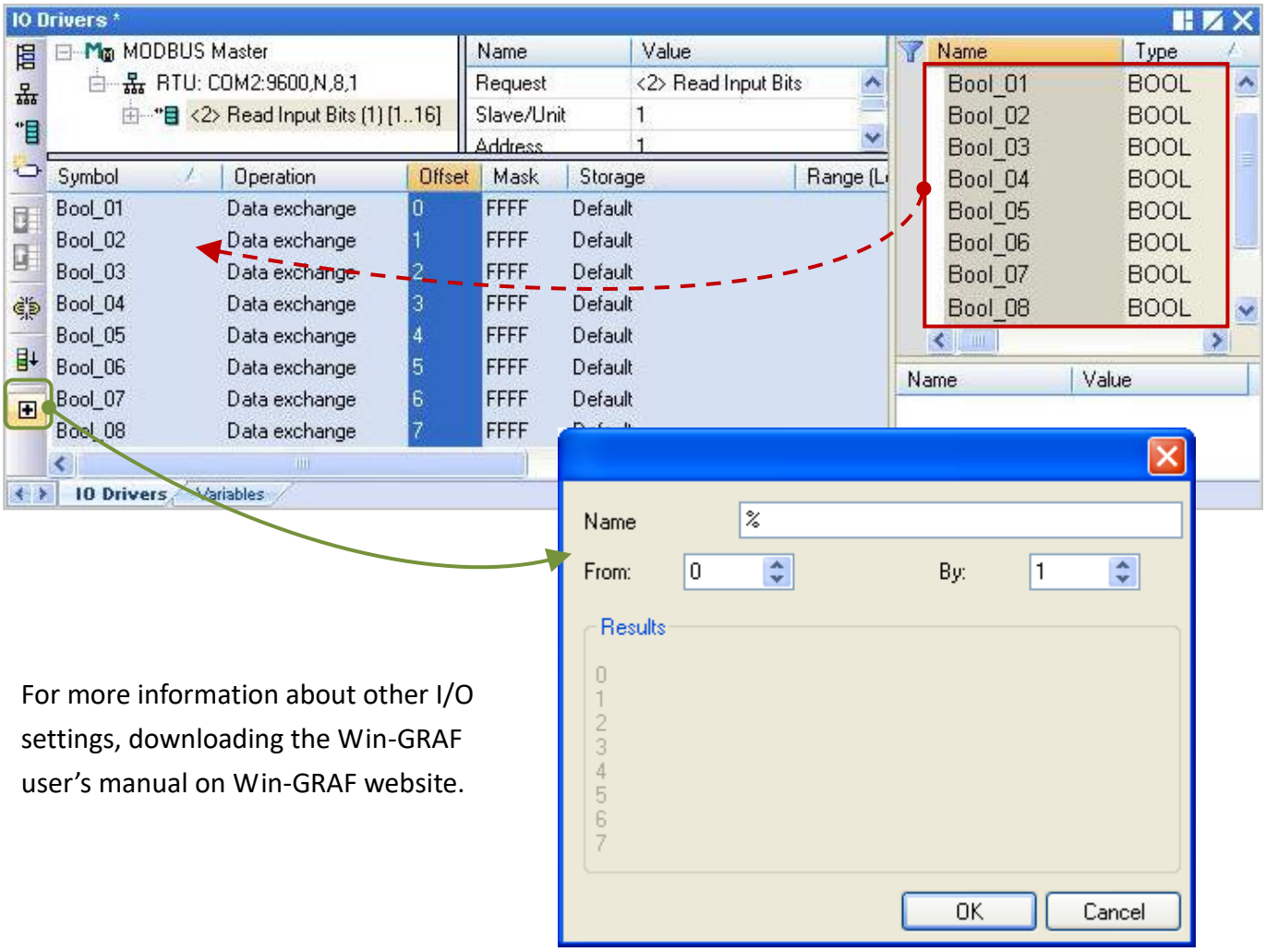


There are 8 variables (Name: “Boo_01 to "Boo_08”; Type: BOOL) that can be used to read 8 DI data.



7. Drag variables (i.e., “Boo_01” to “Boo_08“) into the “Symbol” area of the data block, and choose all “Offset” fields, and then click the “Iterate Property” button on the left side to set the “Offset” value.

Note: The “Offset” value starts at “0” and the Modbus address of the variable is equal to this value plus 1 (Base address). If using a 32-bits (or more than 32-bits) data type (e.g., “DINT”), it requires two Modbus addresses.

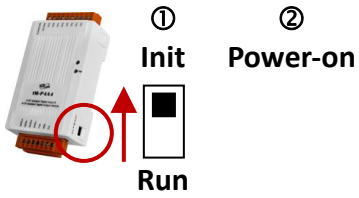


For more information about other I/O settings, downloading the Win-GRAF user’s manual on Win-GRAF website.

Appendix A Function Description

A.1 INIT Mode

tM series module built-in an EEPROM that can be used to save data such as the module address, the type code, the baud rate, etc. Sometimes, the module cannot be used because the user forgot the communication parameters of the module and there is no information to look up the settings. tM series features the “INIT” mode to solve this problem.

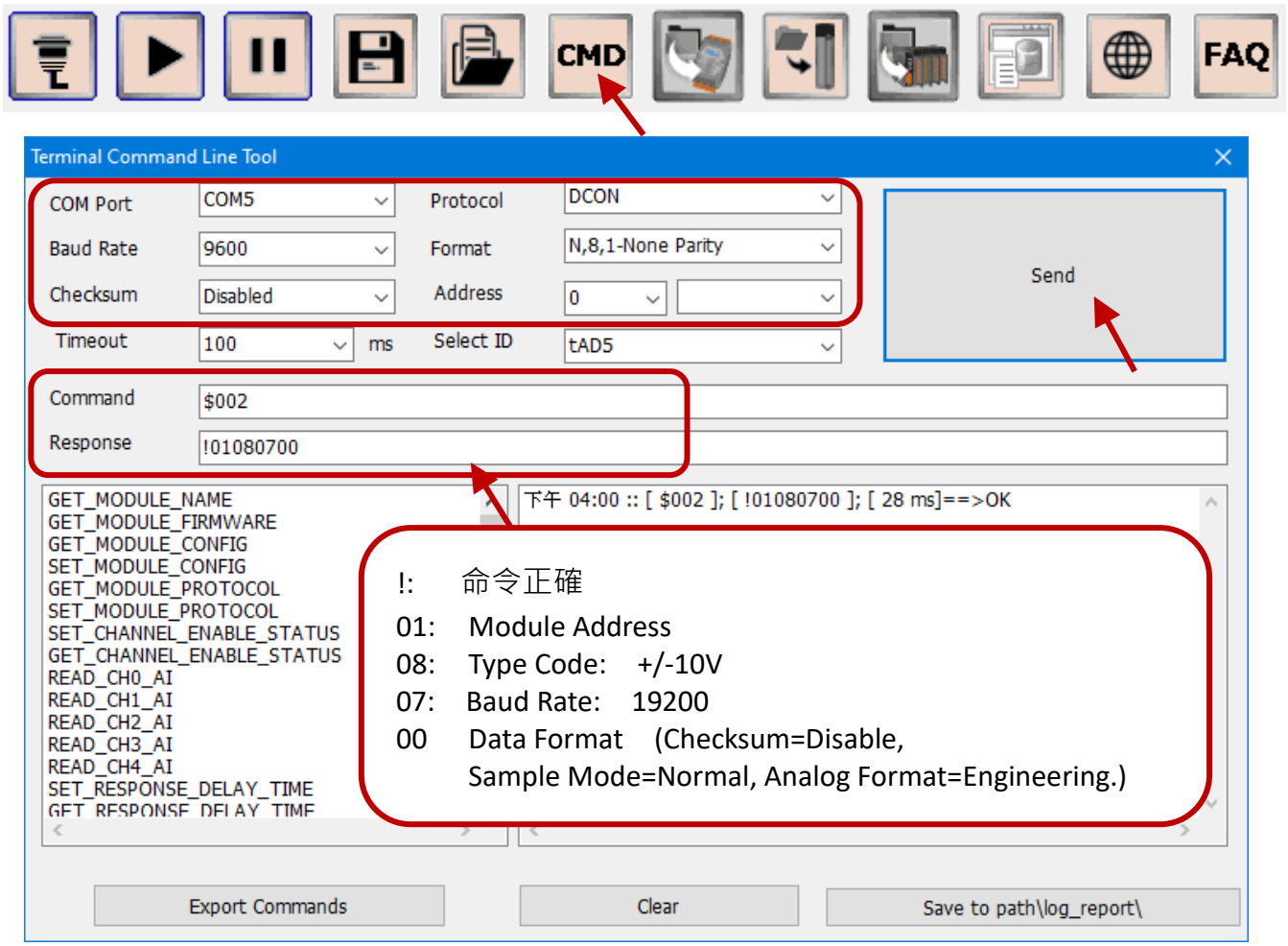


When the module is powered on in “INIT” mode, the communication parameters of the module are loaded as follows so that users can quickly found the module.

1. Protocol: **DCON**
2. Address: **00**
3. Baud Rate: **9600 bps**
4. Checksum: **Disable**

Under the “INIT” mode, the Command Line function of the DCON Utility Pro allows the user to enter “$002” command to read the original module settings that are stored in an EEPROM. Refer to the $AA2 command in Appendix D to understand the response value.

Note: The user only needs to enter “$002”, and the <CR> character will automatically be added and hidden by DCON Utility Pro.



A.2 Dual Watchdog

For different application fields in smart buildings, important medical automation, traffic flow monitoring, and factory production process, systems may be shut down due to natural disasters (e.g., lightning strike) or the harsh environment (e.g., noise or external signals) or any unexpected situations.

If the system is installed at a place where hard to manually reboot, or if the crashed system will lead to a dangerous accident or a huge loss of capacity and cost, the dual watchdog can be used to automatically reboot the equipment and load a Safe Value to ensure the system is working properly and safely.

tM series I/O module features a dual watchdog protection mechanism.

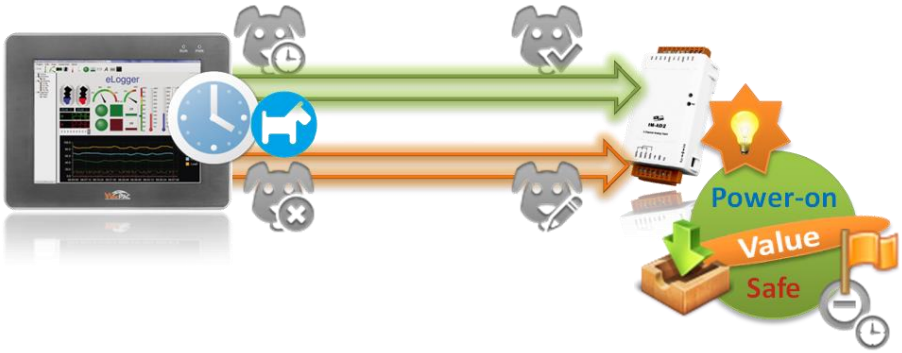
Dual Watchdog = Module Watchdog + Host Watchdog

Module Watchdog:

Module Watchdog is a hardware reset circuit that monitors the operating status of the module. Hardware watchdog using a timer that will regularly be reset to 0 and prevent it from timing out when the module is operating normally. If a timeout occurs, the module will automatically reboot and load a predefined Power-on Value.

Host Watchdog:

Host Watchdog is a software function to monitor the communication between the host and the I/O module. Software watchdog using a CPU timer of a host. When the system is working normally, it will regularly send a host-ok signal to the module. If a timeout occurs, the module will immediately load a Safe Value. If the status of time-out is cleared, a predefined Power-on Value will be loaded after a reboot.



Using DCON Utility Pro to test the host watchdog function, refer to Section 3.4.5 - Configuration Page - Host WDT. On the “Host WDT” page, the user can set the duration of WDT (Watchdog Timeout), also clear the status of timeout.

A.2.1 Power-on Value

When the module is powered on or warm boot or reboot due to the module Watchdog timeout, a predefined Power-on Value will be loaded into the output channel.

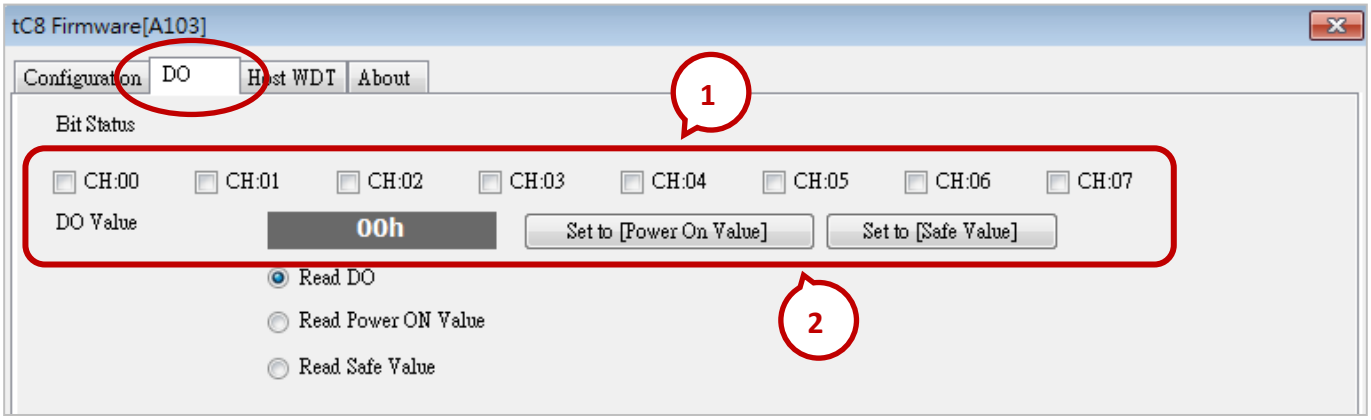
A.2.2 Safe Value

If host watchdog is enabled and the timeout occurs, a Safe Value for the output channel will be loaded. Note that the PWR LED on the tM module will be blinking if the timeout occurs. The time-out status will be stored in the EEPROM and will not be changed even if the module is rebooting. If the status of the timeout is cleared, a predefined Power-on Value will be loaded after a reboot.

Refer to Appendix E.3 Host Watchdog for more details about commands.

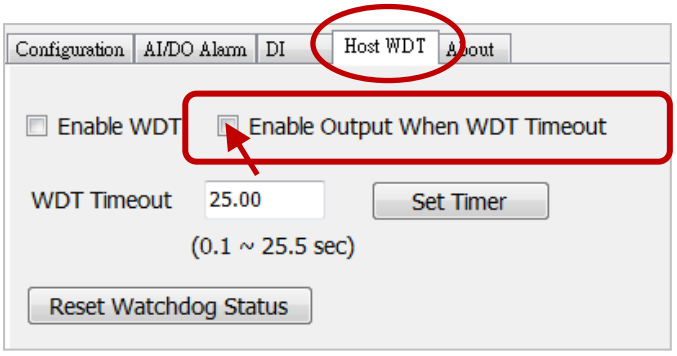
Command	Description
~AA1	Reset the host watchdog status of a module
~AA5N	Set the current AO value as Safe Value
~AA5V	Set the current DO value as Power-on Value or Safe Value
$AA4N	Set the current AO value as Power-on Value

Using DCON Utility Pro to set the Power-on Value or the Safe Value, also enable the host watchdog function (reference: Section 3.4.2 to Section 3.4.5). On the “AO/DO” page, the user can set the output value for each channel of the tM module.



Note:

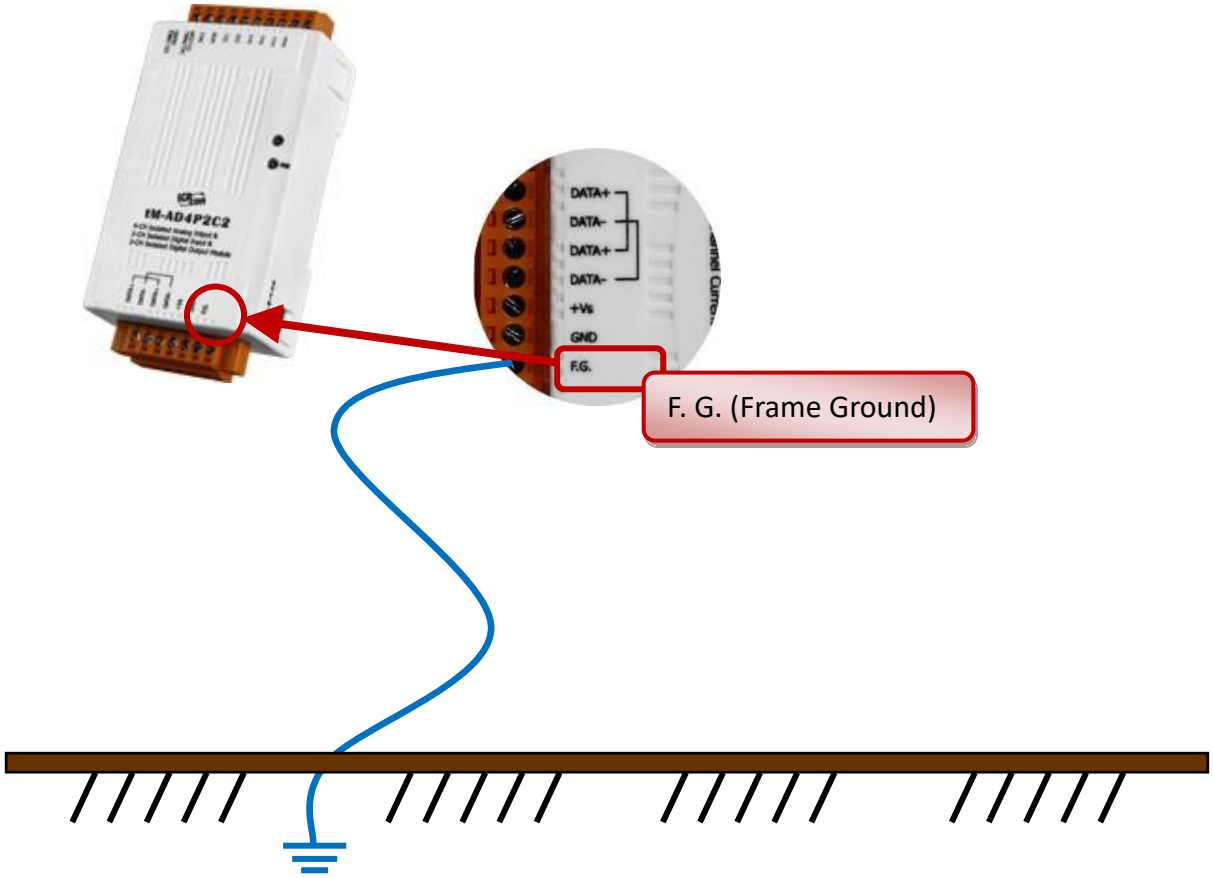
When using the DCON protocol, in case of the host watchdog timeout occurs, the output value will automatically be set to a Safe Value, and cannot be changed. When using the Modbus protocol, the DCON Utility Pro provides the function to change the AO/DO value even if the WDT timeout occurs.



A.3 Frame Ground

Electronic circuits are constantly vulnerable to Electro-Static Discharge (ESD), which becomes worse in a continental climate area. The tM modules feature a new design for the frame ground, which provides a path for bypassing ESD, allowing enhanced static protection (ESD) capability and ensures that the module is more reliable.

Connect the frame ground terminal to a wire/DIN rail and connect the wire/DIN rail to the earth ground will provide better protection for the module.

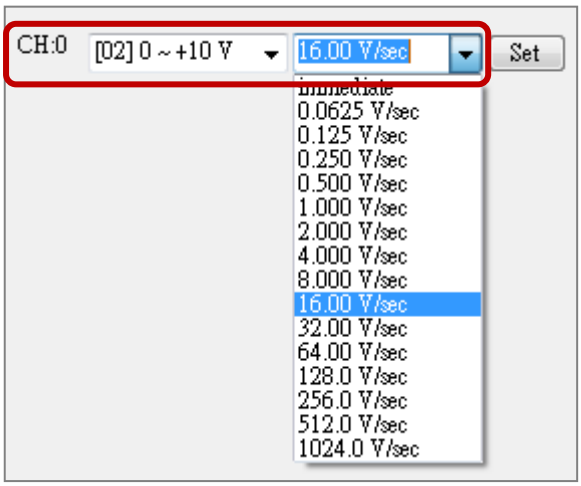


A.4 Slew Rate Control

The feature is used for an analog output module, e.g., tM-DA1P1R1. In general, the output of an analog output module changes instantaneously. That is, when the module receives an output command, its output changes to the specified value immediately.

However, it may require that the output change to the specified value gradually in some applications. The slew rate control is to adjust the output change rate.

Note: The tM-DA1P1R1 modules update the analog output every 10 ms.

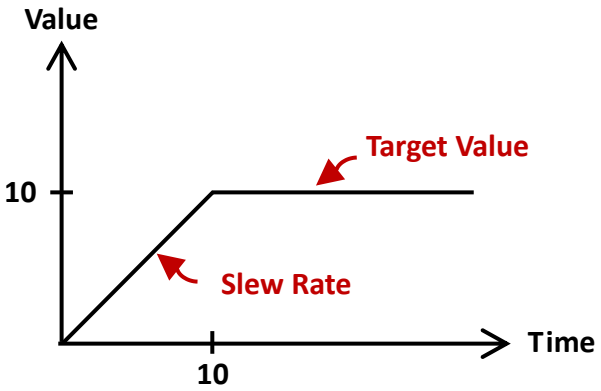


The AO page of DCON Utility Pro

Slew Rate (S) Configuration Table

S	V/sec	mA/sec
0	Immediate	Immediate
1	0.0625	0.125
2	0.125	0.25
3	0.25	0.5
4	0.5	1.0
5	1.0	2.0
6	2.0	4.0
7	4.0	8.0
8	8.0	16.0
9	16.0	32.0
A	32.0	64.0
B	64.0	128.0
C	128.0	256.0
D	256.0	512.0
E	512.0	1024.0

For example, the current output value of the module is 0 V. The user can change the output value to 10 V immediately or specify a slew rate to change the value to 10 V gradually.



A.5 AO Read-back Value

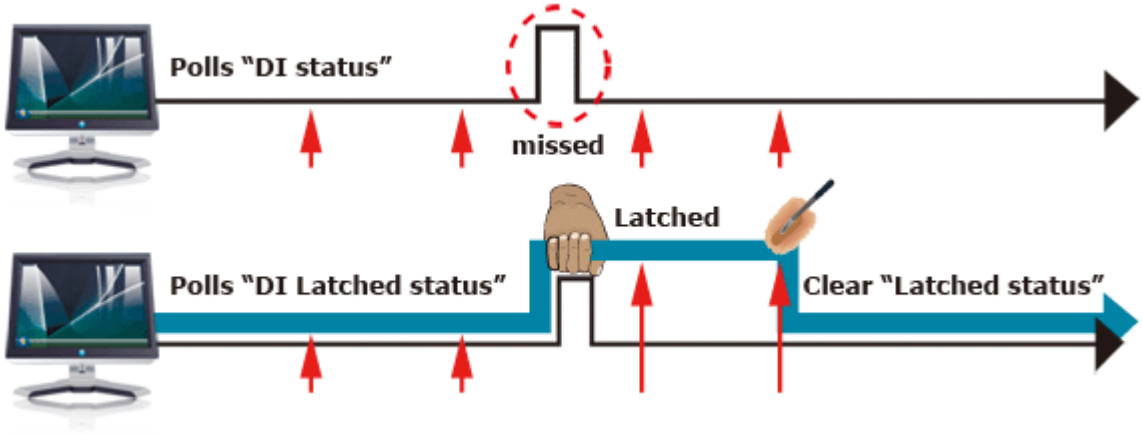
Some modules have built-in electrical circuits to measure the physical output value, also called the read-back value. The tM-DA1P1R1 module doesn't have these circuits so the read-back value equals the AO value. Besides, if users specify a slew rate, the AO value will be changed gradually so both the read-back value and AO value are different.

A.6 Advanced DI Features (Latch and Counter)

The DI channel of a module is not only used to read the digital input status, but also provides several kinds of advanced functions.

DI Latch Function

All DI channels of tM series modules provide the Latch function to keep the high/low events in the internal registers of the module. In general, the host controller polls modules one by one to get all DI status. Because RS-485 is a low-speed field bus, the polling will take time and probably miss a short duration signal. With the DI latch function, the short duration (>=5ms) signal will not be lost anymore.



tM series modules support multiple commands to read the high/low latched status, refer to Appendix E.4 – DI Latched Commands for more information.

Low-Speed Counter

The DI module not only detects the high/low level but also automatically counts the DI signal in the background. The signal under 100Hz can be detected and counted.

Max. Count:
65535 (16-bit)

Appendix B Type Codes and Ranges for AI/AO Data

All type codes used for the tM series analog input/output module are listed as follows. Each type-code supports two kinds of data formats:

■ **Engineering:**

Normally, the AO value gets from the software is 1000 or 10000 times the real value. For example,

Type Code 05: The value gets from the software is 15000 which means the real value is 1.5 V;
The value gets from the software is -23000 which means the real value is -2.3 V.

Type Code 08: The value gets from the software is 2500 which means the real value is 2.5 V;
The value gets from the software is -6500 which means the real value is -6.5 V.

Type Code	Range	Data Format	Min.	Max.
05	-2.5 V to +2.5 V	Engineering	-25000	+25000
		2's Complement	8000h	7FFFh
06	-20 mA to +20 mA	Engineering	-20000	+20000
		2's Complement	8000h	7FFFh
07	+4 mA to +20 mA	Engineering	+4000	+20000
		2's Complement	0000h	FFFFh
08	-10 V to +10 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh

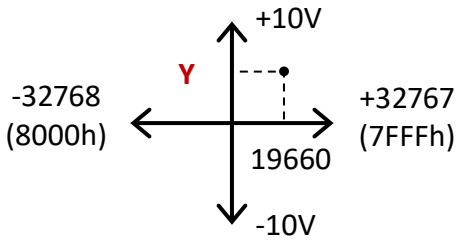
■ **2's Complement:**

The value is usually displayed in a hexadecimal format on the document or the software. For example, "1230h" or "ABCDh" (the "h" refers to hexadecimal). The value gets from the software can be converted to the real value by using the following formula, for example,

Type Code 08: The value gets from the software is 19660 (Dec.) or 4CCCh (Hex.) which means the real value is 6.0 V.

Value Conversion: $\frac{Y}{10} = \frac{19660}{32767}$

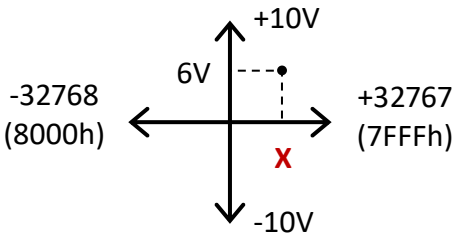
$Y = 6.0$



Type Code 08: To output 6 V, the value 19660 (Dec.) or 4CCCh (Hex.) must be set in the software.

Value Conversion: $\frac{6}{10} = \frac{X}{32767}$

$X = 19660.2$



B.1 Data Ranges for tM-AD2

The tM-AD2 module only supports positive voltage or current. The range of positive voltage, current, and values for each type-code are listed in the table. For example,

Type Code **05:** The data ranges from 0 to +2.5 V.
The value ranges from 0 (0000h) to +32767 (**7**FFFh).

Type Code **1A**: The data ranges from 0 to 20 mA.
The value ranges from 0 (0000h) to +65535 (**F**FFFh).

Type Code	Range	Data Format	Min.	Max.
05	0 V to +2.5 V	Engineering	0	+25000
		2's Complement	0000h	7FFFh
07	4 mA to 20 mA	Engineering	4000	20000
		2's Complement	0000h	FFFFh
08	0 V to +10 V	Engineering	0	+10000
		2's Complement	0000h	7FFFh
09	0 V to +5 V	Engineering	0	+5000
		2's Complement	0000h	7FFFh
0A	0 V to +1 V	Engineering	0	+10000
		2's Complement	0000h	7FFFh
0B	0 V to +500 mV	Engineering	0	+5000
		2's Complement	0000h	7FFFh
0D	0 mA to +20 mA	Engineering	0	+20000
		2's Complement	0000h	7FFFh
1A	0 mA to 20 mA	Engineering	0	20000
		2's Complement	0000h	FFFFh

Note: If the Analog Format is set to "% FSR", the data of any type code ranges from +000.00 to +100.00.

B.2 Data Ranges for tM-AD5

The tM-AD5 module supports positive and negative voltage and the value ranges from -32768 to 32767.

Type Code	Range	Data Format	Min.	Max.
05	-2.5 V to +2.5 V	Engineering	-25000	+25000
		2's Complement	8000h	7FFFh
08	-10 V to +10 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh
09	-5 V to +5 V	Engineering	-5000	+5000
		2's Complement	8000h	7FFFh
0A	-1 V to +1 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh

Note: If the Analog Format is set to "% FSR", the data of any type code ranges from -100.00 to +100.00.

B.3 Data Ranges for tM-AD5C

The tM-AD5C module supports positive and negative current and the value ranges from -32768 to 32767 or from 0 to 65535.

Type Code	Range	Data Format	Min.	Max.
07	4 mA to 20 mA	Engineering	4000	20000
		2's Complement	0000h	FFFFh
0D	-20 mA to +20 mA	Engineering	-20000	+20000
		2's Complement	8000h	7FFFh
1A	0 mA to 20 mA	Engineering	0	20000
		2's Complement	0000h	FFFFh

Note: If the Analog Format is set to "% FSR", the data of any type code ranges from -100.00 to +100.00.

B.4 Data Ranges for tM-AD8

The tM-AD5 module only supports positive voltage and the value ranges from 0 to 32767. Take the type code **0B** as an example, the data ranges from **0 to +500 mV** and the value ranges from 0 (0000h) to +32767 (**7**FFFh).

Type Code	Range	Data Format	-F.S	+F.S
05	0 V to +2.5 V	Engineering	0	+25000
		2's Complement	0000h	7FFFh
08	0 V to +10 V	Engineering	0	+10000
		2's Complement	0000h	7FFFh
09	0 V to +5 V	Engineering	0	+5000
		2's Complement	0000h	7FFFh
0A	0 V to +1 V	Engineering	0	+10000
		2's Complement	0000h	7FFFh
0B	0 mV to +500 mV	Engineering	0	+5000
		2's Complement	0000h	7FFFh

Note: If the Analog Format is set to "% FSR", the data of any type code ranges from +000.00 to +100.00.

B.5 Data Ranges for tM-AD8C

The tM-AD5C module only supports positive current (0-20 mA and 4-20 mA). Take the type code **0D** as an example, the data ranges from **0 to +20 mA** and the value ranges from 0 (0000h) to +32767 (**7**FFFh).

Type Code	Range	Data Format	Min.	Max.
07	4 mA to 20 mA	Engineering	4000	20000
		2's Complement	0000h	FFFFh
0D	0 mA to +20 mA	Engineering	0	20000
		2's Complement	0000h	7FFFh
1A	0 mA to 20 mA	Engineering	0	20000
		2's Complement	0000h	FFFFh

Note: If the Analog Format is set to "% FSR", the data of any type code ranges from +000.00 to +100.00.

B.6 Data Ranges for tM-AD4P2C2

The tM-AD4P2C2 module supports both positive and negative voltage/current, and the value ranges from -32768 to 32767 or from 0 to 65535.

Type Code	Range	Data Format	Min.	Max.
05	-2.5 V to +2.5 V	Engineering	-25000	+25000
		2's Complement	8000h	7FFFh
06	-20 mA to +20 mA	Engineering	-20000	+20000
		2's Complement	8000h	7FFFh
07	4 mA to 20 mA	Engineering	4000	20000
		2's Complement	0000h	FFFFh
08	-10 V to +10 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh
09	-5 V to +5 V	Engineering	-5000	+5000
		2's Complement	8000h	7FFFh
0A	-1 V to +1 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh
0D	-20 mA to +20 mA	Engineering	-20000	+20000
		2's Complement	8000h	7FFFh
1A	0 mA to 20 mA	Engineering	0	20000
		2's Complement	0000h	FFFFh

B.7 Data Ranges for tM-DA1P1R1

The tM-DA1P1R1 module supports positive voltage/current and the following value ranges.

Type Code	Range	Data Format	Min.	Max.
0	0 mA to 20 mA	Engineering	0	+20000
1	4 mA to 20 mA		4000	20000
2	0 V to 10 V		0	+10000
4	0 V to 5 V		0	+5000

Note: If the Analog Format is set to "2's Complement", the data of any type code ranges from 0000h to FFFFh, i.e., 0-65535.

B.8 Data Ranges for tM-TH8

The tM-TH8 module supports the following type codes and value ranges for the thermistor.

Type Code	Thermistor Type (Range)	Data Format	Min.	Max.
60	PreCon Type III 10K @ 25°C (-30 °F to 240 °F)	Engineering	-3000	24000
		2's Complement	F000	7FFF
		% FSR	-012.50	+100.00
		Ohms	+000539.4	+173600.0
61	Fenwell Type U 2K @ 25°C (-50 °C to 150 °C)	Engineering	-5000	15000
		2's Complement	D556	7FFF
		% FSR	-033.33	+100.00
		Ohms	+000037.2	+134020.0
62	Fenwell Type U 2K @ 25°C (0 °C to 150 °C)	Engineering	0	15000
		2's Complement	0000	7FFF
		% FSR	+000.00	+100.00
		Ohms	+000037.2	+006530.0
63	YSI L Mix 100 @ 25°C (-80 °C to 100 °C)	Engineering	-8000	10000
		2's Complement	999A	7FFF
		% FSR	-080.00	+100.00
		Ohms	+000014.3	+014470.0
64	YSI L Mix 300 @ 25°C (-80 °C to 100 °C)	Engineering	-8000	10000
		2's Complement	999A	7FFF
		% FSR	-080.00	+100.00
		Ohms	+000035.8	+067660.0
65	YSI L Mix 1000 @ 25°C (-70 °C to 100 °C)	Engineering	-7000	10000
		2's Complement	A667	7FFF
		% FSR	-070.00	+100.00
		Ohms	+000106.4	+132600.0
66	YSI B Mix 2252 @ 25°C (-50 °C to 150 °C)	Engineering	-5000	15000
		2's Complement	D556	7FFF
		% FSR	-033.33	+100.00
		Ohms	+000041.8	+151000.0
67	YSI B Mix 3000 @ 25°C (-40 °C to 150 °C)	Engineering	-4000	15000
		2's Complement	DDDE	7FFF
		% FSR	-026.67	+100.00
		Ohms	+000055.6	+101000.0

Type Code	Thermistor Type (Range)	Data Format	Min.	Max.
68	YSI B Mix 5000 @ 25°C (-40 °C to 150 °C)	Engineering	-4000	15000
		2's Complement	DDDE	7FFF
		% FSR	-026.67	+100.00
		Ohms	+000092.7	+168300.0
69	YSI B Mix 6000 @ 25°C (-30 °C to 150 °C)	Engineering	-3000	15000
		2's Complement	E667	7FFF
		% FSR	-020.00	+100.00
		Ohms	+000111.5	+106200.0
6A	YSI B Mix 10000 @ 25°C (-30 °C to 150 °C)	Engineering	-3000	15000
		2's Complement	E667	7FFF
		% FSR	-020.00	+100.00
		Ohms	+000185.9	+177000.0
6B	YSI H Mix 10000 @ 25°C (-30 °C to 150 °C)	Engineering	-3000	15000
		2's Complement	E667	7FFF
		% FSR	-020.00	+100.00
		Ohms	+000237.0	+135200.0
6C	YSI H Mix 30000 @ 25°C (-10 °C to 200 °C)	Engineering	-1000	20000
		2's Complement	F99A	7FFF
		% FSR	-005.00	+100.00
		Ohms	+000186.7	+158000.0
70 to 77	User-defined (-50 °C to 150 °C)	Engineering	-5000	15000
		2's Complement	D556	7FFF
		% FSR	-033.33	+100.00
		Ohms	+000000.0	+000000.0

Note: For "User-defined" types, if the resistance is larger than 204800 ohms, it will be treated as Under Range.

Analog Format	Under Range	Over Range	Protocol
Engineering	-9999.9	+9999.9	DCON
	-32768	32767	Modbus RTU/ASCII
2's Complement	8000	7FFF	DCON Modbus RTU/ASCII
Percent (% FSR)	-999.99	+999.99	DCON

B.8.1 Steinhart–Hart Resistance (User-defined Type)

The method to get the Steinhart–Hart coefficients of the thermistor:

1. To get Steinhart-Hart coefficients from the thermistor vendor.
2. Using the Resistance-Temperature table released by the thermistor vendor. Choose three pairs of resistance values and temperatures such as (R1, T1), (R2, T2), and (R3, T3) to calculate coefficients.

Refer to Section 3.4.1 Configuration Page - User Defined Type to calculate Steinhart–Hart coefficients and convert them to IEEE-754 form by using DCON Utility Pro.

For a typical thermistor, the relationship between resistance and temperature can be expressed by the Steinhart–Hart equation:

$$1/T = A + B \ln R_T + C (\ln R_T)^3$$

where R_T is the resistance in ohms at temperature **T** in degrees Kelvin (K = °C + 273.15). The values of A, B, and C are called Steinhart Coefficients. The error of the equation is less than +/- 0.01°C in a 100°C span. The tM-TH8 supports user-defined types by specifying the Steinhart coefficients using the @AASxTttC(data) command.

The data sent is a 32-bit hexadecimal value in the IEEE-754 standard format:

Bit	Description
31 (sign)	0: positive ; 1 : negative
30-23 (exponent)	The exponent base is 2. The actual exponent is calculated by subtracting 127 from the stored value.
22-00 (mantissa)	The mantissa is expressed as “1.f” where f is the fractional part and is stored in this field

For example, a hex. value **C3694000h** also equals to 1100 0011 0110 1001 0100 0000 0000 0000 (2)

C	3	6	9	4	0	0	0
1100	0011	0110	1001	0100	0000	0000	0000

Bit 31 is **1**, indicating a negative number.

Bit 30 to 23 **10000110**$_{(2)}$ or 134$_{(10)}$, the exponent is 7 (i.e., 134 – 127 =7).

Bit 22 to 00 The mantissa is **1.10100101000000000000000**$_{(2)}$. After adjusting the mantissa with the exponent (7), the binary is 11101001.0100000000000000. That is 233.25$_{(2)}$.

So, the floating-point number of “C3694000h” is -233.25.

Appendix C Modbus Register Mapping

For AI, AO, DI, and DO Modules

<table>
<tr><th>Address
(Base 1)</th><th>R/W</th><th colspan="2">Description</th></tr>
<tr><td>00257</td><td rowspan="2">R/W</td><td>The protocol to be used</td><td>0: DCON, 1: Modbus RTU</td></tr>
<tr><td>00258</td><td>The protocol to be used</td><td>0: Determined by 00257;
1: Modbus ASCII</td></tr>
<tr><td>00273</td><td rowspan="5">R</td><td>The module resets the status</td><td>0: Not yet rebooting
1: Ever reboot or first power-on</td></tr>
<tr><td>40481</td><td colspan="2">The firmware version (Low Word)</td></tr>
<tr><td>40482</td><td colspan="2">The firmware version (High Word)</td></tr>
<tr><td>40483</td><td colspan="2">The module name (Low Word)</td></tr>
<tr><td>40484</td><td colspan="2">The module name (High Word)</td></tr>
<tr><td>40485</td><td rowspan="6">R/W</td><td>The module address</td><td>The valid range: 1 to 247</td></tr>
<tr><td>40486</td><td colspan="2">Bits 5: 0
Baud Rate: 0x03 to 0x0A

Bits 7: 6
00: No Parity, 1 Stop Bit
01: No Parity, 2 Stop Bit
10: Even Parity, 1 Stop Bit
11: Odd Parity, 1 Stop Bit
<table>
<tr><th>Code</th><th>Baud Rate</th></tr>
<tr><td>0x03</td><td>1200</td></tr>
<tr><td>0x04</td><td>2400</td></tr>
<tr><td>0x05</td><td>4800</td></tr>
<tr><td>0x06</td><td>9600</td></tr>
<tr><td>0x07</td><td>19200</td></tr>
<tr><td>0x08</td><td>38400</td></tr>
<tr><td>0x09</td><td>57600</td></tr>
<tr><td>0x0A</td><td>115200</td></tr>
</table></td></tr>
<tr><td>40488</td><td>The Modbus response delay (ms)</td><td>Valid range: 0 to 30</td></tr>
<tr><td>00261</td><td>The Host watchdog</td><td>0: Disable, 1: Enable</td></tr>
<tr><td>00270</td><td>The Host watchdog timeout status</td><td>Set to “1” to clear the status</td></tr>
<tr><td>40489</td><td>The Host watchdog timeout value</td><td>Valid range: 0 to 255 (0.1s)</td></tr>
<tr><td>40492</td><td>The Host watchdog timeout count</td><td>Set to “0” to clear the value</td></tr>
</table>

For AI/AO Modules

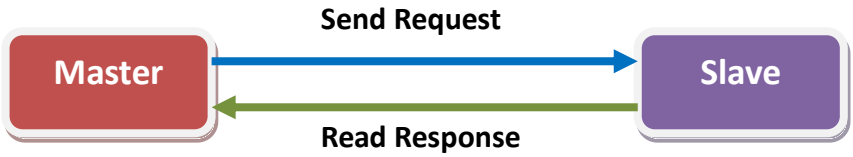
Address (Base 1)	R/W	Description	
30001 - 30008 40001 - 40008	R	The AI values for channels 0 to 7	
40490	R/W	Enable/Disable the AI channels (n)	Valid range: 00h to (2^n-1)h
00269		The Modbus data format	0: Hex., 1: Engineering
00271		The sampling mode (0: Normal, 1: Fast)	Not for tM-TH8/DA1P1R1
00129 - 00136 10129 - 10136	R	The over/under range status for the AI channels for +4 to +20 mA or 0 to +20 mA	For tM-AD2/AD5C/AD8C
00225 - 00228 10225 - 10228			For tM-AD4P2C2
40257 - 40264	R/W	The Type code for the AI channels	For tM-AD2/TH8/AD4P2C2
40487			For tM-AD5/AD8/AD5C/AD8C
For tM-AD4P2C2			
40225 - 40226	R/W	The upper alarm limit for AI channels 0 and 1	
40233 - 40234		The lower alarm limit of AI channels 0 and 1	
For tM-AD2			
40494	R/W	The lower threshold in +4 to +20 mA	Valid range: 0 to 40 (0.1mA)
For tM-TH8			
40385 - 40392	R/W	The resistance offset for AI channels 0 to 7 at 0.1 ohms	Valid range: 0 to 255
40449 - 40456		The temperature offset of AI channels 0 to 7 at 0.1 °C	Valid range: -128 to 127
40769 - 40784		The Steinhart coefficient A for type codes 70 to 77	
40801 - 40816		The Steinhart coefficient A for type codes 70 to 77	
40833 - 40848		The Steinhart coefficient A for type codes 70 to 77	
00267		The temperature format	0: Fahrenheit, 1: Celsius
00272	W	Set to “1” to load the factory calibration parameters	
For tM-DA1P1R1			
40033	R/W	The AO value for channel 0	
30065, 40065	R	The Read-back value for the AO channel	
40097	R/W	The Safe value for the AO channel	
40193		The Power-on value for the AO channel	
40289		The Slew rate for the AO channel	
40417		The Type code for the AO channel	

For DI/DO Modules

Address (Base 1)	R/W	Description	
00033 - 00040 10033 - 10040	R	The DI values for channels 0 to 7	
00001 - 00008	R/W	The DO values for channels 0 to 7	
00265		The current DI state	0: Normal (ON/OFF returns 1/0) 1: Inverse (ON/OFF returns 0/1)
00266		The current DO state	
00065 - 00072 10065 - 10072	R	The high latched value for the DI channels	
00097 - 00104 10097 - 10104		The low latched value for the DI channels	
00073 - 00080 10073 - 10080		The high latched value for the DO channels	
00105 - 00112 10105 - 10112		The low latched value for the DO channels	
00264	W	Set to “1” to clear the DIO latched value	
30001 - 30008	R	The Counter values of the DI channel (16-bit, 0 to 65535)	For DI modules
30129 - 30130 40129 - 40130			For tM-DA1P1R1/ AD4P2C2
00193 - 00200	R/W	The update mode for the edge-triggered Counter 0: The counter updates when there is a falling-edge signal in the DI channel. 1: The counter updates when there is a rising-edge signal in the DI channel.	
00513 - 00520	W	Set to “1” to Clear the Counter Value	
00260	R/W	The host watchdog mode 0: If a timeout occurs, the Timeout status must be cleared to write the AO/DO value 1: If a timeout occurs, the AO/DO value can still be written and the Timeout status will be automatically reset.	
00129 - 00136		The Safe value for the DO channels	
00161 - 00168		The Power-on value for the DO channels	
For tM-AD4P2C2			
00289 - 00290	R/W	The low alarm status for channels 0 and 1 Set to 1 to clear the low latched alarm	
00305 - 00306		The high alarm status for channels 0 and 1 Set to 1 to clear the high latched alarm	
00321 - 00322		Enable/Disable the alarm on channels 0 and 1	
00337 - 00338		The alarm type for channels 0 and 1, which can be either momentary or latched	

Appendix D Modbus RTU Protocol Commands

Modbus is a communication protocol developed by MODICON Inc. in 1979. It's a standard, truly open, and the most widely used network communication protocol in the industrial automation field. SCADA and HMI software can easily integrate serial devices via Modbus protocol. Visit the website https://www.modbus.org for more information about Modbus.



Data structure of Modbus RTU:

Byte 00	Byte 01	Byte 02-03	Byte 04-05	CRC
Slave address (Net ID, **1 to 247**)	Function Code	Data		Checksum (2 Bytes)
		Start Address	Channels	

Note: 1) The content of CRC will be omitted in the following section.
2) Refer to Appendix C Modbus Register Mapping to look up the start address.

The following Modbus RTU functions are supported by tM series module:

Function Code	Bit	Description		Reference Address
01 (0x01)	1	Read Coil	Read the DO status	**0x**xxx
02 (0x02)	1	Read Discrete Input	Read the DI status	**1**xxxx
03 (0x03)	16	Read Multiple Registers	Read the AO value	**4**xxxx
04 (0x04)	16	Read Multiple Input Registers	Read AI values	**3**xxxx
05 (0x05)	1	Force Single Coil	Write a DO status	**0x**xxx
06 (0x06)	1	Preset Single Register	Write an AO value	**4**xxxx
15 (0x0F)	16	Force Multiple Coils	Write multiple DO status	**0x**xxx
16 (0x10)	16	Preset Multiple Registers	Write multiple AO values	**4**xxxx

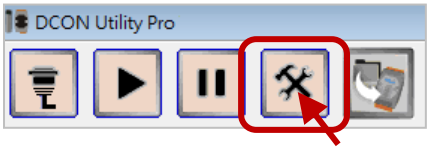
Note:

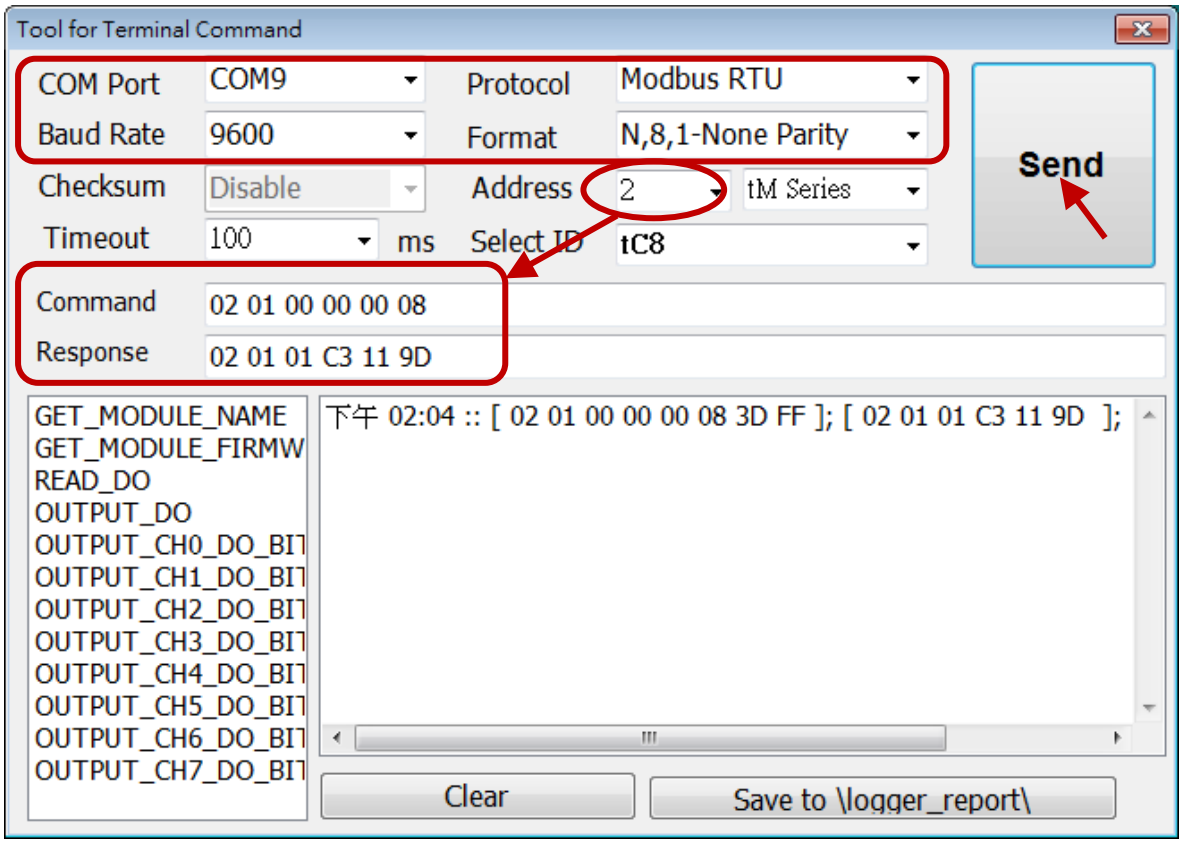
Modbus address **0x**xxx can be used to read/write the status of **DO** channels.
Modbus address **1**xxxx can be used to read the status of **DI** channels.
Modbus address **3**xxxx can be used to read the value of **AI** channels.
Modbus address **4**xxxx can be used to read/write the value of **AO** channels.

Exception Code

Exception Code (Hex.)	Name	Description
01	Illegal Function	Not supported function
02	Illegal Data Address	Invalid address
03	Illegal Data Value	Invalid data
04	Slave Device Failure	Fails to connect the slave device
05	Acknowledge	Inform the command is processing
06	Slave Device Busy	The slave device is busy

DCON Utility Pro provides the **Command Line** function for testing the Modbus command.





Command: 02 01 00 00 00 08

02	01	00	00	00	08
module ID = 02	Read DO	Start address = 0000 (Base 0)		8-channel to be read	

Response: 02 01 01 C3 11 9D

02	01	01	C3	11	9D
module ID = 02	Read DO	Data byte is 1	1100 0011 means DO 0, 1, 6, 7 = ON	CRC	

D.1 01 (0x01) Read Coil

The function code is used to read the DO or Coil status.

[Request]

Slave Address	**Function Code**	**Start Address** (High, Low)	**Channels** (High, Low)
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x01	See Appendix C Modbus Register Mapping	tM module: Max. 8-channel

[Response]

Slave Address	**Function Code**	**Byte Count**	**Data**
1 Byte	1 Byte	1 Byte	**n** Byte
1 to 247	0x01	**n** = (Channels+7)/8	n = 1; Byte 03 = bit 7 - 0 n = 2; Byte 04 = bit 15 - 8 ..... n = m; Byte m+2 = bit (8m-1) - 8(m-1)

[Error Response]

Slave Address	**Function Code**	**Exception Code**
1 Byte	1 Byte	1 Byte
1 to 247	0x81 (80+ Function Code)	See "Exception Code"

Examples:

Request (Master): **01 01 00 00 00 02** (Hex.)

Slave Address	**Function Code**	**Start Address**		**Channels**	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	**01**	**00**	**00**	**00**	**02**
Module ID = 01	Read DO Status	Address = **0**0000 (Base 0)		Read two channels	

Note: The base address for tM series module is **0** (i.e., base 0) so the start address must be set to the Modbus registers decrease 1, refer to Appendix C.

Response (Slave): **01 01 01 03** (Hex.)

Slave Address	**Function Code**	**Byte Count**	**Data**
Byte 00	Byte 01	Byte 02	Byte 03
01	**01**	**01**	**03**
Module ID = 01	Read DO Status	1 Byte	"0000 0011" means DO0, 1 = ON

D.2 02 (0x02) Read Discrete Inputs

The function code is used to read the DI status.

[Request]

Slave Address	**Function Code**	**Start Address** (High, Low)	**Channels** (High, Low)
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x02	See Appendix C Modbus Register Mapping	tM module: Max. 8-channel

[Response]

Slave Address	**Function Code**	**Byte Count**	**Data**
1 Byte	1 Byte	1 Byte	**n** Byte
1 to 247	0x02	**n** = (Channels+7)/8	n = 1; Byte 03 = bit 7 - 0 n = 2; Byte 04 = bit 15 - 8 ..... n = m; Byte m+2 = bit (8m-1) - 8(m-1)

[Error Response]

Slave Address	**Function Code**	**Exception Code**
1 Byte	1 Byte	1 Byte
1 to 247	0x82 (80+ Function Code)	See "Exception Code"

Examples:

Request (Master): **01 02 00 32 00 02** (Hex.)

Slave Address	**Function Code**	**Start Address**		**Channels**	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	**02**	**00**	**32**	**00**	**02**
Module ID = 01	Read DI Status	Address = **1**0032 (Base 0)		Read two channels	

Note: The base address for tM series module is **0** (i.e., base 0) so the start address must be set to the Modbus registers decrease 1, refer to Appendix C.

Response (Slave): **01 02 01 03** (Hex.)

Slave Address	**Function Code**	**Byte Count**	**Data**
Byte 00	Byte 01	Byte 02	Byte 03
01	**02**	**01**	**03**
Module ID = 01	Read DI Status	1 Byte	0000 0011 means DO0, 1 = ON

D.3 03 (0x03) Read Multiple Registers

The function code is used to read multiple AO or Holding Registers values.
Note: **tM-DA1P1R1 supports a single analog output channel.**

[Request]

Slave Address	Function Code	Start Address (High, Low)	Channels (High, Low)
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x03	See Appendix C Modbus Register Mapping	Word (16-bit Register)

[Response]

Slave Address	Function Code	Byte Count	Data
1 Byte	1 Byte	1 Byte	n Byte
1 to 247	0x03	n = Channels * 2 Bytes	n = m; Byte 03 = High Byte Byte 04 = Low Byte … Byte m+1 = High Byte Byte m+2 = High Byte

[Error Response]

Slave Address	Function Code	Exception Code
1 Byte	1 Byte	1 Byte
1 to 247	0x83 (80+ Function Code)	See "Exception Code"

Examples:

Request (Master): **01 03 01 ED 00 01** (Hex.)

Slave Address	Function Code	Start Address		Channels	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	**03**	**01**	**ED**	**00**	**01**
Module ID = 01	Read Holding Registers	Address = **4**0493 (Base 0)		Read one channel	

Note: The base address for tM series module is **0** (i.e., base 0) so the start address must be set to the Modbus registers decrease 1, refer to Appendix C.

Response (Slave): **01 03 02 00 1E** (Hex.)

Slave Address	Function Code	Byte Count	Data	
Byte 00	Byte 01	Byte 02	Byte 03 (H)	Byte 04 (L)
01	**03**	**02**	**00**	**1E**
Module ID = 01	Read Holding Registers	2 Bytes	If the current is less than 3mA, it will be treated as Under Range	

D.4 04 (0x04) Read Multiple Input Registers

The function code is used to read multiple AI, Input Registers, or Counter values.

[Request]

Slave Address	**Function Code**	**Start Address** (High, Low)	**Channels** (High, Low)
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x04	See Appendix C Modbus Register Mapping	Word (16-bit Register)

[Response]

Slave Address	**Function Code**	**Byte Count**	**Data**
1 Byte	1 Byte	1 Byte	**n** Byte
1 to 247	0x04	**n** = Channels * 2 Bytes	n = m; Byte 03 = High Byte Byte 04 = Low Byte .... Byte m+1 = High Byte Byte m+2 = Low Byte

[Error Response]

Slave Address	**Function Code**	**Exception Code**
1 Byte	1 Byte	1 Byte
1 to 247	0x84 (80+ Function Code)	See "Exception Code"

Examples:

Request (Master): **01 04 00 07 00 01** (Hex.)

Slave Address	**Function Code**	**Start Address**		**Channels**	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	**04**	**00**	**07**	**00**	**01**
Module ID = 01	Read Input Registers	Address = **3**0007 (Base 0)		Read one channel	

Note: The base address for tM series module is **0** (i.e., base 0) so the start address must be set to the Modbus registers decrease 1, refer to Appendix C.

Response (Slave): **01 04 02 00 05** (Hex.)

Slave Address	**Function Code**	**Byte Count**	**Data**	
Byte 00	Byte 01	Byte 02	Byte 03 (H)	Byte 04 (L)
01	**04**	**02**	**00**	**05**
Module ID = 01	Read Input Registers	2 Bytes	The counter value of DI7 is 5	

D.5 05 (0x05) Force Single Coil

The function code is used to write a single DO or Coil status.

[Request]

Slave Address	Function Code	Start Address (High, Low)	Data (High, Low)
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x05	See Appendix C Modbus Register Mapping	0x**FF**00 means set to ON. 0x**00**00 means set to OFF. (All other values are invalid.)

[Response]

Slave Address	Function Code	Start Address	Data
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x05	The same as the Byte 02-03 of the request	The same as the Byte 04-05 of the request

[Error Response]

Slave Address	Function Code	Exception Code
1 Byte	1 Byte	1 Byte
1 to 247	0x85 (80+ Function Code)	See "Exception Code"

Examples:

Request (Master): **02 05 00 03 FF 00** (Hex.)

Slave Address	Function Code	Start Address		Data	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
02	**05**	**00**	**03**	**FF**	**00**
Module ID = 02	Write DO Status	Address = **0**0003 (Base 0)		Set to ON	

Note: The base address for tM series module is **0** (i.e., base 0) so the start address must be set to the Modbus registers decrease 1, refer to Appendix C**.**

Response (Slave): **02 05 00 03 FF 00** (Hex.)

Slave Address	Function Code	Start Address		Data	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
02	**05**	**00**	**03**	**FF**	**00**
Module ID = 02	Write DO Status	**0**0003 means DO3		Set to ON	

D.6 06 (0x06) Preset Single Register

The function code is used to write a single AO or Holding Registers value.

[Request]

Slave Address	**Function Code**	**Start Address** (High, Low)	**Data** (High, Low)
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x06	See Appendix C Modbus Register Mapping	Set the output value

[Response]

Slave Address	**Function Code**	**Start Address**	**Data**
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x06	The same as the Byte 02-03 of the request	The same as the Byte 04-05 of the request

[Error Response]

Slave Address	**Function Code**	**Exception Code**
1 Byte	1 Byte	1 Byte
1 to 247	0x86 (80+ Function Code)	See "Exception Code"

Examples:

Request (Master): **01 06 01 E7 00 0A** (Hex.)

Slave Address	**Function Code**	**Start Address**		**Data**	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	**06**	**01**	**E7**	**00**	**0A**
Module ID = 01	Write Holding Register	Address = **4**0487 (Base 0)		Set to 10	

Note: The base address for tM series module is **0** (i.e., base 0) so the start address must be set to the Modbus registers decrease 1, refer to Appendix C.

Response (Slave): **01 06 01 E7 00 0A** (Hex.)

Slave Address	**Function Code**	**Start Address**		**Data**	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	**06**	**01**	**E7**	**00**	**0A**
Module ID = 01	Write Holding Register	Modbus Respond Delay is set to 10 (ms)			

D.7 15 (0x0F) Force Multiple Coils

The function code is used to write multiple DO or Coil values.

[Request]

Slave Address	Function Code	Start Address	Channels	Byte Count	Data
1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	n Byte
1 to 247	0x0F	Refer to Appendix C Byte 02 = High Byte Byte 03 = Low Byte	Byte 04 = High Byte Byte 05 = Low Byte	n = (Channels+7)/8	A bit corresponds to a channel. 1: ON ; 0: OFF. n = 1; Byte 07 = bit 7 - 0 n = 2; Byte 08 = bit 15 - 8 … n = m; Byte m+6 = bit (8m-1) - 8(m-1)

[Response]

Slave Address	Function Code	Start Address	Channels
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x0F	The same as the Byte 02-03 of the request	The same as the Byte 04-05 of the request

[Error Response]

Slave Address	Function Code	Exception Code
1 Byte	1 Byte	1 Byte
1 to 247	0x8F (80+ Function Code)	See "Exception Code"

Examples:

Request (Master): **02 0F 00 03 00 05 01 1F** (Hex.)

Slave Address	Function Code	Start Address		Channels		Byte Count	Data
Byte 00	Byte 01	Byte 02	Byte 03	Byte 04	Byte 05	Byte 06	Byte 07
02	**0F**	**00**	**03**	**00**	**05**	**01**	**1F**
Module ID = 02, write multiple DO. Set five bits to ON from the address **0**0003 (i.e., DO3).							

Note: The base address for tM series module is **0** (i.e., base 0) so the start address must be set to the Modbus registers decrease 1, refer to Appendix C.

Response (Slave): **02 0F 00 03 00 05** (Hex.)

Slave Address	Function Code	Start Address		Channels	
Byte 00	Byte 01	Byte 02	Byte 03	Byte 04	Byte 05
02	**0F**	**00**	**03**	**00**	**05**
Module ID = 02	Write DO Status	Start from DO3		Five bits are set	

D.8 16 (0x10) Set Multiple Register

The function code is used to write multiple AO or Holding Registers values.

[Request]

Slave Address	Function Code	Start Address	Channels	Byte Count	Data
1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	**n** Byte
1 to 247	0x10	Byte 02 = High Byte Byte 03 = Low Byte	Byte 04 = High Byte Byte 05 = Low Byte	(Word number) **n** = Channels * 2 Bytes	n = m; Byte 07 = High Byte Byte 08 = Low Byte … Byte m+5 = High Byte Byte m+6 = Low Byte

[Response]

Slave Address	Function Code	Start Address	Channels
1 Byte	1 Byte	2 Bytes	2 Bytes
1 to 247	0x10	The same as the Byte 02-03 of the request	The same as the Byte 04-05 of the request

[Error Response]

Slave Address	Function Code	Exception Code
1 Byte	1 Byte	1 Byte
1 to 247	0x90 (80+ Function Code)	See "Exception Code"

Examples:

Request (Master): **03 10 01 C0 00 02 04 00 0A 00 0A** (Hex.)

Slave Address	Function Code	Start Address		Channels		Byte Count	Data
Byte 00	Byte 01	Byte 02	Byte 03	Byte 04	Byte 05	Byte 06	Byte 07 - 10
03	**10**	**01**	**C0**	**00**	**02**	**04**	**000A 000A**
Module ID = 03, write multiple Holding Registers. Set the temperature offset of two channels to 10 (1℃) from address **4**0448 (base 0).							

Note: The base address for tM series module is **0** (i.e., base 0) so the start address must be set to the Modbus registers decrease 1, refer to Appendix C.

Response (Slave): **03 10 01 C0 00 02** (Hex.)

Slave Address	Function Code	Start Address		Channels	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
03	**10**	**01**	**C0**	**00**	**02**
Module ID = 03	Write Holding Registers	The temperature offset of the AI0 and AI1 of tM-TH8 are set to 1 (℃)			

Appendix E DCON Protocol Commands

All communication with tM modules consists of commands generated by the host and responses transmitted by the tM modules. Each module has a unique ID number that is used for addressing purposes and is stored in non-volatile memory.

The ID is 01 by default and can be changed using a user command. All commands to the modules contain the ID address, meaning that only the addressed module will respond. Except for both the ~** and #** commands that will be sent to all modules, in these cases, the modules do not reply to the command.

Request Command Format:

Leading	Address	Command	[CHK]	CR

Response Format:

Leading	Address	Response	[CHK]	CR

Leading	Character, like ~, $, #, @, %, !, ?, >.
Address	The slave address of a module that can be set by using DCON Utility Pro, refer to Chapter 3.
Command/Response	Command with parameters or response with data.
CHKSUM	2-character checksum. If the Checksum is disabled, no [CHK] here. (Reference: Section E.1 %AANNTTCCFF Command)
CR	The end-of-command character, character return(0Dh)

Calculate Checksum:

1. Sum the ASCII code of all the characters contained in the command in addition to the 'CR' terminator.
2. The Checksum is the last 2 digits of the sum value expressed in Hexadecimal format.

For example, if the string of request command is **$012**(CR)

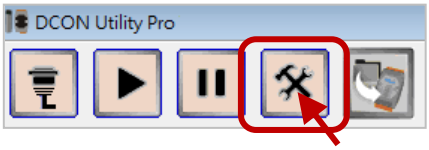
1. Sum = "$"+ "0"+ "1"+ "2" = 24h + 30h + 31h + 32h = **B7h**
2. Checksum = "**B7**"
3. The request command with the checksum is $012B7(CR)

If the string of response command is **!01200600**(CR)

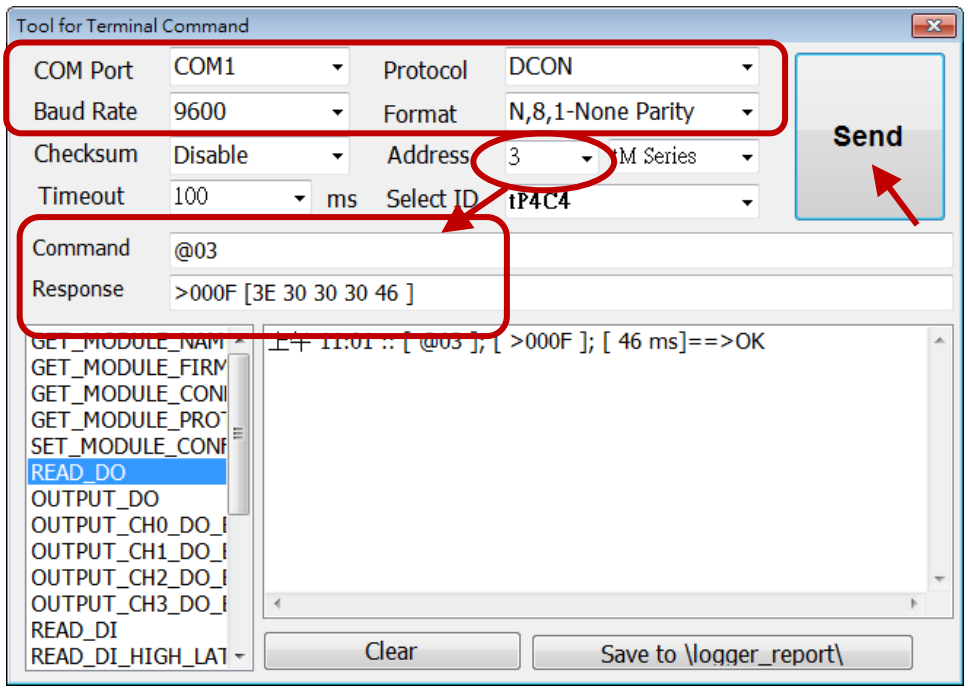
1. Sum = "!" + "0" + "1" + "2" + "0" + "0" + "6" + "0" + "0"
   = 21h + 30h + 31h + 32h + 30h + 30h + 36h + 30h + 30h = **1AAh**
2. Checksum = "**AA**"
3. The response command with the checksum is = !01200600AA(CR)

Note: All characters should be in capital letters.

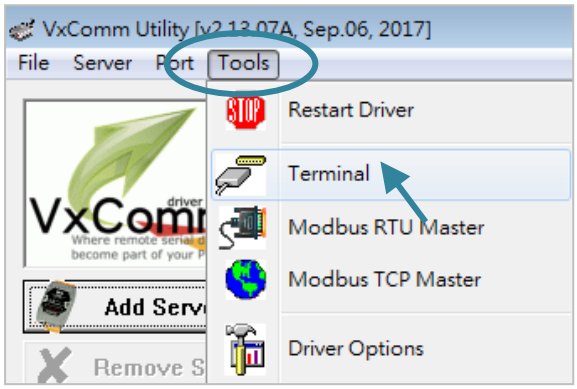
Users can test the DCON protocol command by using the **Command Line** function of DCON Utility Pro or the Terminal function of VxComm Utility.

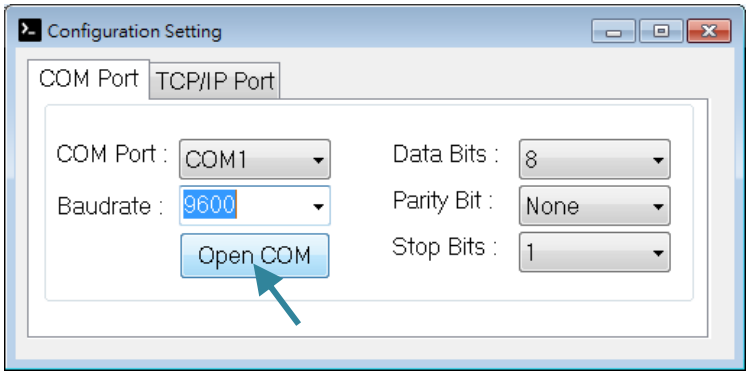


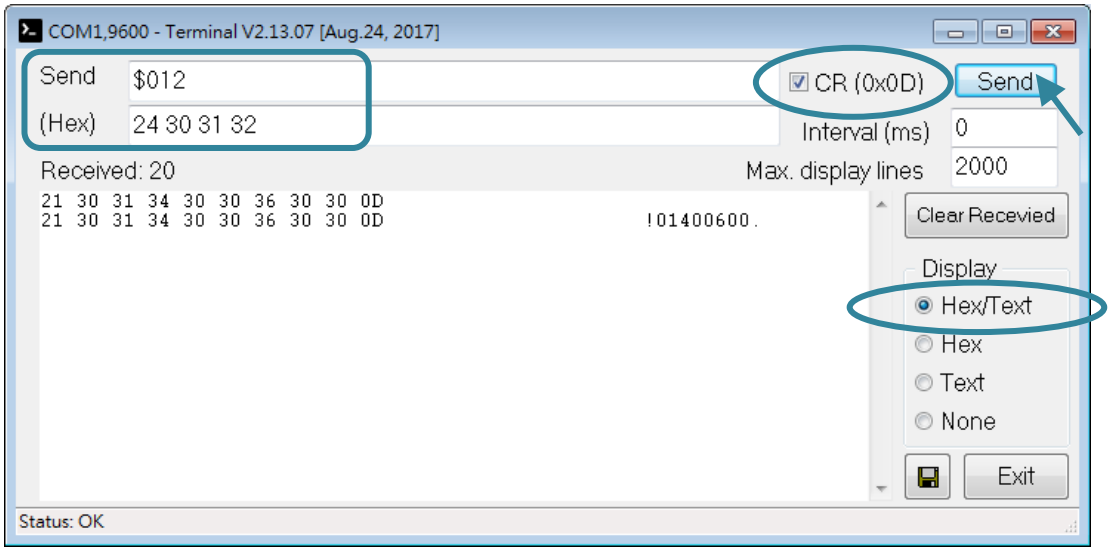
DCON Utility Pro



VxComm Utility







E.1 General Commands

Command	Response	Description
$AA2	!AATTCCFF	Read the configuration of a module
%AANNTTCCFF	!AA	Set the communication parameters of a module
$AA8Ci	!AACiRrr	Read the type code for an AI channel
$AA7CiRrr	!AA	Set the type code for an AI channel
$AA9N	!AATS	Read the type code and slew rate for an AO channel
$AA9NTS	!AA	Set the type code and slew rate for an AO channel
$AA5	!AAS	Read the reset status of a module
$AAF	!AA(Data)	Read the firmware version of a module
$AAM	!AA(Data)	Read the name of a module
$AAP	!AASC	Read the communication protocol of a module
$AAPN	!AA	Set the communication protocol of a module
$AAI	!AAS	Read the INIT terminal status of a module
~AARD	!AAVV	Read the response delay time value of a module
~AARDVV	!AA	Set the response delay time value of a module
Only available for tM-TH8		
~AAI	!AA	The Soft Init command
~AATnn	!AA	Set the Soft Init timeout

E.2 I/O Commands

Command	Response	Description
$AA6	!AAVV	Read the status of AI/DI/DO channels (The format of response DIO data can be queried in this section)
$AA5VV	!AA	Set the status of AI channels
#**	No Response	Notify all modules to synchronously access the DIO status
$AA4	!S(Data)	Read the synchronous sampling DIO status
@AA	>(Data)	Read the status of DI/DO channels
@AADI	!AA0OOII	Read the status of DI/DO channels (for Multi-function module)
@AA(Data)	>	Set the status of DO channels
@AADODD	!AA	Set the status of DO channels (for Multi-function module)

Command	Response	Description
#AA00(Data)	>	Set the status of DO channels (DO0 to DO7)
#AA0A(Data)	>	Set the status of DO channels (DO0 to DO7)
#AA0B(Data)	>	Set the status of DO channels (DO8 to DO15)
#AA1cDD	>	Set the status of a DO channel (DO0 to DO7)
#AAAcDD	>	Set the status of a DO channel (DO0 to DO7)
#AABcDD	>	Set the status of a DO channel (DO8 to DO15)
#AA	>(Data)	Read data from all AI channels
#AAN	>(Data)	Read data from an AI channel
$AAA	>(Data)	Read hexadecimal data from all AI channels
$AA8N	!AA(Data)	Read data from an AO channel
$AA6N	!AA(Data)	Read the last written data from an AO channel
#AAN(Data)	>	Set the value of an AO channel

E.3 Host Watchdog, Power-on Value, and Safe Value Commands

Command	Response	Description
~**	No Response	Notify all modules that the Host works normally
~AA0	!AASS	Read the host watchdog status of a module
~AA1	!AA	Reset the host watchdog status of a module
~AA2	!AAETT	Read the host watchdog timeout value of a module
~AA3EVV	!AA	Set the host watchdog and a timeout value
~AA4V	!AA(data)	Read the Power-on value or Safe value of the DO module
~AA5V	!AA	Set the DO value as the Power-on value or Safe value
~AA4	!AAPPSS	Read the Power-on value or Safe value (Multi-function-DO)
~AA5PPSS	!AA	Set the Power-on value or Safe value (Multi-function-DO)
$AA7N	!AA	Read the Power-on value of the AO module
$AA4N	!AA	Set the AO value as the Power-on value
~AA4N	!AA(data)	Read the Safe Value of the AO module
~AA5N	!AA	Set the AO value as the Safe Value

E.4 DI Latched and Counter Commands

Command	Response	Description
$AALS	!AA(data)	Read the latched status of DI channels
$AAC	!AA	Clear the latched status of DI channels
#AAN	!AA(data)	Read a counter value of the DI channel
$AACN	!AA	Clear a counter value of the DI channel
For Multi-function Modules		
@AARECi	!AA(data)	Read a counter value of the DI channel for tM-AD4P2C2
@AACECi	!AA	Clear a counter value of the DI channel for tM-AD4P2C2
@AARECN	!AA(data)	Read a counter value of the DI channel for tM-DA1P1R1
@AACECN	!AA	Clear a counter value of the DI channel for tM-DA1P1R1

E.5 Alarm Commands

Note: The following commands only available for tM-AD4P2C2.

Command	Response	Description
@AAEATCi	!AA	Enable the alarm function for the AI channel
@AADACi	!AA	Disable the alarm function for the AI channel
@AARACi	!AAS	Read the setting of alarm type
@AARAO	!AAHHLL	Read the status of high/low alarm
@AARHCi	!AA(data)	Read the value of high alarm limit
@AARLCi	!AA(data)	Read the value of low alarm limit
@AAHI(data)Ci	!AA	Set the value of high alarm limit
@AALO(data)Ci	!AA	Set the value of low alarm limit
@AACHCi	!AA	Clear the status of the high latched alarm
@AACLCi	!AA	Clear the status of the low latched alarm

E.6 Module Calibration Commands

Command	Response	Description
~AAEV	!AA	Enable/Disable the calibration (For analog input modules and tM-AD4P2C2)
$AA1	!AA	Perform zero calibration (V, A) (For analog input modules and tM-AD4P2C2)
$AA1Ci	!AA	Perform zero calibration (For tM-AD2)
$AA0	!AA	Perform span calibration (V, Ω) (For tM-AD5/AD8/TH8/AD4P2C2)
$AA0Ci	!AA	Perform span calibration (A) (For tM-AD2/AD5C/AD8C/AD4P2C2)
Only for the tM-DA1P1R1		
$AA0N	!AA	Perform zero calibration
$AA1N	!AA	Perform span calibration
$AA3NVV	!AA	Adjust AO value for calibration
Only for the tM-TH8		
$AAS1	!AA	Reload the default calibration parameters

E.7 User-defined Type Commands (for Thermistor)

Note: The following commands only support the tM-TH8.

Command	Response	Description
@AAA3Ci	!AA(data)	Read the temperature offset of a channel
@AAA2CiToo	!AA	Set the temperature offset of a channel
@AAA7Ci	!AA(data)	Read the resistance offset of a channel
@AAA6CiRrr	!AA	Set the resistance offset of a channel
~AAD	!AAT	Read the temperature scale (℃/℉)
~AADT	!AA	Set the temperature scale (℃/℉)
@AAGxTtt	!AA(data)	Read the Steinhart-Hart coefficient of a user-defined type
@AASxTttC(data)	!AA	Set the Steinhart-Hart coefficient of a user-defined type
@AARTTttR(Data)	!AA(data)	Read the scaling temperature based on input resistance of a user-defined type

$AA2 Read the configuration of a module

Syntax:

$AA2[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

2 Command to read module settings

Response:

Valid Command: **!AATTCCFF** [CHKSUM] (CR)

Invalid Command: **?AA** [CHKSUM] (CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

TT Read the type code, refer to Appendix B.

Note: The Type Code for each channel of tM-AD2, tM-TH8, tM-DA1P1R1, and tM-AD4P2C2 modules must be set individually so the return value of the TT is **"00"**. For DIO modules, the return value is fixed to "**40**".

CC The Parity and Baud Rate settings.

FF The Checksum, Sample Mode, and Analog Format settings for analog modules.
The Checksum and Counter Update Direction settings for digital modules.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $012	Read the configuration of module 01
Response: !01400600	Module 01 settings: Parity= N,8,1-no parity, Baud Rate=9600, Checksum=Disable, Sample Mode=Normal, Analog Format=Engineering.

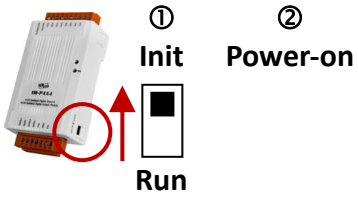
Command: $022	Read the configuration of module 02
Response: !02000602	Module 02 settings: Parity= N,8,1-no parity, Baud Rate=9600, Checksum=Disable, Analog Format= 2's Complement

Related Commands: %AANNTTCCFF

Related Sections: 3.2 Configuration Page - Configuration

%AANNTTCCFF Set the communication parameters of a module

Note: The settings for parity, baud Rate, and checksum must be configured in **INIT mode**. And then, adjust the switch to Run position and reboot to apply the settings.



Syntax:

%AANNTTCCFF[CHKSUM](CR)

% Delimiter Character

AA The address of the module (Hex., 00 to FF)

NN Set an address to the module (Hex., 00 to FF)

TT The type code setting, refer to Appendix B Type Codes and Ranges

Note: Using the $AA7CiRrr command to set the type code for each channel when the model is tM-AD2, tM-TH8, or tM-AD4P2C2. Using the $AA9NTS command when the model is tM-DA1P1R1. And, set "TT" as "**00**". When using DIO modules, set "TT" as "**40**".

CC Set a parity and baud rate

(*), it stands for items that need to be configured in Init mode.

Bit	7	6	5	4	3	2	1	0
[CC]	Parity		Baud Rate					
Parity (* Init)	0		1		2		3	
	N,8,1-no parity		N,8,2-no parity		E,8,1-even parity		O,8,1-odd parity	
Baud Rate (* Init)	3	4	5	6	7	8	9	A
	1200	2400	4800	9600	19200	38400	57600	115200

FF **For using Analog module:** Set the Checksum, Sample Mode, and Analog Format.

Bit	7	6	5	4	3	2	1	0
[FF]	R	C	M	(Reserved, 0)			F	
Checksum (* Init)	**0:** Disable **1:** Enable							
Sample Mode	**0:** Normal Mode (14 bits) **1:** Fast Mode (12 bits)							
Analog Format	**0:** Engineering **1:** Percent (% of FSR)				**2:** 2's Complement **3:** Ohms			

Note: When using tM-TH8 or tM-DA1P1R1, set the reserved bit 5 as 0.

For using the Digital module: Set the Counter Update Direction and Checksum.

Bit	7	6	5	4	3	2	1	0
[FF]	CU	CS	(Reserved, 0)			CD		
Counter Update	**0:** Update the counter when there is a falling-edge signal **1:** Update the counter when there is a rising-edge signal							
Checksum (* Init)	**0:** Disable **1:** Enable							
Code	tM-P4C4: 1 (read-only) By default, the code is "0" and can be changed for other modules.							

Response:

Valid Command: **!AA** [CHKSUM] (CR)

Invalid Command: **?AA** [CHKSUM] (CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command If the **Baud Rate** or **Checksum** setting is not configured in Init mode, the module will return an invalid command.

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: %0102400600	Set the address of the digital module '01' to '02', the parity to 'N,8,1-no parity', the baud rate to '9600', the checksum to 'Disable', the sample mode to 'Normal' and the analog format to 'Engineering'
Response: !02	Module 02 respond the command is valid

Command: %0101000A00	Set the baud rate of module 01 as 115200 bps
Response: ?01	The command is invalid because it is not set in Init mode

Command: %0101000A00	In Init mode, set the baud rate of module 01 as 115200 bps
Response: !01	Module 01 respond the command is valid

Related Commands: $AA2, $AA7CiRrr, $AA9NTS

Related Sections: 3.2 Configuration Page - Configuration

$AA8Ci Read the type code for an AI channel

Note: The command only available for tM-AD2, tM-TH8, and tM-AD4P2C2 modules.

Syntax:

$AA8Ci[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

8 Command to read the type code of an analog input channel

Ci i, the channel to be read (0 to 7)

Response:

Valid Command: **!AACiRrr**[CHKSUM] (CR)

Invalid Command: **?AA** [CHKSUM] (CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Ci i, the channel to be read (0 to 7)

Rrr rr, the type code of the specified channel

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $018C0	Read the type code for channel 0 of module 01, e.g., tM-AD2
Response: !01C0R08	The type code of channel 0 is '08', i.e., 0 to 10 V

Command: $028C3	Read the type code for channel 3 of module 02, e.g., tM-AD4P2C2
Response: !02C3R0D	The type code of channel 3 is '0D', i.e., -20 to +20 mA

Command: 038C5	Read the type code for channel 5 of module 03, e.g.,tM-TH8
Response: !03C5R6C	The type code of channel 5 is '6C', i.e., YSI H Mix 30000 @ 25°C -10°C to 200°C

Related Commands:

$AA7CiRrr, %AANNTTCCFF

Related Sections:

3.3.1 Configuration Page (tM-AD2), 3.3.3 Configuration Page (tM-TH8),
3.4.3 Configuration Page (tM-AD4P2C2)

$AA7CiRrr Set the type code for an AI channel

Note: The command only available for tM-AD2, tM-TH8, and tM-AD4P2C2 modules. All channels of the modules can be configured individually.

Syntax:

$AA7CiRrr[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

7 Command to set the type code of an analog input

Ci i, the channel to be set (0 to 7)

Rrr rr, the type code of the channel

Response:

Valid Command: **!AA** [CHKSUM] (CR)

Invalid Command: **?AA** [CHKSUM] (CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $017C0R08	Set the type code of channel 0 of module 01 (e.g., tM-AD2) as '08' (i.e., 0 to 10 V)
Response 01	The command is valid

Command: $027C3R0D	Set the type code of channel 3 of module 02 (e.g., tM-AD4P2C2) as '0D' (i.e., -20 to +20 mA)
Response: !02	The command is valid

Command: $037C5R6C	Set the type code of channel 5 of module 03 (e.g., tM-TH8) as '6C' (i.e., YSI H Mix 30000 @ 25°C -10°C to 200°C)
Response: !03	The command is valid

Command: $037C1R30	Set the type code of channel 1 of module 03 (e.g., tM-TH8) as '30'
Response: ?03	The command is invalid

Related Commands: $AA8Ci, %AANNTTCCFF

Related Sections: 3.3.1 Configuration Page (tM-AD2), 3.3.3 Configuration Page (tM-TH8), 3.4.3 Configuration Page (tM-AD4P2C2)

$AA9N Read the type code and slew rate for an AO channel

Note: The command is available for tM-DA1P1R1.

Syntax:

$AA9N[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

9 Command to read the settings of an analog output channel

N The channel to be read. It is '0' for the tM-DA1P1R1 module.

Response:

Valid Command: **!AATS**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

T The type code of an analog output channel
Refer to Appendix B.7 Data Ranges for tM-DA1P1R1

S The slew rate of an analog output channel
Refer to Appendix A.4 Slew Rate Control

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $0190	Read the settings for analog output channel 0 of module 01
Response: !0110	The type code is '1' (i.e., 4 to 20 mA); The slew rate is '0' (i.e., Immediate)

Related Commands:

$AA9NTS

Related Sections:

3.4.2 Configuration Page - AO,
Appendix A.4 Slew Rate Control,
Appendix B.7 Data Ranges for tM-DA1P1R1

$AA9NTS Set the type code and slew rate for an AO channel

Note: The command is available for tM-DA1P1R1.

Syntax:

$AA9NTS[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

9 Command to configure an analog output channel

N The channel to be set. It is '0' for tM-DA1P1R1

T The type code of an analog output channel
Refer to Appendix B.7 Data Ranges for tM-DA1P1R1

S The slew rate of an analog output channel
Refer to Appendix A.4 Slew Rate Control

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $019021	Set the type code of analog output channel 0 as '2' (i.e., 0 to 10 V) and set the slew rate as '1' (i.e., 0.625 V/sec)
Response: !01	The command is valid

Related Commands:

$AA9N

Related Sections:

3.4.2 Configuration Page - AO,
Appendix A.4 Slew Rate Control,
Appendix B.7 Data Ranges for tM-DA1P1R1

$AA5 Read the reset status of a module

Syntax:

$AA5[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

5 Command to read the reset status of a module

Response:

Valid Command: **!AAS** [CHKSUM] (CR)

Invalid Command: **?AA** [CHKSUM] (CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

S Used to check if the module is ever rebooted

The first time to execute the command after the module is powered on, it will return a value '1'. Afterward, executing the command will get a value '0'. If you get a value "1" again which means the module has been rebooted.

1: The module has been rebooted (or just turned on).

0: The module is never rebooted after it is powered on.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $015	Read the reset status of module 01
Response: !011	Module 01 has been rebooted (or just turned on)

Command: $015	Read the reset status of module 01
Response: !010	Module 01 is never rebooted after it is powered on

$AAF Read the firmware version of a module

Syntax:

$AAF[CHKSUM](CR)

$ Delimiter Character
AA The address of the module (Hex., 00 to FF)
F Command to read the firmware version

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command
? Delimiter character for an invalid command
AA The address of the module (Hex., 00 to FF)
(Data) A string to represent the firmware version

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $01F	Read the firmware version of module 01
Response: !01A2.0	The version is “A2.0”

$AAM Read the name of a module

Syntax:

$AAM[CHKSUM](CR)

$ Delimiter Character
AA The address of the module (Hex., 00 to FF)
M Command to read the module name

Response:

Valid Command: **!AA(Name)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command
? Delimiter character for an invalid command
AA The address of the module (Hex., 00 to FF)
(Name) A string to represent the module name

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $03M	Read the name of module 03
Response: !03tTH8	The module name is "tM-TH8"

Command: $02M	Read the name of module 02
Response: !02tAD4P2C2	The module name is "tM-AD4P2C2"

Note: It is recommended NOT to change the name of tM series module.

$AAP Read the communication protocol of a module

Syntax:

$AAP[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

P Command to read the communication protocol

Response:

Valid Command: **!AASC**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

S The protocol supported by the module

0: Only DCON protocol is supported

1: Both the DCON and Modbus RTU protocols are supported

3: All of the DCON and Modbus RTU/ASCII protocols are supported

C The Protocol setting of the module that is saved in EEPROM

0: the protocol set in EEPROM is DCON

1: the protocol set in EEPROM is Modbus RTU

3: the protocol set in the EEPROM is Modbus ASCII

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $01P	Read the Protocol setting of module 01
Response: !0130	Module 01 supports both the DCON and Modbus RTU/ASCII protocol and the Protocol setting is DCON

Command: $00P	Read the Protocol setting of the module in Init mode, that is the address is 00 and the protocol is DCON by default
Response: !0233	The address of the module is 02, it supports both the DCON and Modbus RTU/ASCII protocols and the protocol set in the EEPROM is Modbus ASCII Note that the module settings are saved in EEPROM, , the user needs to set the switch to Run and reboot to apply the settings.

Related Commands: $AAPN

Related Sections: 3.2 Configuration Page - Configuration

$AAPN Set the communication protocol of a module

Note: The command MUST be used in Init mode.

Syntax:

$AAPN[CHKSUM](CR)

$	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
P	Command to set the communication protocol
N	0: DCON protocol
	1: Modbus RTU protocol
	3: Modbus ASCII protocol

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Note: After setting Protocol in Init mode (i.e., AA = **00**, Protocol = DCON), the settings will be saved in EEPROM, and it will become effective after rebooting in Run mode.

Examples:

Command: $00P1	Set the communication protocol of module to Modbus RTU in Init mode
Response: ?01	The command is invalid because it is not set in Init mode

Command: $00P1	Set the communication protocol of module to Modbus RTU in Init mode
Response: !02	The command is invalid for Module 02

Command: $00P	Read the Protocol setting of the module in Init mode, that is the address is 00 and the protocol is DCON by default
Response: !0231	The address of the module is 02, it supports both the DCON and Modbus RTU/ASCII protocols and the protocol set in the EEPROM is Modbus RTU Note that the module settings are saved in an EEPROM, the user needs to set the switch to Run and reboot to apply the settings.

Related Commands: $AAP

Related Sections: 3.2 Configuration Page - Configuration

$AAI Read the INIT terminal status of a module

Syntax:

$AAI[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

I Command to read the module INIT status

Response:

Valid Command: **!AAS**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

S The Init status of the module

0: The current position of the DIP switch is “Init”

1: The current position of the DIP switch is “Run”

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $01I	Read the Init status of module 01
Response: !010	The DIP switch on the right side of the module is in the “Init” position

~AARD Read the response delay time value of a module

Syntax:

~AARD[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

RD Command to read the response delay time value

Response:

Valid Command: **!AAVV**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

VV Two hexadecimal digits to represent the response delay time value in milli-second, for example, 01 denotes 1ms and 1E denotes 30ms. The max allowable value is 30 (1Eh)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~01RD	Read the response delay time value of module 01
Response: !0102	The module returns 02, which means the response delay time value is 2ms

Related Commands:

~AARDVV

Related Sections:

3.2.2 to 3.2.4 Configure AI Items

~AARDVV Set the response delay time value of a module

Syntax:

~AARDVV[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

RD Command to set the response delay time value

VV Using two hexadecimal digits to represent the response delay time value in milli-second, for example, 01 denotes 1ms and 1E denotes 30ms. The max allowable value is 30 (1Eh)

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~01RD06	Set the response delay time value of module 01 to 6 ms
Response: !01	A valid response returned by the module

Command: ~01RD	Read the response delay time value of module 01
Response: !0106	The module returns 06, which means the response delay time value is 6ms

Related Commands:

~AARD

Related Sections:

3.2.2 to 3.2.4 Configure AI Items

~AAI The Soft INIT command

Description:

Normally, the Parity, Baud Rate, and Checksum setting of the module must be set in INIT mode. Alternatively, users can also configure them by using the software INIT command.

Note: The command is available for tM-TH8.
Make sure that the timeout has been set (~AATnn) before using this command.

Syntax:

~AAI[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

I Command to set Soft Init

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~01T10	Set the Soft INIT timeout of module 01 to 16 seconds
Response: !01	A valid response returned by the module

Command: ~01I	Set the status of module 01 to Soft INIT
Response: !01	A valid response returned by the module

Command: %0101000700	Set Module 01 as follows: Parity = N,8,1-no parity, Baud Rate = 19200, Checksum = 0, and Analog Format = Engineering
Response: !01	A valid response returned by the module

Related Commands: ~AATnn, %AANNTTCCFF

Related Sections: 2.5 Operating Switch and Parameters

~AATnn Set the Soft Init timeout

Note: The command is available for tM-TH8.

Syntax:

~AATnn[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

T Command to set the soft INIT timeout

nn Using two hexadecimal digits to represent the timeout value in seconds. The maximum timeout value is 60 seconds.

Note: After sending the Soft INIT command (i.e., ~ AAI), the %AANNTTCCFF command must be set before the Soft INIT timeout (i.e., the value =0), or the change of Parity, Baud Rate, and Checksum settings will be invalid. After rebooting, the Soft INIT timeout is 0.

Response:

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~01T10	Set the Soft INIT timeout of module 01 to 16 seconds
Response: !01	A valid response returned by the module

Command: ~01I	Set the status of module 01 to Soft INIT
Response: !01	A valid response returned by the module

Command: %0101000700	Set Module 01 as follows: Parity = N,8,1-no parity, Baud Rate = 19200, Checksum = 0, and Analog Format = Engineering Note: After completing the settings, it is recommended to set the Init timeout to 0 to disable the Soft INIT
Response: !01	A valid response returned by the module

Related Commands: ~AAI, %AANNTTCCFF

Related Sections: 2.5 Operating Switch and Parameters

$AA6 Read the status of AI/DI/DO channels

Description:

- The command is used to read the enabled/disabled status of each channel. It is available for analog input and tM-AD4P2C2 modules.
- The command is used to read the ON/OFF status of each channel. It is available for digital input/ output and tM-DA1P1R1 modules.

Syntax:

$AA6[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

6 Command to read the channel status

Response:

Valid Command: [For AI modules] **!AAVV**[CHKSUM](CR)

[For DIO module] **!(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

VV [For AI modules] Using two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1, which means the channel is enabled, and 0 denotes that the channel is disabled.

(Data) [For DIO modules] Using four-digit hexadecimal value followed by 00.

DIO Data Format Table:

1) The $AA4, $AA6, and $AALS command: **[the First Data] [The Second Data]00**
2) The @AA command: **[the First Data] [The Second Data]**

Module Name	DIO	The First Data	The Second Data
tM-DA1P1R1	1 DO ; 1 DI	DO0 **00 to 01**	DI0 **00 to 01**
tM-P3R3	3 DO ; 3 DI	DO0 to DO2 **00 to 07**	DI0 to DI2 **00 to 07**
tM-P3POR3			
tM-P4C4	4 DO ; 4 DI	DO0 to DO3 **00 to 0F**	DI0 to DI3 **00 to 0F**
tM-P4A4			
tM-R5	5 DO	DO0 to DO4 **00 to 1F**	**00**
tM-C8	8 DO	DO0 to DO7 **00 to FF**	
tM-P8	8 DI	DI0 to DI7 **00 to FF**	

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

[For AI and tM-AD4P2C2 modules]

Command: $016	Read the enabled/disabled status of the AI module 01 (e.g., tM-AD8)
Response: !013F	AI0 to AI5 are enabled; AI6 and AI7 are disabled

AI7	AI6	AI5	AI4	AI3	AI2	AI1	AI0
0	0	1	1	1	1	1	1
3				F			

Command: $026	Read the enabled/disabled status of the AI module 02 (e.g., tM-AD4P2C2)
Response: !020A	AI1 and AI3 are enabled；AI0 and AI2 are disabled

AI3	AI2	AI1	AI0
1	0	1	0
0A			

[For DIO and tM-DA1P1R1 modules]

Command: $016	Read the ON/OFF status of the DIO module 01 (e.g., tM-DA1P1R1)
Response: !010100	DO0 and DI0 are ON.

DO0	DI0
1	1

Command: $026	Read the ON/OFF status of the DIO module 02 (e.g., tM-C8)
Response: !D50000	DO0, DO2, DO4, and DO6 are ON；DO1, DO3, DO5, and DO7 are OFF

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
1	1	0	1	0	1	0	1
D				5			

Related Commands: @AA, $AA5VV

Related Sections:

3.3 Configuration Page – AI

3.4.3 Configuration Page – AI/DO Alarm

3.4.4 Configuration Page – DO

3.4.6 Configuration Page - DI

$AA5VV Set the status of AI channels

Description:

Specify the channel (s) to be enabled

Note: The command is available for AI and tM-AD4P2C2 modules. It is recommended that only the channels to be used are enabled.

Syntax:

$AA5VV[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

5 Command to set the channels to enabled/disabled

VV Using two-digit hexadecimal value, where bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1, which means the channel is enabled, and 0 denotes that the channel is disabled.

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Command: $0151A	Enable AI1, AI3 and AI4 channels of module 01 (e.g., tM-AD5) AI4: 1, AI3: 1, AI2: 0, AI1: 1, AI0: 0
Response: !01	The command is valid

Command: $016	Read the enabled/disabled status of module 01
Response: !011A	AI1, AI3, and AI4 are enabled

Related Commands: $AA6

Related Sections: 3.3 Configuration Page - AI, 3.4.3 Configuration Page – AI/DO Alarm

#** Notify all modules to synchronously access the DIO status

Description:

All DIO modules with the same communication settings on one RS-485 bus will receive this command and record the current DIO status.

Note: 1. The command is available for DIO and Multi-function (tM-DA1P1R1, tM-AD4P2C2) modules.
2. Test it with the $AA4 command by using VxComm Utility.

Syntax:

#**[CHKSUM](CR)

#	Delimiter Character
****	Synchronized sampling command

Response:

No response.

Examples:

Command: #**	Notify all modules to synchronously read and store the DIO data
Response:	No response.

Command: $014	Read the synchronized data of module 01 (e.g., tM-P8).
Response: !1C30000	

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
1	1	0	0	0	0	1	1

Command: $024	Read the synchronized data of module 01 (e.g., tM-AD4P2C2)
Response: !1020000	

DO1	DO0	DI1	DI0
1	0	0	0

Command: $024	Read the synchronized data of module 01 (e.g., tM-AD4P2C2)
Response: !0020000	0, means that it is not the first time to read the synchronized data. Users can execute the #** command to get the latest synchronized data.

Related Commands: $AA4

Related Sections: $AA6 Command – DIO Data Format Table

$AA4 Read the synchronous sampling DIO status

Note: 1. The command supports DIO and Multi-function (i.e., tM-DA1P1R1, tM-AD4P2C2) modules.
2. Test it with the #** command by using VxComm Utility.

Syntax:

$AA4[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

4 Command to read the synchronized data

Response:

Valid Command: **!S(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

S The status of the synchronized data

1: The first time to read data after sampling

0: Not the first time to read data, that is not the latest data.

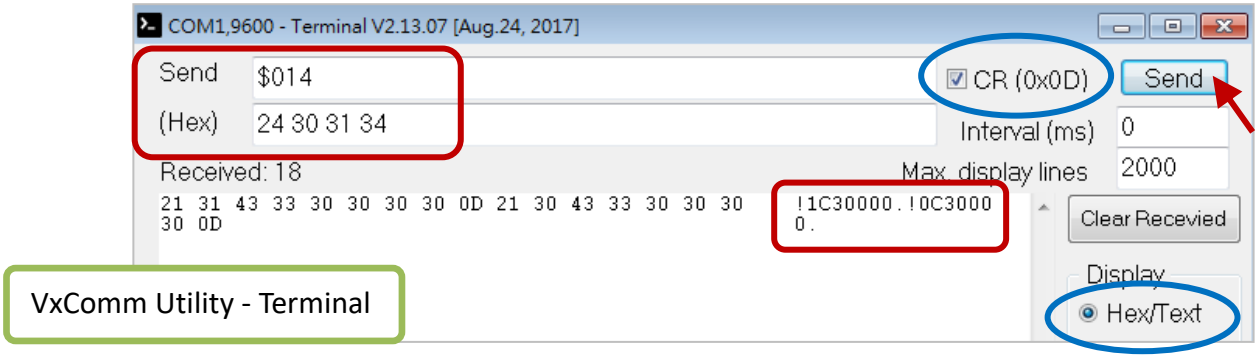
Also, execute the #** command to synchronously record the current data.

Examples:

Command: #**	Notify all modules to synchronously read and store the DIO data
Response:	No response.

Command: $014	Read the synchronized data of module 01 (e.g., tM-P8)
Response: !1C30000	

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
1	1	0	0	0	0	1	1



Related Commands: #**

Related Sections: $AA6 Command – DIO Data Format Table

@AA Read the status of DI/DO channels

Description:

Read the ON/OFF status of each channel.

Note: The command supports DIO and Multi-function (i.e., tM-DA1P1R1, tM-AD4P2C2) modules.

Syntax:

@AA[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

Response:

Valid Command: **>(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

> Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) Using four-digit hexadecimal to represent the DIO status

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Command: @01	Read the DIO status of module 01 (e.g., tM-DA1P1R1)
Response: >0101	DO0 and DI0 are ON DO0 DI0 1 1

Command: @02	Read the DIO status of module 02 (e.g., tM-AD4P2C2)
Response: >0203	DO1, DI0, and DI1 are ON；DO0 is OFF DO1 DO0 DI1 DI0 1 0 1 1

Command: @03	Read the DIO status of module 03 (e.g., tM-P3R3)
Response: >0207	DO1, DI0, and DI2 are ON；DO0 and DO2 are OFF DO2 DO1 DO0 DI2 DI1 DI0 0 1 0 1 1 1 02 07

Related Commands: $AA6

Related Sections: 3.4.4 Configuration Page - DO, 3.4.6 Configuration Page - DI

@AADI Read the status of DI/DO channels (for Multi-function module)

Description:

Read the DI and DO status of a module

Note: The command is available for tM-DA1P1R1 and tM-AD4P2C2 modules.

Syntax:

@AADI[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
DI	Command to read the digital input and digital output status

Response:

Valid Command: **!AA0OOII**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)
OO	Using two-digit hexadecimal value to represent the DO status
II	Using two-digit hexadecimal value to represent the DI status Bit 0 corresponds to DI0 (or DO0), Bit 1 corresponds to DI1 (or DO1), etc. When the bit is 1, indicates the DI/DO status is ON; When the bit is 0, indicates the DI/DO status is OFF.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Command: @01DI	Read the DIO status of module 01 (e.g., tM-DA1P1R1)
Response: !0100101	DO0 and DI0 are ON DO0: 1, DI0: 1

Command: @02DI	Read the DIO status of module 02 (e.g., tM-AD4P2C2)
Response: !0200203	DO1, DI0, and DI1 are ON; DO0 is OFF DO1: 1, DO0: 0, DI1: 1, DI0: 1

Related Commands: @AADODD

@AA(Data) Set the status of DO channels

Note: The command is available for DO and Multi-function (i.e., tM-DA1P1R1, tM-AD4P2C2) modules.

Syntax:

@AA(Data)[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

(Data) Data to be written to the DO channel

For DIO modules (e.g., tM-P4C4), using one-digit hexadecimal value to represent the DO status

For DO module (e.g., tM-C8), using two-digit hexadecimal value to represent the DO status

Bit 0 corresponds to DI0 (or DO0), Bit 1 corresponds to DI1 (or DO1), etc.

When the bit is 1, indicates the DO status is ON;

When the bit is 0, indicates the DO status is OFF.

Response:

Valid Command: **>**[CHKSUM](CR)

Invalid Command: **?**[CHKSUM](CR)

Ignored Command: **!**[CHKSUM](CR)

> Delimiter character for a valid command

? Delimiter character for an invalid command

! Delimiter character for an ignored command

When a host watchdog timeout occurs, the module will set all outputs to a Safe Value.

In this case, the data received from DO channels will be ignored.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Command: @017	For module 01 (e.g., tM-P4C4), set DO0, DO1, and DO2 to ON, and DO3 to OFF. DO3: 0, DO2: 1, DO1: 1, DO0: 1
Response: >	The command is valid

Command: @0307	For module 03 (e.g., tM-C8), set DO0, DO1, and DO2 to ON, and other channels to OFF. DO7: 0, DO6: 0, DO5: 0, DO4: 0, DO3: 0, DO2: 1, DO1: 1, DO0: 1
Response: >	The command is valid

Related Commands: @AA

Related Sections: 3.4.4 Configuration Page - DO

@AADODD Set the status of DO channels (for Multi-function module)

Description:

Set the DO status of a module

Note: The command is available for tM-DA1P1R1 and tM-AD4P2C2 modules.

Syntax:

@AADODD[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

DO Command to set the DO status

DD Using two-digit hexadecimal value,
bit 0 corresponds to DO0, and bit 1 corresponds to DO1, etc.
When the bit is 1, indicates the DO status is ON;
When the bit is 0, indicates the DO status is OFF.

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Command: @01DO01	For module 02 (e.g., tM-DA1P1R1), set DO0 to ON. DO0 1
Response: !01	The command is valid

Command: @02DO03	For module 02 (e.g., tM-AD4P2C2), set DO0 and DO1 to ON. DO1 DO0 1 1
Response: !02	The command is valid

Related Commands: @AADI

Related Sections: 3.4.3 Configuration Page – AI/DO Alarm

#AA00(Data) Set the status of DO channels (DO0 to DO7)

Note: The command is available for DO modules, and it is the same as the #AA0A(Data) command.

Syntax:

#AA00(Data)[CHKSUM](CR)

Delimiter Character

AA The address of the module (Hex., 00 to FF)

00 For more than 8-channel, only the lower channel (i.e., DO0 to DO7) can be set.

(Data) Using the two-digit hexadecimal value, bit 0 corresponds to DO0 and bit 1 corresponds to DO1, etc. When the bit is 1, indicates the DO status is ON; When the bit is 0, indicates the DO status is OFF.

Response:

Valid Command: **>**[CHKSUM](CR)

Invalid Command: **?**[CHKSUM](CR)

Ignored Command: **!**[CHKSUM](CR)

> Delimiter character for a valid command

? Delimiter character for an invalid command

! Delimiter character for an ignored command

When a host watchdog timeout occurs, the module will set all outputs to a Safe Value. In this case, the data received from DO channels will be ignored.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Command: **#010005**	For module 01 (e.g., tM-P4A4), set DO0 and DO2 to ON, and DO1, DO3 to OFF. DO3: 0, DO2: 1, DO1: 0, DO0: 1
Response: >	The command is valid

Command: **#030033**	For module 03 (e.g., tM-C8), set DO0, DO1, DO4, and DO5 to ON, and DO2, DO3, DO6, DO7 to OFF. DO7: 0, DO6: 0, DO5: 1, DO4: 1, DO3: 0, DO2: 0, DO1: 1, DO0: 1
Response: >	The command is valid

Related Commands: #AA0A(Data), #AA0B(data), #AAAcDD, #AABcDD, $AA6, @AA

#AA0A(Data) Set the status of DO channels (DO0 to DO7)

Note: The command is available for DO modules, and it is the same as the #AA00(Data) command.

Syntax:

#AA0A(Data)[CHKSUM](CR)

#	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
0A	For more than 8-channel, only the lower channel (i.e., DO0 to DO7) can be set.
(Data)	Using two-digit hexadecimal value, bit 0 corresponds to DO0, and bit 1 corresponds to DO1, etc. When the bit is 1, indicates the DO status is ON; When the bit is 0, indicates the DO status is OFF.

Response:

Valid Command: **>**[CHKSUM](CR)

Invalid Command: **?**[CHKSUM](CR)

Ignored Command: **!**[CHKSUM](CR)

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
!	Delimiter character for an ignored command When a host watchdog timeout occurs, the module will set all outputs to a Safe Value. In this case, the data received from DO channels will be ignored.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

<table>
<tr><td>Command:
#010A06</td><td>For module 01 (e.g., tM-P3POR3), set DO1, DO2 to ON, and DO0 to OFF
<table>
<tr><td>DO2</td><td>DO1</td><td>DO0</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>Response: ></td><td>The command is valid</td></tr>
</table>

<table>
<tr><td>Command:
#030A33</td><td>For module 03 (e.g., tM-C8), set DO0, DO1, DO4, and DO5 to ON, and DO2, DO3, DO6, DO7 to OFF.
<table>
<tr><td>DO7</td><td>DO6</td><td>DO5</td><td>DO4</td><td>DO3</td><td>DO2</td><td>DO1</td><td>DO0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Response: ></td><td>The command is valid</td></tr>
</table>

Related Commands: #AA00(Data), #AA0B(data), #AA1cDD, #AAAcDD, #AABcDD, $AA6, @AA

#AA0B(Data) Set the status of DO channels (DO8 to DO15)

Note: The command is available for DO modules.
tM modules support up to 8 DO channels (0 to 7).

Syntax:

#AA0B(Data)[CHKSUM](CR)

#	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
0B	For more than 8-channel, only the higher channel (i.e., DO8 to DO15) can be set.
(Data)	Using two-digit hexadecimal value, bit 0 corresponds to DO8, and bit 1 corresponds to DO9, etc. When the bit is 1, indicates the DO status is ON; When the bit is 0, indicates the DO status is OFF.

Response:

Valid Command: **>**[CHKSUM](CR)
Invalid Command: **?**[CHKSUM](CR)
Ignored Command: **!**[CHKSUM](CR)

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
!	Delimiter character for an ignored command When a host watchdog timeout occurs, the module will set all outputs to a Safe Value. In this case, the data received from DO channels will be ignored

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

<table>
<tr><td>Command:
#020B33</td><td>For module 02, set DO8, DO9, DO12, and DO13 to ON, and
DO10, DO11, DO14, DO15 to OFF
<table>
<tr><th>DO15</th><th>DO14</th><th>DO13</th><th>DO12</th><th>DO11</th><th>DO10</th><th>DO9</th><th>DO8</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>Response: ></td><td>The command is valid</td></tr>
</table>

Related Commands:

#AA00(Data), #AA0A(Data), #AA1cDD, #AAAcDD, #AABcDD, $AA6, @AA

#AA1cDD Set the status of a DO channel (DO0 to DO7)

Note: The command is available for DO modules, and it is the same as the #AAAcDD command.

Syntax:

#AA1cDD[CHKSUM](CR)

#	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
1	Command to set a single digital output channel. For more than 8-channel, only the lower channel (i.e., DO0 to DO7) can be set.
c	Specify the digital output channel to be set (0 to 7)
DD	00, set the DO channel to OFF 01, set the DO channel to ON

Response:

Valid Command: **>**[CHKSUM](CR)

Invalid Command: **?**[CHKSUM](CR)

Ignored Command: **!**[CHKSUM](CR)

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
!	Delimiter character for an ignored command When a host watchdog timeout occurs, the module will set all outputs to a Safe Value. In this case, the data received from DO channels will be ignored

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Command: **#011201**	For module 01, set DO2 to ON. DO7: 0, DO6: 0, DO5: 0, DO4: 0, DO3: 0, DO2: 1, DO1: 0, DO0: 0
Response: >	The command is valid

Command: **#011401**	For module 01, set DO4 to ON. DO7: 0, DO6: 0, DO5: 0, DO4: 1, DO3: 0, DO2: 1, DO1: 0, DO0: 0
Response: >	The command is valid

Related Commands: #AA00(Data), #AA0A(Data), #AA0B(data), #AAAcDD, #AABcDD, $AA6, @AA

#AAAcDD Set the status of a DO channel (DO0 to DO7)

Note: The command is available for DO modules, and it is the same as the #AA1cDD command.

Syntax:

#AAAcDD[CHKSUM](CR)

Delimiter Character

AA The address of the module (Hex., 00 to FF)

A Command to set a single digital output channel

For more than 8-channel, only the lower channel (i.e., DO0 to DO7) can be set

c Specify the digital output channel to be set (0 to 7)

DD 00, set the DO channel to OFF

01, set the DO channel to ON

Response:

Valid Command: **>**[CHKSUM](CR)

Invalid Command: **?**[CHKSUM](CR)

Ignored Command: **!**[CHKSUM](CR)

> Delimiter character for a valid command

? Delimiter character for an invalid command

! Delimiter character for an ignored command

When a host watchdog timeout occurs, the module will set all outputs to a Safe Value.

In this case, the data received from DO channels will be ignored

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Command: **#01A201**	For module 01, set DO2 to ON

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
0	0	0	0	0	1	0	0

Response >	The command is valid

Command: **#01A401**	For module 01, set DO4 to ON

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
0	0	0	1	0	1	0	0

Response: >	The command is valid

Related Commands: #AA00(Data), #AA0A(Data), #AA0B(data), #AA1cDD, #AABcDD, $AA6, @AA

#AABcDD Set the status of a DO channel (DO8 to DO15)

Note: The command is available for DO modules.
tM modules support up to 8 DO channels (0 to 7).

Syntax:

#AA0B(Data)[CHKSUM](CR)

#	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
B	Command to set a single digital output channel For more than 8-channel, only the higher channel (i.e., DO8 to DO15) can be set
c	Specify the digital output channel to be set (8 to 15)
DD	00, set the DO channel to OFF 01, set the DO channel to ON

Response:

Valid Command: **>**[CHKSUM](CR)
Invalid Command: **?**[CHKSUM](CR)
Ignored Command: **!**[CHKSUM](CR)

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
!	Delimiter character for an ignored command When a host watchdog timeout occurs, the module will set all outputs to a Safe Value. In this case, the data received from DO channels will be ignored

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

<table>
<tr><td>Command:
#020B33</td><td>For module 02, set DO8, DO9, DO12, and DO13 to ON, and
DO10, DO11, DO14, DO15 to OFF
<table>
<tr><th>DO15</th><th>DO14</th><th>DO13</th><th>DO12</th><th>DO11</th><th>DO10</th><th>DO9</th><th>DO8</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>Response: ></td><td>The command is valid</td></tr>
</table>

Related Commands:

#AA00(Data), #AA0A(Data), #AA0B(data), #AA1cDD, #AAAcDD, $AA6, @AA

#AA Read data from all AI channels

Note: The command is available for AI and tM-AD4P2C2 modules.

Syntax:

#AA[CHKSUM](CR)

#	Delimiter Character
AA	The address of the module (Hex., 00 to FF)

Response:

Valid Command: **>(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
(Data)	Data from every analog input channels
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: #01	Read data from all AI channels of module 01 (e.g., tM-AD2)
Response: >+00.001+00.007 Data in engineering format: Ch00 = +00.001, Ch01 = +00.007.	

Command: #02	Read data from all AI channels of module 02 (e.g., tM-AD4P2C2)
Response: >+07.389+07.389+00.002+00.002 Data in engineering format: Ch00 = +07.389, Ch01 = +07.389, Ch02 = +00.002, Ch03 = +00.002.	

Command: #03	Read data from all AI channels of module 03 (e.g., tM-TH8)
Response: >-9999.9-9999.9-9999.9-9999.9-9999.9-9999.9-9999.9-9999.9 Data in engineering format: Ch00 to Ch07 are -9999.9, which is under the range	

Related Commands: #AAN, #AAA

Related Sections:

3.3 Configuration Page - AI, 3.4.3 Configuration Page – AI/DO Alarm

#AAN Read data from an AI channel

Note: The command is available for AI and tM-AD4P2C2 modules.

Syntax:

#AAN[CHKSUM](CR)

#	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
N	The channel to be read, zero-based

Response:

Valid Command: **>(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
(Data)	Analog input data of the specified channel
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	**#010**	Read data from channel 0 of module 01 (e.g., tM-AD5)
Response:	**>+06.251**	It indicates Ch00 = +06.250

Command:	**#013**	Read data from channel 3 of module 01 (e.g., tM-AD2)
Response:	**?02**	The command is invalid due to no channel 3

Command:	**#023**	Read data from channel 3 of module 02 (e.g., tM-AD4P2C2)
Response:	**>+00.002**	It indicates Ch03 = +00.002

Related Commands: #AA, #AAA

Related Sections:

3.3 Configuration Page - AI, 3.4.3 Configuration Page – AI/DO Alarm

$AAA Read hexadecimal data from all AI channels

Note: The command is available for AI and tM-AD4P2C2 modules.

Syntax:

$AAA[CHKSUM](CR)

$	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
A	Command to read every analog input

Response:

Valid Command: **>(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

>	Delimiter character for a valid command
?	Delimiter character for an invalid command
(Data)	Data from every analog input channel in hex. format
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	**$01A**	Read data from all channels of module 01 (e.g., tM-AD2)
Response:	**>7FFA7FF7**	Ch00 = +09.998, Ch01 = +09.998 (V)

Command:	**$02A**	Read data from all channels of module 02 (e.g., tM-AD4P2C2)
Response:	**>5E945E9400090008**	Ch00 = +07.389, Ch01 = +07.389, Ch02 = +00.002, Ch03 = +00.002.

Related Commands: #AA, #AAN

Related Sections:

3.3 Configuration Page - AI, 3.4.3 Configuration Page – AI/DO Alarm

$AA8N Read data from an AO channel

Note: The command is available for tM-DA1P1R1.

Syntax:

$AA8N[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

8 Command to read the AO value

N The channel to be read, 0 for tM-DA1P1R1

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) The analog output value

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: #010+04.000	Set the AO value of channel 0 of module 01 to 4.0
Response: >	The command is valid

Command: ~0180	Read the AO value of channel 0 of module 01
Response: !01+04.000	Module 01 returns 4.0 (Refer to B.7 Data Ranges for tM-DA1P1R1)

Related Commands:

#AAN(Data), $AA6N

Related Sections:

3.4.2 Configuration Page - AO, Appendix B.7 Data Ranges for tM-DA1P1R1

$AA6N Read the last written data from an AO channel

Note: The command is available for tM-DA1P1R1.

Syntax:

$AA6N[CHKSUM](CR)

~	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
6	Command to read the last written analog output value
N	The channel to be read, 0 for tM-DA1P1R1

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)
(Data)	The last written analog output value.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: #010+10.000	Set the AO value of channel 0 of module 01 to 10.0
Response: >	The command is valid

Command: $0160	Read the last written AO value of channel 1 of module 01
Response: !01+10.000	Module 01 returns 10.0

Related Commands:

#AAN(Data), $AA8N

Related Sections:

3.4.2 Configuration Page - AO, Appendix B.7 Data Ranges for tM-DA1P1R1

#AAN(Data) Set a value of an AO channel

Note: The command is available for tM-DA1P1R1.

Syntax:

#AAN(Data)[CHKSUM](CR)

#	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
N	The analog output channel to be written, 0 for tM-DA1P1R1
(Data)	Data to be written to the analog output channel, refer to Appendix B.7

Response:

Valid Command: **>**[CHKSUM](CR)

Out of Range: **?**[CHKSUM](CR)

Ignored Command: **!**[CHKSUM](CR)

>	Delimiter character for a valid command
?	Delimiter character indicates that the data is out of range. If it is over the range, then the output will be set to the maximum value of the range. If it is under range, then the output will be set to the minimum value of the range.
!	Delimiter character for an ignored command When a host watchdog timeout occurs, the module will set all outputs to a Safe Value.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $0190	Read the AO configuration of channel 0 of module 01
Response: !0120	It returns: Type Code is 2 and Slew Rate is 0, which means output type is 0 to 10 V and output changing immediately.

Command: #010+05.000	Set the AO value of channel 0 of module 01 to 5.0
Response: >	The command is valid

Command: #010+15.000	Set the AO value of channel 0 of module 01 to 15.0.
Response: ?	The value is out of range, and it will be set to 10 V

Related Commands: $AA9N, %AANNTTCCFF

Related Sections: 3.4.2 Configuration Page - AO, Appendix B.7 Data Ranges for tM-DA1P1R1

~** Notify all modules that the Host works normally

Description:

Informs all modules that the host is OK.

Note: After sending the command, the delay time for the next command sending is 2ms.

Syntax:

~**[CHKSUM](CR)

~ Delimiter Character

** Host OK command.

Response:

No response.

Examples:

Command: ~**	Send a “Host OK” command to all modules
Response: No response	

Related Commands:

~AA0, ~AA1, ~AA2, ~AA3EVV

Related Sections:

3.4.5 Configuration Page – Host WDT, A.2 Dual Watchdog

~AA0 Read the host watchdog status of a module

Syntax:

~AA0[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

0 Command to read the host watchdog status

Response:

Valid Command: **!AASS**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

SS Using two hexadecimal digits to indicates the status of the host watchdog.

Bit 7: 0, indicates that the host watchdog is disabled.
1, indicates that the host watchdog is enabled.

Bit 2: 0, indicates that a host watchdog timeout has occurred.
1, indicates that no host watchdog has occurred.

Note: The host watchdog status is stored in EEPROM and can only be reset using the ~AA1 command.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~010	Read the host watchdog status of module 01
Response: !0100	Return 00, which means the host watchdog is disabled and no timeout occurs

Command: ~020	Read the host watchdog status of module 02
Response: !0204	Return 04, which means the host watchdog timeout has occurred

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

Related Commands: ~**, ~AA0, ~AA1, ~AA2, ~AA3EVV

Related Sections: 3.4.5 Configuration Page – Host WDT, A.2 Dual Watchdog

~AA1 Reset the host watchdog status of a module

Syntax:

~AA1[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

1 Command to reset the host watchdog timeout status

Response:

Valid Command: **!AASS**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	~010	Read the host watchdog status of module 01
Response:	!0104	Return 04, which means the host watchdog timeout has occurred

Command:	**~011**	Reset the host watchdog timeout status of module 01
Response:	**!01**	The command is valid

Command:	~010	Read the host watchdog status of module 01
Response:	!0100	Return 00, which means the host watchdog is disabled and no timeout occurs

Related Commands:

~**, ~AA0, ~AA2, ~AA3EVV

Related Sections:

3.4.5 Configuration Page – Host WDT, A.2 Dual Watchdog

~AA2 Read the host watchdog timeout value of a module

Syntax:

~AA2[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

2 Command to read the host watchdog timeout value

Response:

Valid Command: **!AAEVV**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

E 0, means that the host watchdog is disabled
1, means that the host watchdog is enabled

VV Using two hexadecimal digits to represent the timeout value in tenths of a second, for example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~012	Read the host watchdog timeout value of module 01
Response: !011FF	Return 1FF, which means the host watchdog is enabled and the timeout value is 25.5 seconds

Related Commands:

~**, ~AA0, ~AA1, ~AA3EVV

Related Sections:

3.4.5 Configuration Page – Host WDT, A.2 Dual Watchdog

~AA3EVV Set the host watchdog and a timeout value

Description:

To enable or disable the host watchdog and set the host watchdog timeout value of a module.

Syntax:

~AA3EVV[CHKSUM](CR)

~	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
3	Command to set the host watchdog
E	0, to disable the host watchdog. 1, to enable the host watchdog.
VV	Using two hexadecimal digits to represent the timeout value in tenths of a second, for example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	**~013164**	Enable the host watchdog of module 01 and set the timeout value to 10.0 seconds. $64_{(16)} = 100_{(10)}$
Response:	**!01**	The command is valid

Command:	~012	Read the host watchdog timeout value of module 01.
Response:	!01164	Return 164, which means the host watchdog is enabled and the timeout value is 10.0 seconds.

Related Commands:

~**, ~AA0, ~AA1, ~AA2

Related Sections:

3.4.5 Configuration Page – Host WDT, A.2 Dual Watchdog

~AA4V Read the Power-on value or Safe value of the DO module

Note: tM-AD4P2C2 needs to use the ~AA4 command.

Syntax:

~AA4V[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

4 Command to read the Power-on value or the Safe value

V P: Read the Power-on value

S: Read the Safe Value

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) Using two hexadecimal digits followed by 00 to represent the Power-on value or the Safe value

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~014S	Read the Safe value of module 01 (e.g., tM-DA1P1R1)
Response: !010100	The Safe Value is 01 DO0: ON

Command: ~024P	Read the Power-on value of module 02 (e.g., tM-C8)
Response: !02C300	The Power-on Value is C3

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
ON	ON	OFF	OFF	OFF	OFF	ON	ON

Related Commands: ~AA5V

Related Sections: 3.4.4 Configuration Page - DO, A.2 Dual Watchdog

~AA5V Set the DO value as the Power-on value or Safe value

Note: tM-AD4P2C2 needs to use the ~AA5PPSS command.

Syntax:

~AA5V[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

5 Command to set the Power-on value or the Safe value

V P: Set the Power-on value

S: Set the Safe value

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~015S	Set the Safe value of module 01 (e.g., tM-P4A4) to the current DO value
Response: !01	The command is valid

Command: ~014S	Read the Safe value of module 01
Response: !010C00	The Safe Value is 0C DO3: ON, DO2: ON, DO1: OFF, DO0: OFF

Related Commands: ~AA4V

Related Sections: 3.4.3 Configuration Page – AI/DO Alarm, 3.4.4 Configuration Page - DO, A.2 Dual Watchdog

~AA4 Read the Power-on value or Safe value (Multi-function-DO)

Note: The command is available for tM-DA1P1R1 and tM-AD4P2C2.

Syntax:

~AA4[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

4 Command to read the Power-on value and the Safe value

Response:

Valid Command: **!AAPPSS**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

PP Using two hexadecimal digits to represent the Power-on value

SS Using two hexadecimal digits to represent the Safe value

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~014	Read both the Power-on value and Safe value of module 01
Response: !010000	The Power-on value is 00, and the Safe Value is 00

Command: ~024	Read both the Power-on value and Safe value of module 02
Response: !020203	The Power-on value is 02, and the Safe Value is 03

Power-on Value		Safe Value	
DO1	DO0	DO1	DO0
ON	OFF	ON	ON

Related Commands: ~AA5PPSS

Related Sections: 3.4.3 Configuration Page – AI/DO Alarm, 3.4.4 Configuration Page - DO, A.2 Dual Watchdog

~AA5PPSS Set the Power-on value or Safe value (Multi-function-DO)

Note: The command is available for tM-DA1P1R1 and tM-AD4P2C2. **If the Alarm function of tM-AD4P2C2 is enabled, this command will be invalid.**

Syntax:

~AA5PPSS[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

5 Command to set the Power-on value and the Safe value

PP Using two hexadecimal digits to represent the Power-on value

SS Using two hexadecimal digits to represent the Safe value

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~0150000	Set the power-on value to 00 and the safe value to 00 for module 01
Response: !01	The command is valid

Command: ~014	Read both the Power-on value and Safe value of module 01
Response: !010000	The Power-on value is 00, and the Safe Value is 00

Power-on Value		Safe Value	
DO1	DO0	DO1	DO0
OFF	OFF	OFF	OFF

Related Commands: ~AA4

Related Sections: 3.4.3 Configuration Page – AI/DO Alarm, 3.4.4 Configuration Page - DO, A.2 Dual Watchdog

$AA7N Read the Power-on value of the AO module

Description:

Read the Power-on value of the specified analog output channel.

Note: The command is available for tM-DA1P1R1.

Syntax:

$AA7N[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

7 Command to read the Power-on AO value

N The channel to be read, 0 for tM-DA1P1R1

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) The Power-on AO Value

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $0170	Read the Power-on AO value of channel 0 of module 01
Response: !01+03.000	The Power-on value is 3.0

Related Commands: $AA4N

Related Sections: 3.4.2 Configuration Page - AO, A.2 Dual Watchdog

$AA4N Set the AO value as the Power-on value

Description:

Set the Power-on value of the specified channel to an AO value.

Note: The command is available for tM-DA1P1R1.

Syntax:

$AA4N[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

4 Command to set the Power-on value

N The channel to be set, 0 for tM-DA1P1R1

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	#010+05.000	Set the AO value of channel 0 of module 01 to 5.0
Response:	>	The command is valid

Command:	**$0140**	Set the Power-on value of channel 0 of module 01 to the current AO value
Response:	**!01**	The command is valid

Command:	$0170	Read the Power-on value of channel 0 of module 01
Response:	!01+05.000	The Power-on value is 5.0

Related Commands: #AAN(Data), $AA7N

Related Sections: 3.4.2 Configuration Page - AO, A.2 Dual Watchdog

~AA4N Read the Safe Value of the AO module

Description:

Read the Safe value of the specified analog output channel.

Note: The command is available for tM-DA1P1R1.

Syntax:

~AA4N[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

4 Command to read the Safe AO value

N The channel to be read, 0 for tM-DA1P1R1

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) The Safe AO value

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	**~0140**	Read the Safe value of AO channel 0 of module 01
Response:	**!01+02.500**	The Safe value is 2.5

Related Commands: ~AA5N

Related Sections: 3.4.2 Configuration Page - AO, A.2 Dual Watchdog

~AA5N Set the AO value as the Safe Value

Description:

Set the Safe value of the specified channel to an AO value.

Note: The command is available for tM-DA1P1R1.

Syntax:

~AA5N[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

5 Command to set the Safe value

N The channel to be set, 0 for tM-DA1P1R1

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	#010+03.500	Set the AO value of channel 0 of module 01 to 3.5.
Response:	>	The command is valid

Command:	**~0150**	Set the Safe value of channel 0 of module 01 to the current AO value
Response:	**!01**	The command is valid

Command:	~0140	Read the Safe value of channel 0 of module 01
Response:	!01+03.500	The Safe value is 3.5

Related Commands: #AAN(Data), ~AA4N

Related Sections: 3.4.2 Configuration Page - AO, A.2 Dual Watchdog

$AALS Read the latched status of DI channels

Syntax:

$AALS[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

L Command to read the latched status

S 0: Read the low latched status

1: Read the high latched status

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) Using four hexadecimal digits followed by 00 to represent the DI (or DO) latched status

Refer to the $AA6 Command – DIO Data Format Table

For DI/DO module: (DO or DI Latch) (00) 00

For DIO module: (DO Latch) (DI Latch) 00

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $01L0	Read the low latched status of module 01 (e.g., tM-P8)
Response: !3C0000	For DI module: **(DI Latch) (00)** 00

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
0	0	1	1	1	1	0	0

Command: $02L0	Read the low latched status of module 02, (e.g., tM-P3POR3)
Response: !070700	For DIO module: **(DO Latch) (DI Latch)** 00

DO2	DO1	DO0	DI2	DI1	DI0
1	1	1	1	1	1

Related Commands: $AAC

Related Sections: 3.4.6 Configuration Page – DI, A.6 advanced DI Features (Latch and Counter)

$AAC Clear the latched status of DI channels

Description:

Clear the high/low latched status of a module

Syntax:

$AAC[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

C Command to clear the latched status

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $02C	Clear the latched status
Response: !02	The command is valid

Test the command:

When a DI signal changes from OFF to ON, it triggers a high latched status. When a DI signal changes from ON to OFF, it triggers a low latched status. If executing the **$02C** command under the low latched status, the previous triggered high latched status can be cleared.

See Chapter 4 Hardware Wiring for testing to connect both the tM-P8 (8 DI) and tM-C8 (8 DO) modules, and test the command by using DCON Utility Pro on PC.

1. On the DO configuration page (see Section 3.4.4), check the DO0 to DO7 box to set them to ON.
   At this time, the status of DI0 to DI7 is from OFF to ON to trigger a high latch.
2. Uncheck the DO0 to DO7 box to set them to OFF.
   At this time, the status of DI0 to DI7 is from ON to ON to trigger a low latch.
3. Execute the **$02C** command to clear the previous high latched statuses.

Related Commands: $AALS

Related Sections: 3.4.4 Configuration Page - DO, 3.4.6 Configuration Page – DI, A.6 advanced DI Features (Latch and Counter)

#AAN Read a counter value of the DI channel

Syntax:

#AAN[CHKSUM](CR)

Delimiter Character

AA The address of the module (Hex., 00 to FF)

N The channel to be read (0 to F)

Response:

Valid Command: **!(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

(Data) Using five hexadecimal digits to represent the DI Counter value (00000 to 65535)

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

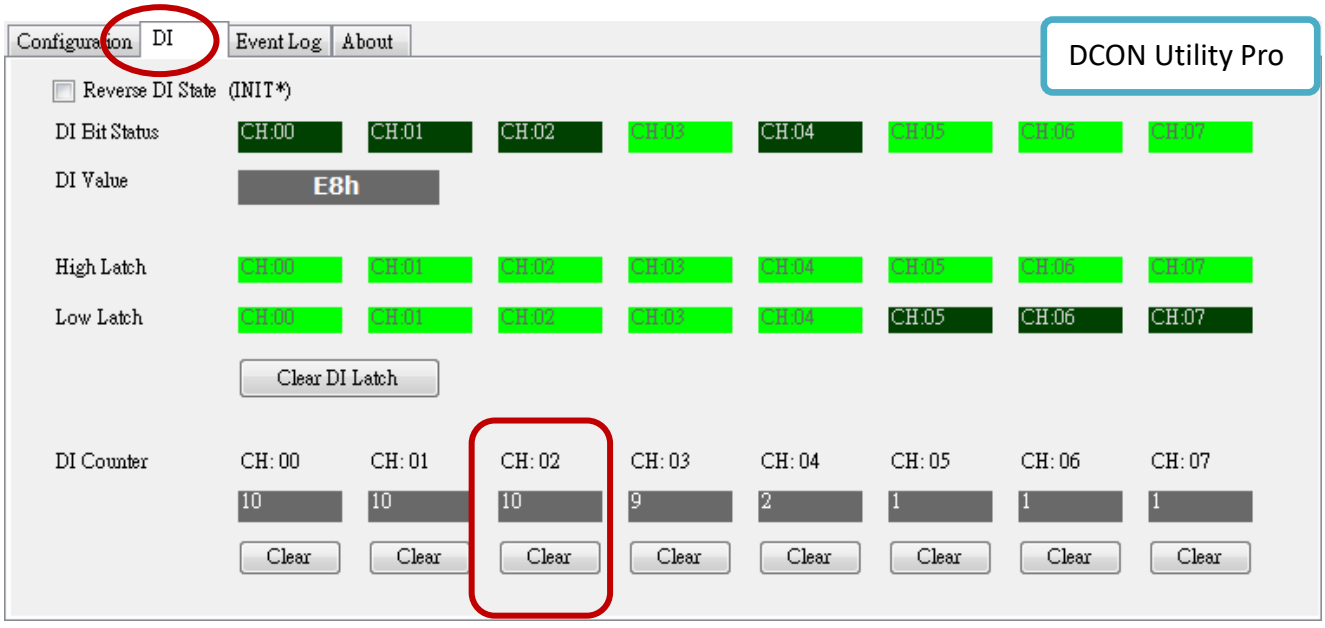
Examples:

Command: #012	Read the Counter value of channel 2 of module 01 (e.g., tM-P8)
Response: !0100010	The return value is 10

Command: #019	Read the Counter value of channel 9 of module 01
Response: ?01	The command is invalid due to no channel 9

Related Commands: $AACN

Related Sections: 3.4.6 Configuration Page – DI, A.6 advanced DI Features (Latch and Counter)



$AACN Clear a counter value of the DI channel

Syntax:

$AACN[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

C Command to clear the value of DI Counter

N The channel to be cleared (0 to F)

Response:

Valid Command: **! AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

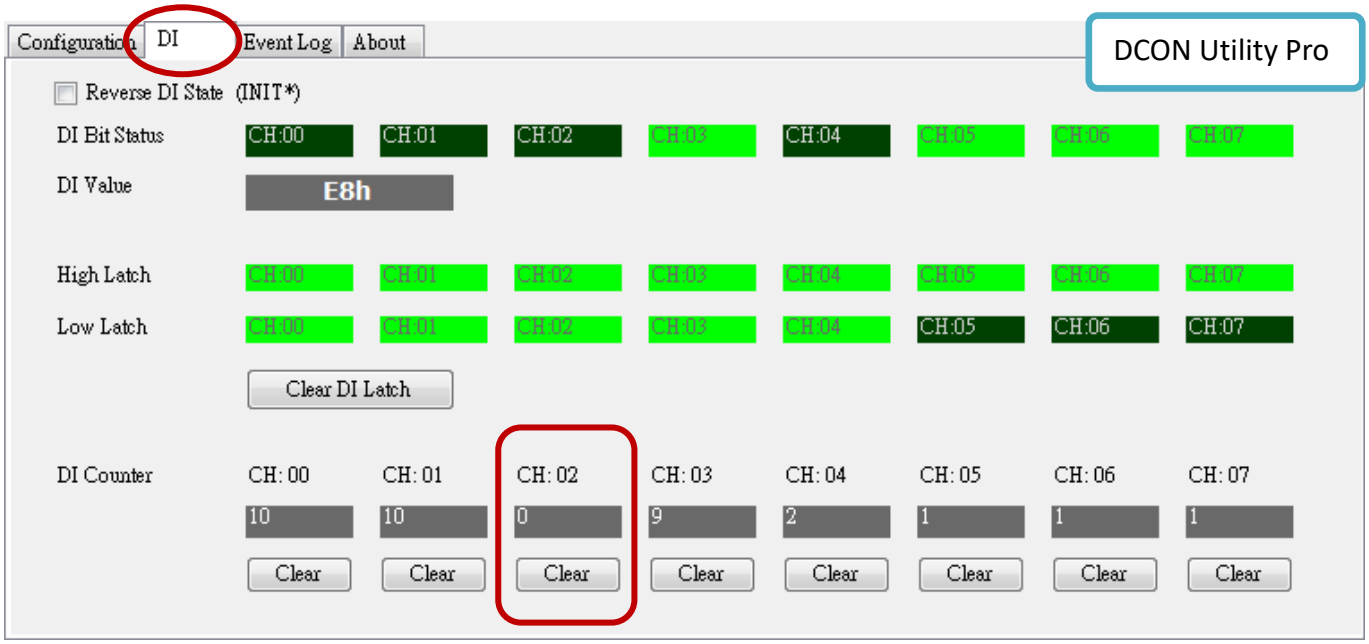
Examples:

Command: #012	Read the Counter value of channel 2 of module 01 (e.g., tM-P8)
Response: !01<u>00010</u>	The return value is 10

Command: $01C2	Clear the Counter value of channel 2 of module 01 (e.g., tM-P8)
Response: !01	The command is valid

Related Commands: #AAN

Related Sections: 3.4.6 Configuration Page – DI, A.6 advanced DI Features (Latch and Counter)



@AARECi Read a counter value of the DI channel for tM-AD4P2C2

Syntax:

@AARECi[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
RE	Command to read a DI Counter value
Ci	i, indicates the channel to be read (0 to 1)

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)
(Data)	Using five hexadecimal digits to represent the DI Counter value (00000 to 65535)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

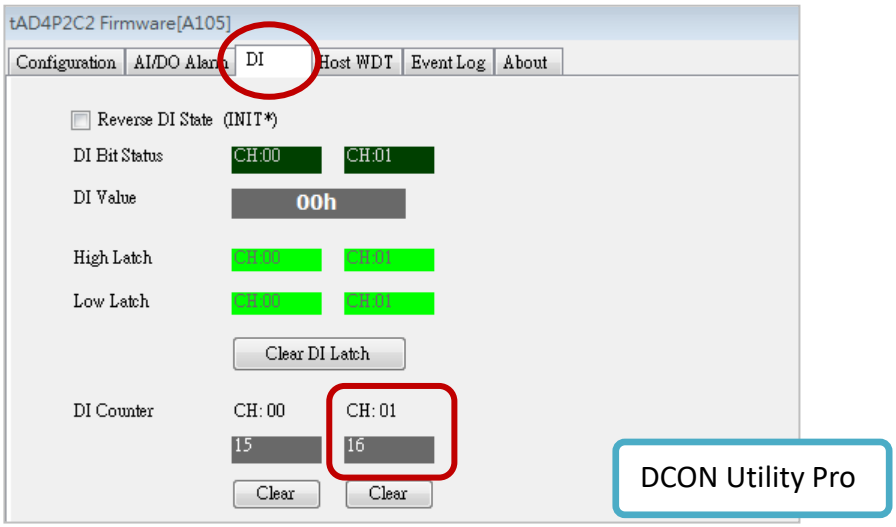
Examples:

Command: @02REC1	Read the Counter value of channel 1 of module 02
Response: !0200016	The return value is 16

Command: @02REC9	Read the Counter value of channel 9 of module 02
Response: ?02	The command is invalid due to no channel 9

Related Commands: @AACECi

Related Sections: 3.4.6 Configuration Page – DI, A.6 advanced DI Features (Latch and Counter)



@AACECi Clear a counter value of the DI channel for tM-AD4P2C2

Syntax:

@AACECi[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

CE Command to clear the value of DI Counter

Ci i, indicates the channel to be cleared (0 to 1)

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

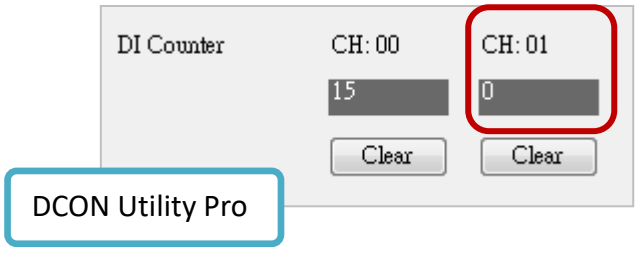
Command: @02REC1	Read the Counter value of channel 1 of module 02
Response: !0200016	The return value is 16

Command: @02CEC1	Clear the Counter value of channel 1 of module 02
Response: !02	The command is valid

Command: @02REC1	Read the Counter value of channel 1 of module 02
Response: !0200000	The return value is 0

Related Commands: @AARECi

Related Sections: 3.4.6 Configuration Page – DI, A.6 advanced DI Features (Latch and Counter)



@AARECN Read a counter value of the DI channel for tM-DA1P1R1

Syntax:

@AARECN[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
RE	Command to read a DI Counter value
CN	N, indicates the channel to be read, 0 for tM-DA1P1R1

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)
(Data)	Using five hexadecimal digits to represent the DI Counter value (00000 to 65535)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

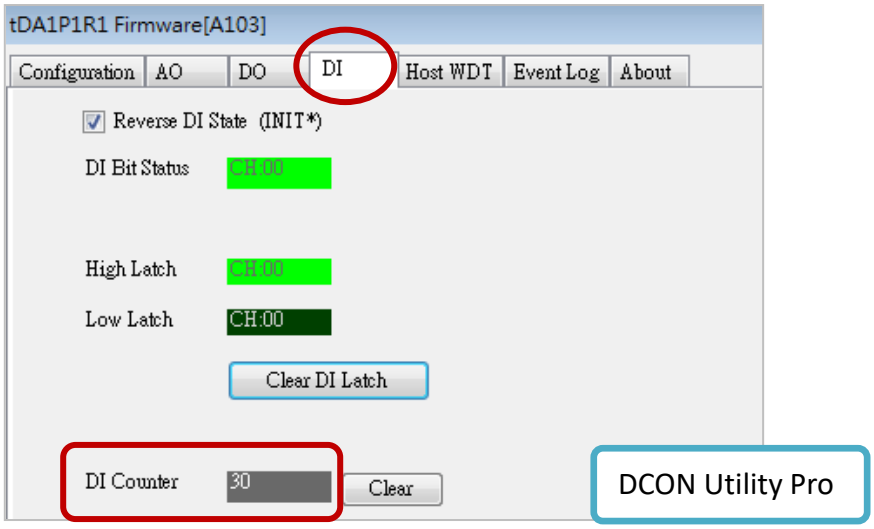
Examples:

Command @01REC0	Read the Counter value of channel 1 of module 01
Response: !0100030	The return value is 30

Command: @01REC9	Read the Counter value of channel 9 of module 01
Response: ?01	The command is invalid due to no channel 9

Related Commands: @AACECN

Related Sections: 3.4.6 Configuration Page – DI, A.6 advanced DI Features (Latch and Counter)



@AACECN Clear a counter value of the DI channel for tM-DA1P1R1

Syntax:

@AACECN[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

CE Command to clear the value of DI Counter

CN N, indicates the channel to be cleared (0)

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

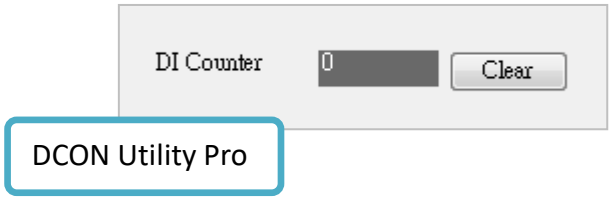
Command @01REC0	Read the Counter value of channel 1 of module 01
Response: !0100030	The return value is 30

Command: @01CEC0	Clear the Counter value of channel 1 of module 01
Response: !01	The command is valid

Command: @01REC0	Read the Counter value of channel 1 of module 01
Response: !0100000	The return value is 0

Related Commands: @AARECN

Related Sections: 3.4.6 Configuration Page – DI, A.6 advanced DI Features (Latch and Counter)



@AAEATCi Enable the alarm function for the AI channel

Note: The command is available for tM-AD4P2C2.

Syntax:

@AAEATCi[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

T Alarm Type

M: Momentary; L: Latched

Ci i, indicates the channel to enable alarm

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @01EAMC0	Enable the alarm of channel 0 of module 01, and set the alarm type to Momentary
Response: !01	The command is valid

Command: @02EALC1	Enable alarm of channel 1 of module 02, and set the alarm type to Latch
Response: !01	The command is valid

Related Commands: @AADACi, @AARACi

Related Sections: 3.4.3 Configuration Page - AI/DO Alarm

@AADACi Disable the alarm function for the AI channel

Note: The command is available for tM-AD4P2C2.

Syntax:

@AADACi[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
DA	Command to disable the alarm
Ci	i**,** indicates the channel to disable the alarm

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @02DAC0	Disable the alarm of channel 0 of module 02
Response: !01	The command is valid

Related Commands: @AAEATCi

Related Sections: 3.4.3 Configuration Page - AI/DO Alarm

@AARACi Read the setting of alarm type

Note: The command is available for tM-AD4P2C2.

Syntax:

@AARACi[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

RA Command to read the alarm setting

Ci i**,** indicates the channel to read the alarm setting

Response:

Valid Command: **!AAS**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

S Alarm Type

0: Disable; 1: Momentary; 2: Latch

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @01RAC0	Read the alarm setting of channel 0 of module 01
Response: !011	The alarm type is Momentary

Command: @01RAC1	Read the alarm setting of channel 1 of module 01
Response: !012	The alarm type is Latch

Related Commands:

@AARAO, @AARHCi, @AARLCi, @AAHI(data)Ci, @AALO(data)Ci, @AACHCi, @AACLCi

Related Sections:

3.4.3 Configuration Page - AI/DO Alarm

@AARAO Read the status of high/low alarm

Note: The command is available for tM-AD4P2C2.

Syntax:

@AARAO[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

RAO Command to read the alarm setting

Response:

Valid Command: **!AAHHLL**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

HH Using two hexadecimal digits to represent the currently activated high alarms.

LL Using two hexadecimal digits to represent the currently activated low alarm.
Bit 0 corresponds to channel 0, and bit 1 corresponds to channel 1, etc.
When the bit is 1, means there is an activated high/low alarm associated with the channel.
When the bit is 0, means there is no activated alarm.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @01RAO	Read the currently activated alarms of module 01
Response: !010003	There are the activated low alarm on channel 0 and 1

Command: @02RAO	Read the currently activated alarms of module 02
Response: !020200	There is an activated high alarm on channel 1

Related Commands:

@AARACi, @AARHCi, @AARLCi, @AAHI(data)Ci, @AALO(data)Ci, @AACHCi, @AACLCi

Related Sections:

3.4.3 Configuration Page - AI/DO Alarm

@AARHCi Read the value of high alarm limit

Note: The command is available for tM-AD4P2C2.

Syntax:

@AARHCi[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

RH Command to read the high alarm limit

Ci i, indicates the channel to read the high alarm limit

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) The high alarm limit in engineering format

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	**@02RHC0**	Read the high alarm limit of channel 0 of module 02
Response:	**!02+09.000**	The high alarm limit is + 9.0

Command:	**@02RHC1**	Read the high alarm limit of channel 1 of module 02
Response:	**!02+08.000**	The high alarm limit is +8.0

Related Commands:

@AARAO, @AARLCi, @AAHI(data)Ci, @AALO(data)Ci, @AACHCi, @AACLCi

Related Sections:

3.4.3 Configuration Page - AI/DO Alarm

@AARLCi Read the value of low alarm limit

Note: The command is available for tM-AD4P2C2.

Syntax:

@AARLCi[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

RL Command to read the low alarm limit

Ci i, indicates the channel to read the low alarm limit

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) The low alarm limit in engineering format

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @02RLC0	Read the low alarm limit of channel 0 of module 02
Response: !02-09.000	The low alarm limit is - 9.0

Command: @02RLC1	Read the low alarm limit of channel 1 of module 02
Response: !02-08.000	The low alarm limit is -8.0

Related Commands: @AARAO, @AARHCi, @AAHI(data)Ci, @AALO(data)Ci, @AACHCi, @AACLCi

Related Sections: 3.4.3 Configuration Page - AI/DO Alarm

@AAHI(data)Ci Set the value of high alarm limit

Note: The command is available for tM-AD4P2C2.

Syntax:

@AAHICi[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
HI	Command to set the high alarm limit
(data)	The high alarm limit in engineering format
Ci	i**,** indicates the channel to set the high alarm limit

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @01HI+10.000C0	Set the high alarm limit of channel 0 of module 01 to + 10.0
Response: !01	The command is valid

Command: @01HI+09.000C1	Set the high alarm limit of channel 1 of module 01 to + 9.0
Response: !01	The command is valid

Related Commands: @AARAO, @AARHCi, @AARLCi, @AALO(data)Ci, @AACHCi, @AACLCi

Related Sections: 3.4.3 Configuration Page - AI/DO Alarm

@AALO(data)Ci Set the value of low alarm limit

Note: The command is available for tM-AD4P2C2.

Syntax:

@AALOCi[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
LO	Command to set the low alarm limit
(data)	The low alarm limit in engineering format
Ci	i**,** indicates the channel to set the low alarm limit

Response:

Valid Command: **!AA**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @01LO-05.000C0	Set the low alarm limit of channel 0 of module 01 to -5.0.
Response: !01	The command is valid

Command: @01HI+01.000C1	Set the low alarm limit of channel 1 of module 01 to + 1.0
Response: !01	The command is valid

Related Commands: @AARAO, @AARHCi, @AARLCi, @AAHI(data)Ci, @AACHCi, @AACLCi

Related Sections: 3.4.3 Configuration Page - AI/DO Alarm

@AACHCi Clear the status of a high latched alarm

Note: The command is available for tM-AD4P2C2.

Syntax:

@AACHCi[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

CH Command to clear the high latch alarm

Ci i**,** indicates the channel to clear the high latch alarm

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @01CHC1	Clear the high latched alarm of channel 0 of module 01
Response: !01	The command is valid

Related Commands: @AARAO, @AARHCi, @AARLCi, @AAHI(data)Ci, @AALO(data)Ci, @AACLCi

Related Sections: 3.4.3 Configuration Page - AI/DO Alarm

@AACLCi Clear the status of a low latched alarm

Note: The command is available for tM-AD4P2C2.

Syntax:

@AACLCi[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

CL Command to clear the low latch alarm

Ci i**,** the channel to clear the low latch alarm

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @01CLC1	Clear the low latched alarm of channel 0 of module 01
Response: !01	The command is valid

Related Commands: @AARAO, @AARHCi, @AARLCi, @AAHI(data)Ci, @AALO(data)Ci, @AACHCi

Related Sections: 3.4.3 Configuration Page - AI/DO Alarm

~AAEV Enable/Disable the calibration

Note: The command is available for AI and tM-AD4P2C2 modules.

Syntax:

~AAEV[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

E Command to enable/disable the calibration

V 1**:** Enable calibration ; 0: Disable calibration

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	$010	Perform a span calibration on module 01
Response:	?01	The command is invalid because the calibration is not yet enabled

Command:	**~01E1**	Enable the calibration on module 01
Response:	**!01**	The command is valid

Command:	$010	Perform a span calibration on module 01
Response:	!01	The command is valid

Related Commands: $AA0, $AA1, $AA0Ci, $AA1Ci (Only available for tM-AD2)

Related Sections: 3.6 Calibration

$AA1 Perform zero calibration (V, A)

Note: The command is available for AI and tM-AD4P2C2 modules. Before using it, execute the **~AAEV** command first.

Syntax:

$AA1[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

1 Command to perform a zero calibration

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $011	Perform a zero calibration on module 01
Response: ?01	The command is invalid because the calibration is not yet enabled

Command: ~01E1	Enable the calibration on module 01
Response: !01	The command is valid

Command: $011	Perform a zero calibration on module 01
Response: !01	The command is valid

Related Commands: ~AAEV, $AA0, $AA0Ci

Related Sections: 3.6.2 Calibrate AI Module

$AA1Ci Perform zero calibration

Note: The command is available for tM-AD4P2C2. Before using it, executing the **~AAEV** command first.

Syntax:

$AA1Ci[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

1 Command to perform a zero calibration

Ci i, indicates the channel to be calibrated

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $011C0	Perform a zero calibration on channel 0 of module 01
Response: ?01	The command is invalid because the calibration is not yet enabled

Command: ~01E1	Enable the calibration on module 01
Response: !01	The command is valid

Command: $011C0	Perform a zero calibration on channel 0 of module 01
Response: !01	The command is valid

Related Commands: ~AAEV, $AA0Ci

Related Sections: 3.6.2 Calibrate AI Module

$AA0 Perform span calibration (V, Ω)

Note: The command is available for tM-AD5, AD8, TH8, and tM-AD4P2C2.
Before using it, executing the **~AAEV** command first.

Syntax:

$AA0[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

0 Command to perform a span calibration

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $010	Perform a span calibration on module 01
Response: ?01	The command is invalid because the calibration is not yet enabled

Command: ~01E1	Enable the calibration on module 01
Response: !01	The command is valid

Command: $010	Perform a span calibration on module 01
Response: !01	The command is valid

Related Commands: ~AAEV, $AA1

Related Sections: 3.6.2 Calibrate AI Module

$AA0Ci Perform span calibration (A)

Note: The command is available for tM-AD2, AD5C, AD8C, and tM-AD4P2C2.
Before using it, executing the **~AAEV** command first.

Syntax:

$AA0Ci[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

0 Command to perform a span calibration

Ci i, indicates the channel to be calibrated

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	~01E1	Enable the calibration on module 01
Response:	!01	The command is valid

Command:	**$010C0**	Perform a span calibration on channel 0 of module 01
Response:	**!01**	The command is valid

Related Commands: ~AAEV, $AA1, $AA1Ci (Only available for tM-AD2)

Related Sections: 3.6.2 Calibrate AI Module

$AA0N Perform zero calibration

Note: The command is available for tM-AD1P1R1.

Syntax:

$AA0[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

0 Command to perform a zero calibration

N The channel to be calibrated, 0 for tM-AD1P1R1

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: 0100	Perform a zero calibration on channel 0 of module 01
Response: !01	The command is valid

Related Commands: $AA1N, $AA3NVV

Related Sections: 3.6.1 Calibrate AO module

$AA1N Perform span calibration

Note: The command is available for tM-AD1P1R1.

Syntax:

$AA1[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

1 Command to perform a span calibration

N The channel to be calibrated, 0 for tM-AD1P1R1

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $0110	Perform a span calibration on channel 0 of module 01
Response: !01	The command is valid

Related Commands: $AA0N, $AA3NVV

Related Sections: 3.6.1 Calibrate AO module

$AA3NVV Adjust AO value for calibration

Note: The command is available for tM-AD1P1R1.

Syntax:

$AA3NVV[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

3 Command to adjust the AO value

N The channel to adjust the value, 0 for tM-AD1P1R1

VV Using two-digits hexadecimal value to adjust the analog output
00 to 5F, indicates to increase 0 to 95 counts ;
FF to A1, indicates to decrease 1 to 95 counts
Each count is about 4.88uA or 2.44mV

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $01301F	Set the value of channel 0 of module 01 to increase by 31 counts (It is about +151.28 uA or +75.64 mV)
Response: !01	The command is valid

Related Commands: $AA0N, $AA1N

Related Sections: 3.6.1 Calibrate AO module

$AAS1 Reload the default calibration parameters

Note: The command is available for the tM-TH8 module.

Syntax:

$AAS1[CHKSUM](CR)

$ Delimiter Character

AA The address of the module (Hex., 00 to FF)

S1 Command to reload the factory default calibration parameters

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: $01S1	Reload the factory default calibration parameters for module 01
Response: !01	The command is valid

Related Commands: ~AAEV

Related Sections: 3.6.2 Calibrate AI Module

@AAA3Ci Read the temperature offset of a channel

Note: The command is available for the tM-TH8 module.

Syntax:

@AAA3Ci[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

A3 Command to read the temperature offset

Ci i, indicates the channel to be read (0 to 7)

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) Using two-digits hexadecimal value to represent the temperature offset

01: indicates 0.1 **02**: indicates 0.2 **FF**: indicates -0.1 **FE**: indicates -0.2

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	**@03A3C2**	Read the temperature offset of channel 02 of module 03
Response:	**!030A**	The offset value is 1.0

Command:	**@03A3C5**	Read the temperature offset of channel 05 of module 03
Response:	**!03F0**	The offset value is -1.6.

Related Commands: @AAA2CiToo

Related Sections: 3.3.3 Configuration Page 03 - AI (for tM-TH8)

@AAA2CiToo Set the temperature offset of a channel

Note: The command is available for the tM-TH8 module.

Syntax:

@AAA2CiToo[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
A2	Command to set the temperature offset
Ci	i, indicates the channel to be set (0 to 7).
Too	oo, indicates the temperature offset in 0.1℃/℉, ranges from -12.8 to 12.7 Using the hexadecimal value, **01:** indicates 0.1, **02**: indicates 0.2, **FF**: indicates -0.1, **FE**: indicates -0.2

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @03A2C2T0A	Set the temperature offset of channel 02 of module 03 to 1.0
Response: !03	The command is valid

Command: @03A2C5TF0	Set the temperature offset of channel 05 of module 03 to -1.6
Response: !03	The command is valid

Related Commands: @AAA3Ci

Related Sections: 3.3.3 Configuration Page 03 - AI (for tM-TH8)

@AAA7Ci Read the resistance offset of a channel

Note: The command is available for the tM-TH8 module.

Syntax:

@AAA7Ci[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

A7 Command to read the resistance offset

Ci i, indicates the channel to be read (0 to 7)

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

(Data) Using two-digits hexadecimal value to represent the resistance offset

01: indicates 0.1 **02**: indicates 0.2 **FF**: indicates 25.5 **FE**: indicates 25.4

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @03A7C2	Read the resistance offset of channel 02 of module 03
Response: !030A	The offset value is 1.0

Command: @03A7C5	Read the resistance offset of channel 05 of module 03
Response: !03F0	The offset value is 24.0.

Related Commands: @AAA6CiRrr

Related Sections: 3.4.1 Configuration Page - User Defined Type

@AAA6CiRrr Set the resistance offset of a channel

Note: The command is available for the tM-TH8 module.

Syntax:

@AAA6CiRrr[CHKSUM](CR)

@ Delimiter Character

AA The address of the module (Hex., 00 to FF)

A6 Command to set the resistance offset

Ci i, indicates the channel to be set (0 to 7)

Rrr rr, indicates the resistance offset in 0.1 ohms, ranges from 0.0 to 25.5 ohms.
01: indicates 0.1 **02**: indicates 0.2 **FF**: indicates 25.5 **FE**: indicates 25.4

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command:	**@03A6C2T0A**	Set the resistance offset of channel 02 of module 03 to 1.0
Response:	**!03**	The command is valid

Command:	**@03A6C5TF0**	Set the resistance offset of channel 05 of module 03 to 24.0.
Response:	**!03**	The command is valid

Related Commands: @AAA7Ci

Related Sections: 3.4.1 Configuration Page - User Defined Type

~AAD Read the temperature scale (℃/℉)

Note: The command is available for the tM-TH8 module.

Syntax:

~AAD[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

D Command to read the temperature scale

Response:

Valid Command: **!AAT** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

T Temperature scale

0: Celsius (by default); 1: Fahrenheit.

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~03D	Read the temperature scale of module 03
Response: !010	Return 0, stands for Celsius

Command: ~03D	Read the temperature scale of module 03
Response: !011	Return 1, stands for Fahrenheit

Related Commands: ~AADT

~AADT Set the temperature scale (℃/℉)

Note: The command is available for the tM-TH8 module.

Syntax:

~AADT[CHKSUM](CR)

~ Delimiter Character

AA The address of the module (Hex., 00 to FF)

D Command to set the temperature scale

T C: Celsius (by default) ; F: Fahrenheit.

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: ~03DC	Set the temperature scale of module 03 to Celsius
Response: !01	The command is valid

Command: ~03DF	Set the temperature scale of module 03 to Fahrenheit
Response: !01	The command is valid

Related Commands: ~AAD

@AAGxTtt Read the Steinhart-Hart coefficient of a user-defined type

Note: The command is available for the tM-TH8 module.

Syntax:

@AAGxTtt[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
G	Command to read the Steinhart-Hart coefficient
x	**A:** Read Steinhart-Hart coefficient A
	B: Read Steinhart-Hart coefficient B
	C: Read Steinhart-Hart coefficient C
Ttt	tt, the code of the user-defined type (70 to 77) for reading the Steinhart-Hart coefficient

Response:

Valid Command: **!AA(Data)**[CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)
(Data)	Using eight-digits hexadecimal value to represent the Steinhart-Hart coefficient in IEEE-754 format

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @03GAT70	Read the coefficient A of type code 70 of module 03
Response: !033A94030A	The return value 3A94030A equals to 1.129241 x 10^{-3}

Command: @03GBT70	Read the coefficient B of type code 70 of module 03
Response: !0339757ACF	The return value 39757ACF equals to 2.341077 x 10^{-4}

Command: @03GCT70	Read the coefficient C of type code 70 of module 03
Response: !0333BC73A5	The return value 33BC73A5 equals to 8.775468 x 10^{-8}

Related Commands: @AASxTttC(data), @AARTTttR(Data)

Related Sections: 3.4.1 Configuration Page - User Defined Type, B.8.1 Steinhart-Hart coefficient (User-defined Type)

@AASxTttC(data) Set the Steinhart-Hart coefficient of a user-defined type

Note: The command is available for the tM-TH8 module.

Syntax:

@AASxTttC(data)[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
S	Command to set the Steinhart-Hart coefficient
x	**A:** Set Steinhart-Hart coefficient A **B:** Set Steinhart-Hart coefficient B **C:** Set Steinhart-Hart coefficient C
Ttt	tt, the code of the user-defined type (70 to 77) for setting the Steinhart-Hart coefficient
C(data)	(data), using eight-digits hexadecimal value to represent the Steinhart-Hart coefficient in IEEE-754 format

Response:

Valid Command: **!AA** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @03SAT70C3A94030A	Set the coefficient A of type code 70 of module 03 to “3A94030A”, also equals to 1.129241 x 10-3
Response: !03	The command is valid

Command: @03SBT70C39757ACF	Set the coefficient B of type code 70 of module 03 to “39757ACF”, also equals to 2.341077 x 10^{-4}
Response: !03	The command is valid

Command: @03SCT70C33BC73A5	Set the coefficient C of type code 70 of module 03 to 33BC73A5, which also equals 8.775468 x 10^{-8}
Response: !03	The command is valid

Related Commands: @AAGxTtt, @AARTTttR(Data)

Related Sections: 3.4.1 Configuration Page - User Defined Type, B.8.1 Steinhart-Hart coefficient (User-defined Type)

@AARTTttR(Data) Read the scaling temperature based on input resistance

Description:

After configuring the Steinhart-Hart coefficients by using the @AASxTttC(data) command, also using this @AARTTttR(Data) command to read the temperature associated with the resistance.

Note: The command is available for the tM-TH8 module.

Syntax:

@AARTTttR(Data)[CHKSUM](CR)

@	Delimiter Character
AA	The address of the module (Hex., 00 to FF)
RT	Command to read the temperature associated with the resistance
Ttt	tt, the user-defined type code (70 to 77, see Appendix B.8 Data Ranges for tM-TH8)
R(Data)	(Data), consists of seven decimal digits, or five digits, decimal point, and one additional digit, to represent the input resistance

Response:

Valid Command: **!AA(Data)** [CHKSUM](CR)

Invalid Command: **?AA**[CHKSUM](CR)

!	Delimiter character for a valid command
?	Delimiter character for an invalid command
AA	The address of the module (Hex., 00 to FF)
(Data)	It consists of a +/- sign, three digits, a decimal point, and two additional digits to represent the temperature associated with the input resistance

Note: There will be no response when the command syntax is incorrect, or a communication error occurs, or the specified address is not correct.

Examples:

Command: @03RTT70R0104500	Read the temperature associated with 104500 ohms of type 70 of module 03
Response: !03-032.64	The return value is -32.64 degree

Command: @03RTT70R00801.2	Read the temperature associated with 801.2 ohms of type 70 of module 03
Response: !03+072.62	The return value is 72.62 degree

Related Commands: @AAGxTtt, @AASxTttC(data)

Related Sections:

3.4.1 Configuration Page - User Defined Type, B.8.1 Steinhart-Hart coefficient (User-defined Type)

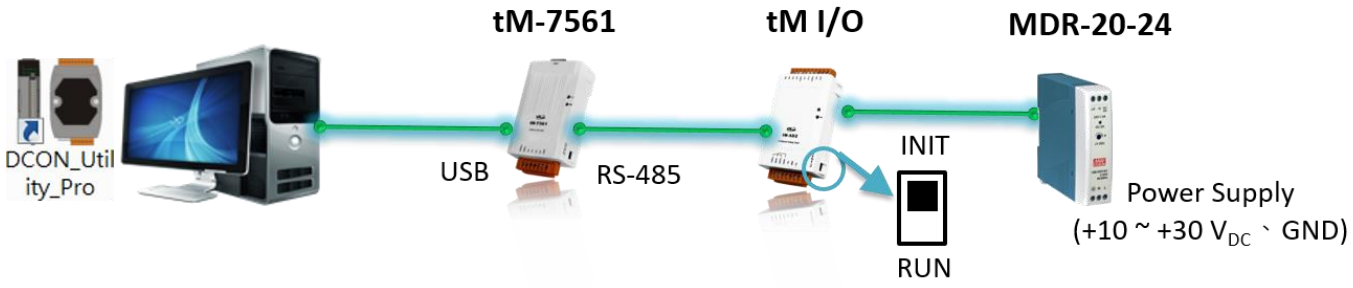
Note: Using the ~AADT command to set the temperature scale;
Using the ~AAD command to read the settings.

Appendix F　Quickly Save and Apply I/O Configurations

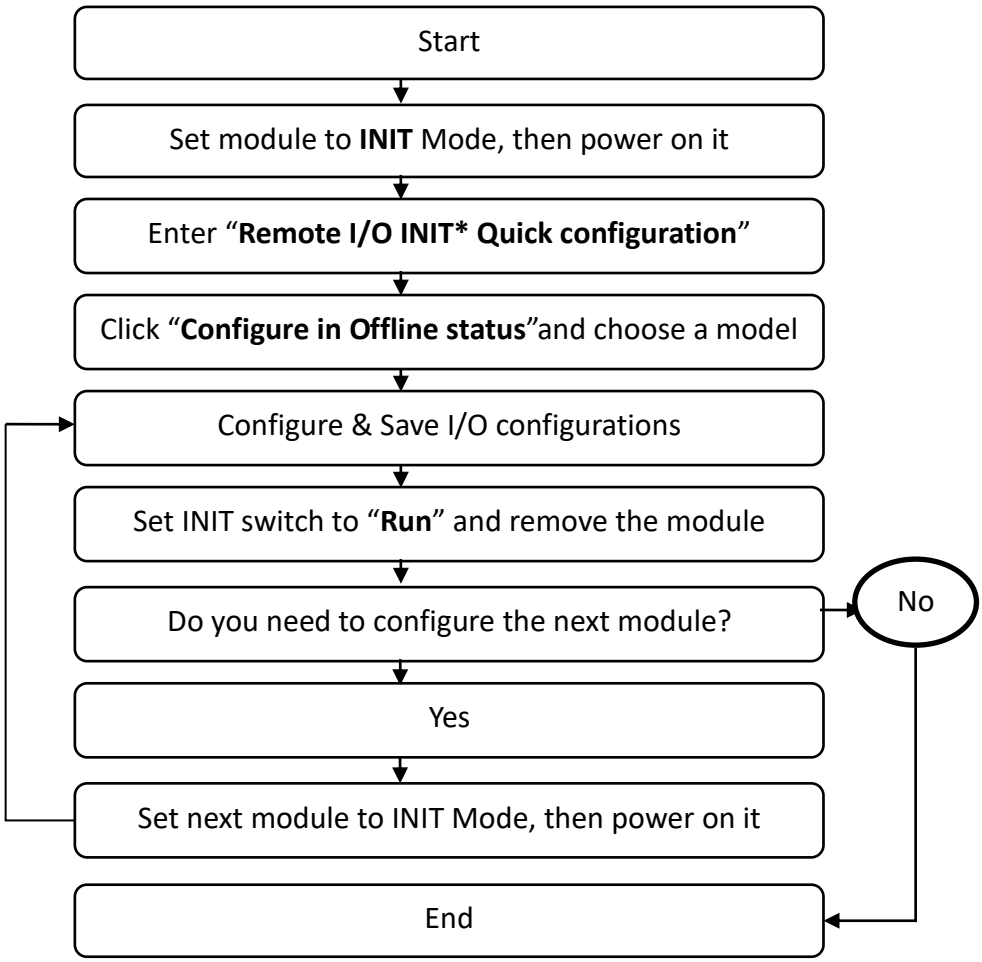
The Chapter will describe the “Remote I/O Configuration” function in DCON Utility Pro. If a case requires 250 modules with the same model to be configured, the user can only set one and then write configurations to the module one-by-one with a little bit of change. Also, save it as a file so that you can quickly write configurations to the module next time. This makes the setting to be easy.

Note:

- Configure one module at a time and turn on the module in INIT mode to allow writing settings.
- Support offline configuration. To configure and save the settings without connecting the module.
- When using a USB to RS-485 converter, make sure the driver has been installed.



Procedure of “Remote I/O INIT* Quick Configuration”:

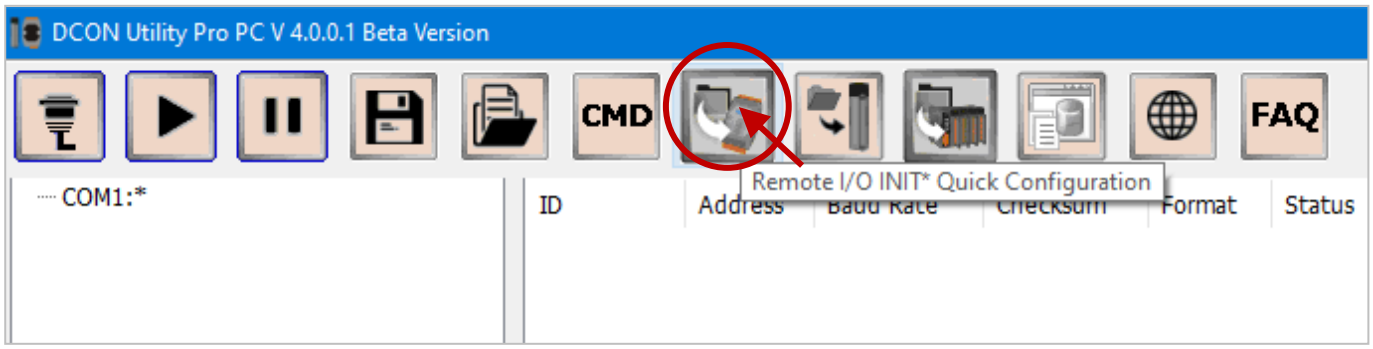


F.1 Set, Write, and Save I/O Configurations

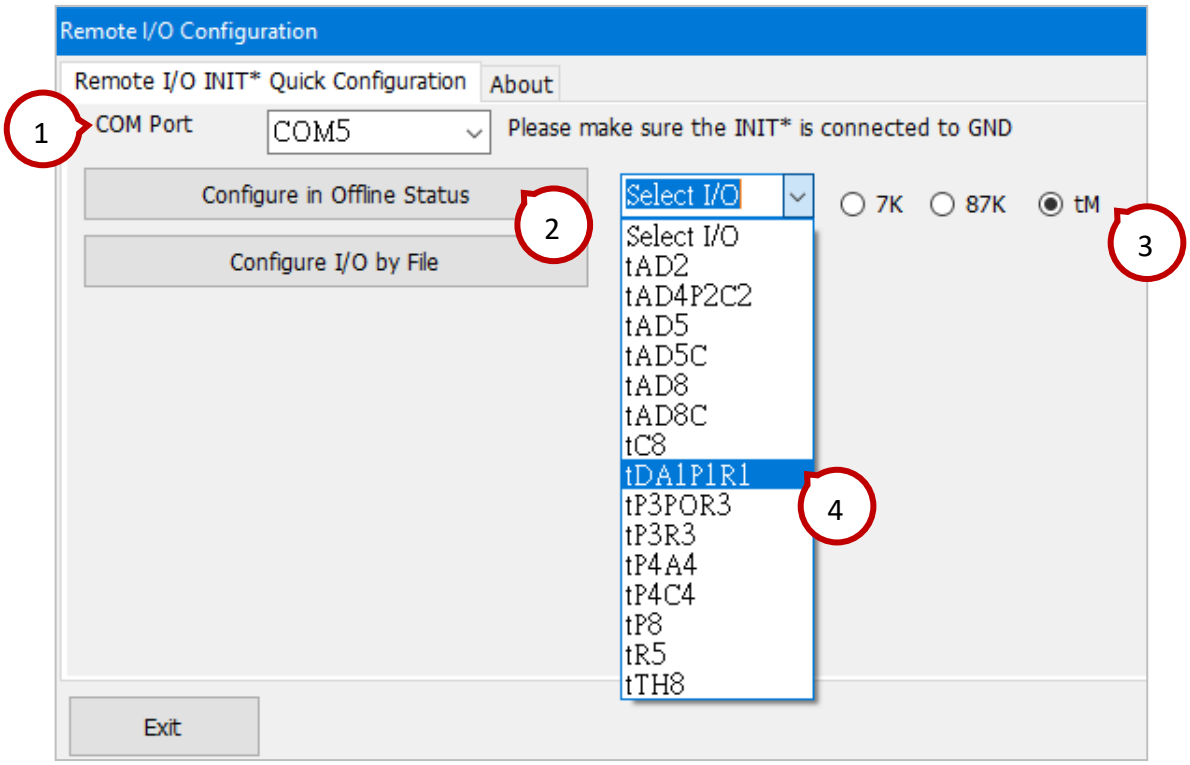
Follow the steps below to set and write configurations to the module, then save it as a file.

Set I/O Configurations

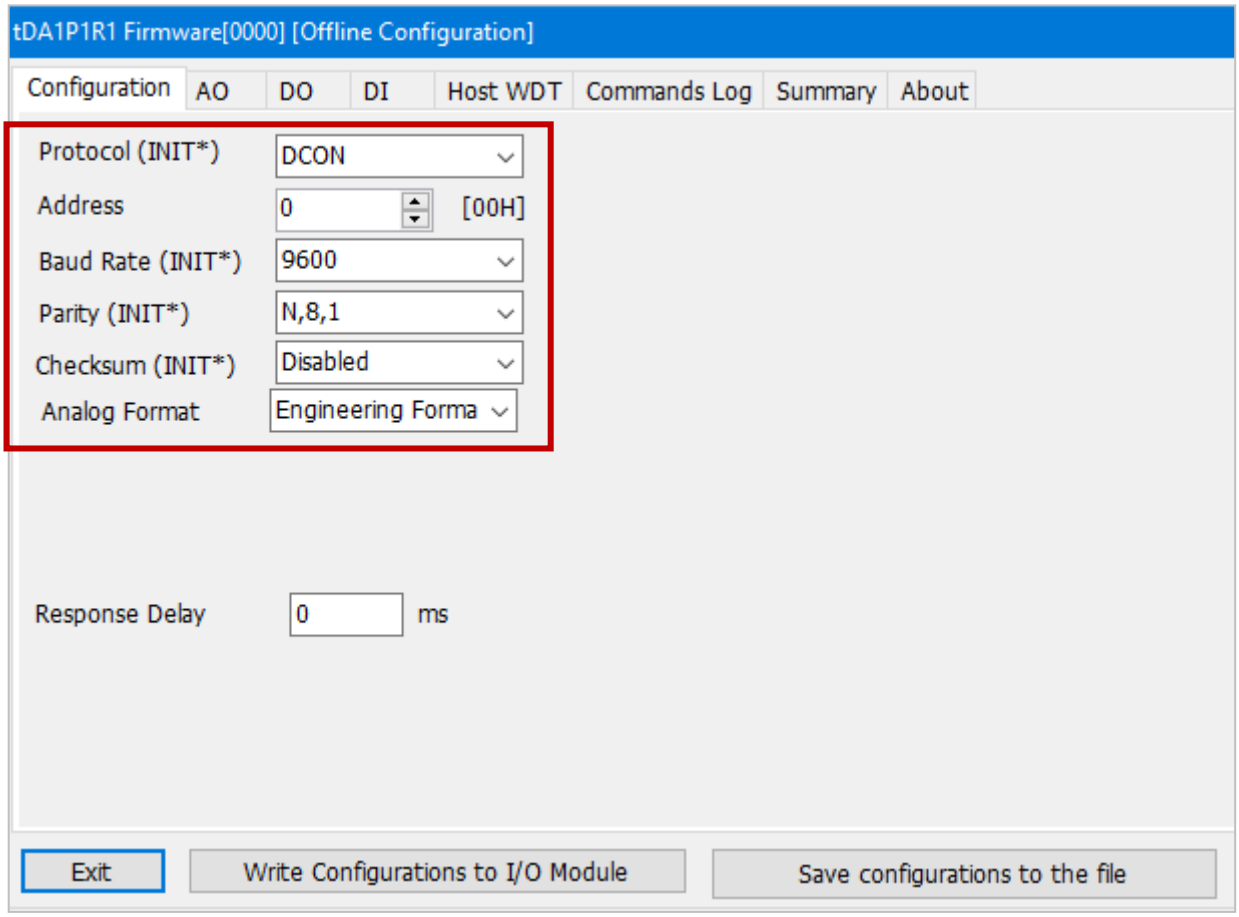
1. Click the **Remote I/O INIT* Quick Configuration** button in DCON Utility Pro on PC.



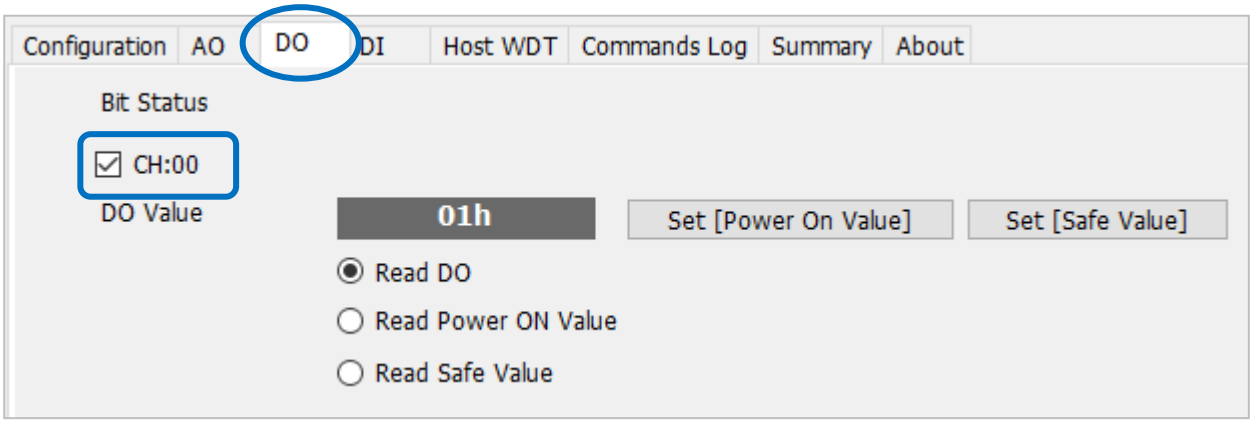
2. Choose a COM Port (Defaults = COM1), click the **Configure in Offline Status** button, and select **tM**, then choose a model.



3. In the Offline Configuration window, the initial parameters of the module (e.g., tM-DA1P1R1) will be displayed on setting pages. You can change any settings.



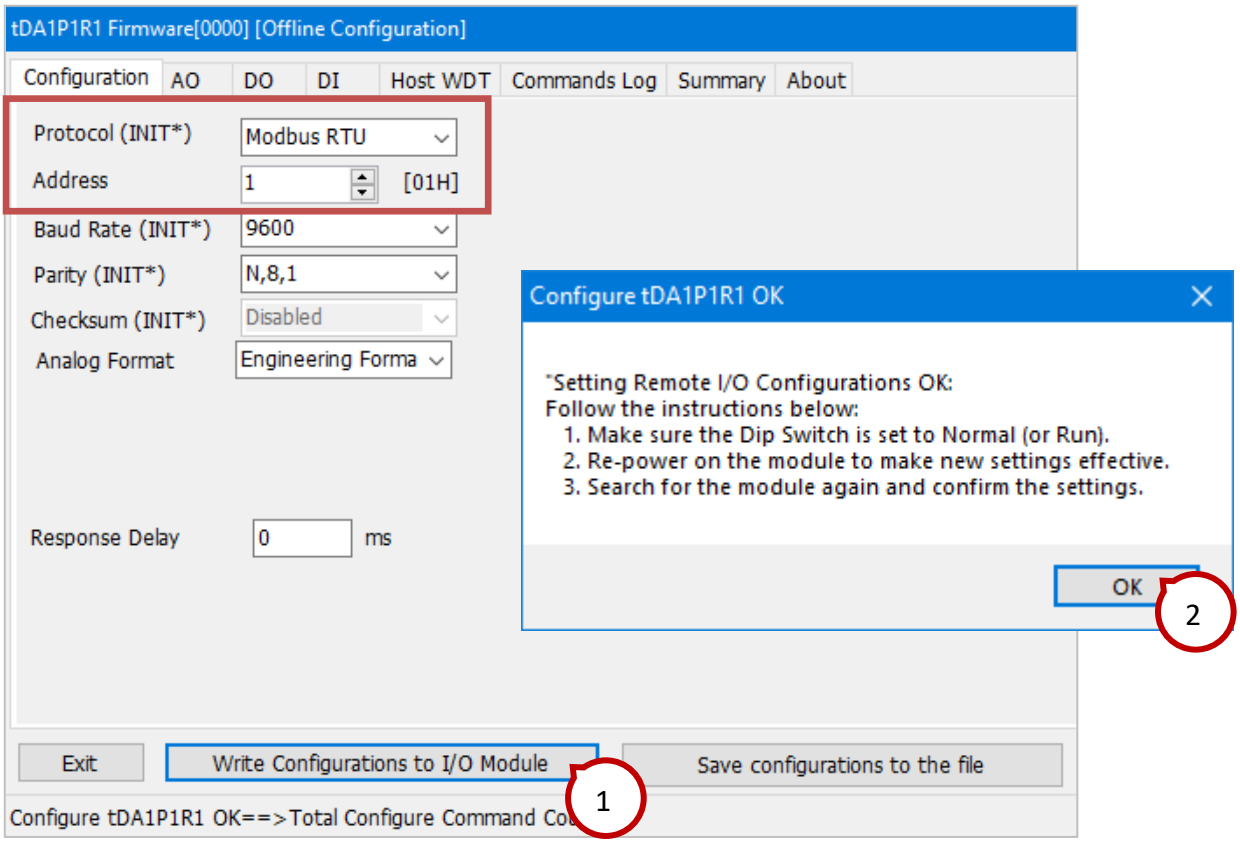
Note: In Offline mode, the hardware will not be changed by the settings.



Write I/O Configurations

After completing the setting, make sure that the module is connected with the PC under INIT mode and the COM Port is set (step 2). Then, continue the next step.

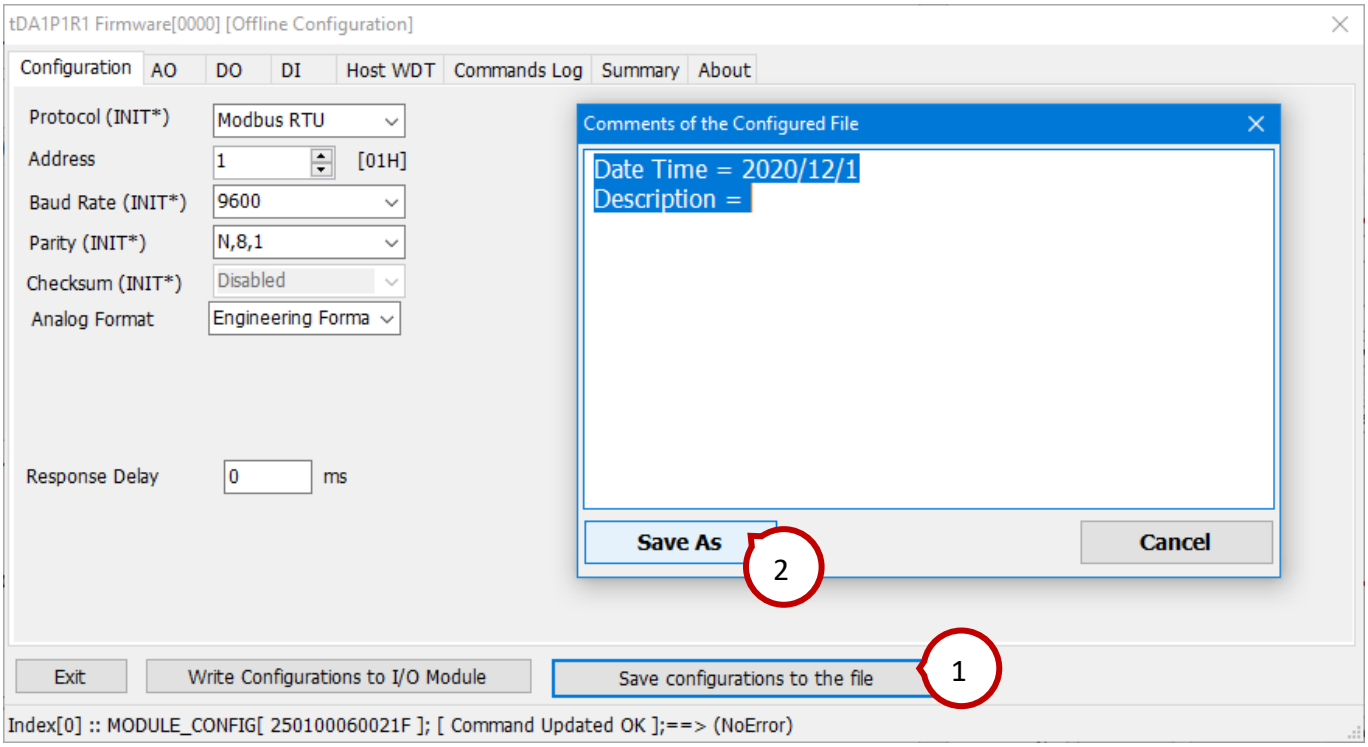
4. Click the **Write Configurations to I/O Module** button to write parameters to the I/O module and click the **OK** button.



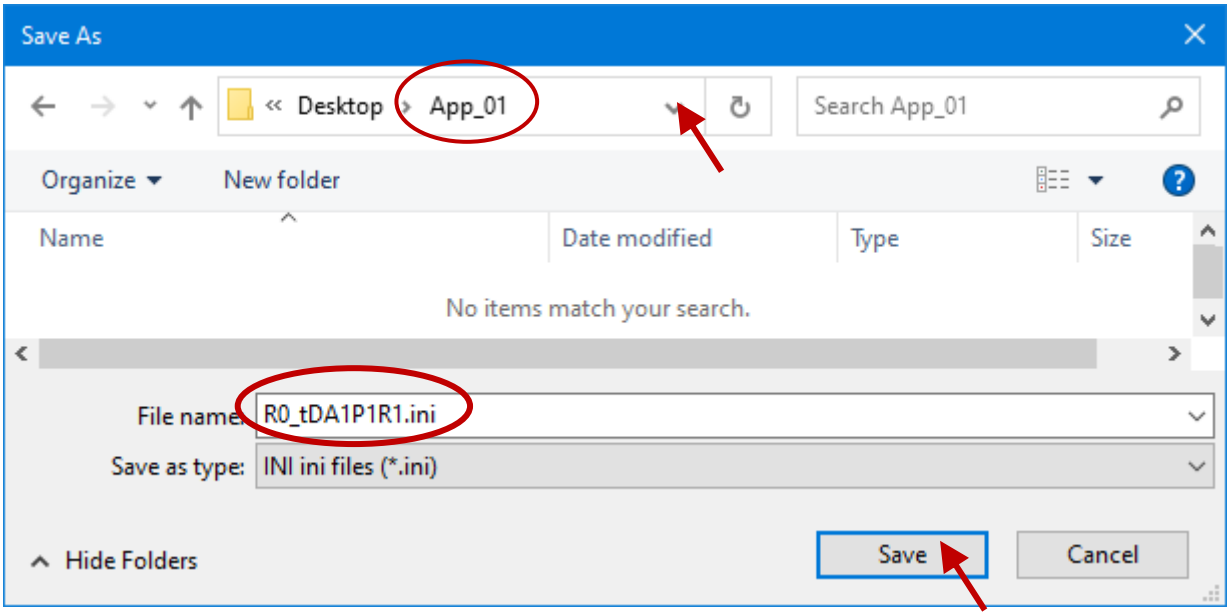
5. Set the DIP switch to the **Run** position and remove the module. If you want to set the next one, keep the window open.
6. Install the module with the same model after setting the DIP switch to INIT and follow steps 3 to 6 to configure the module, e.g., set the address to “2”.

Save I/O Configurations

7. After configuring all modules, click the **Save Configuration to the File** button and click the **Save As** button.

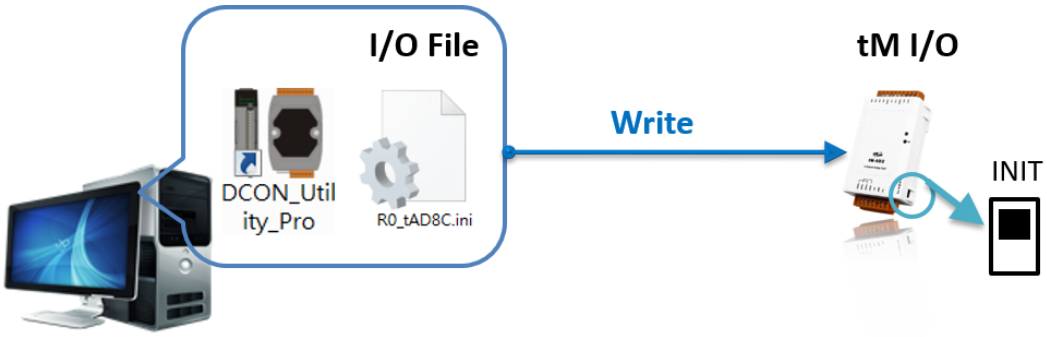


Also, create a folder according to the project name (e.g., App_01) and save it as a configuration file to be used for the next order.



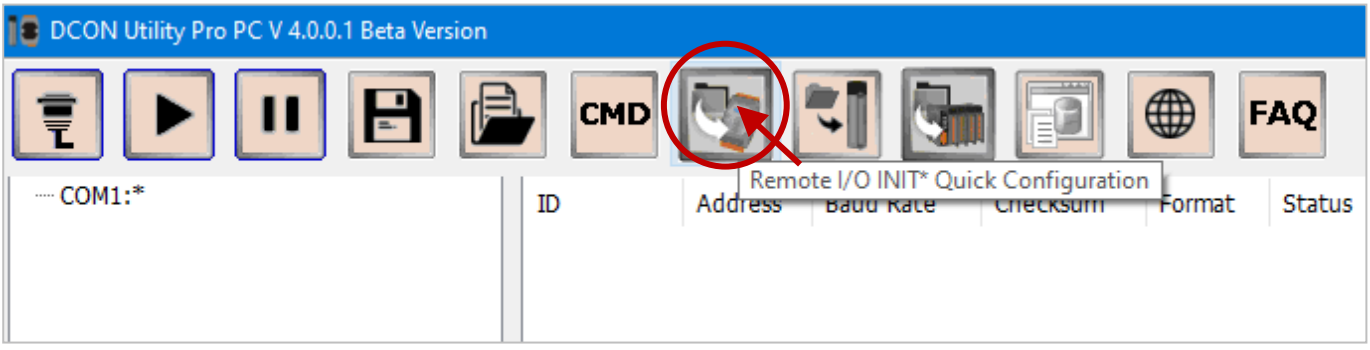
F.2 Configure the I/O Module by a Configuration File

The following will describe how to open the stored file and write configurations to the module.

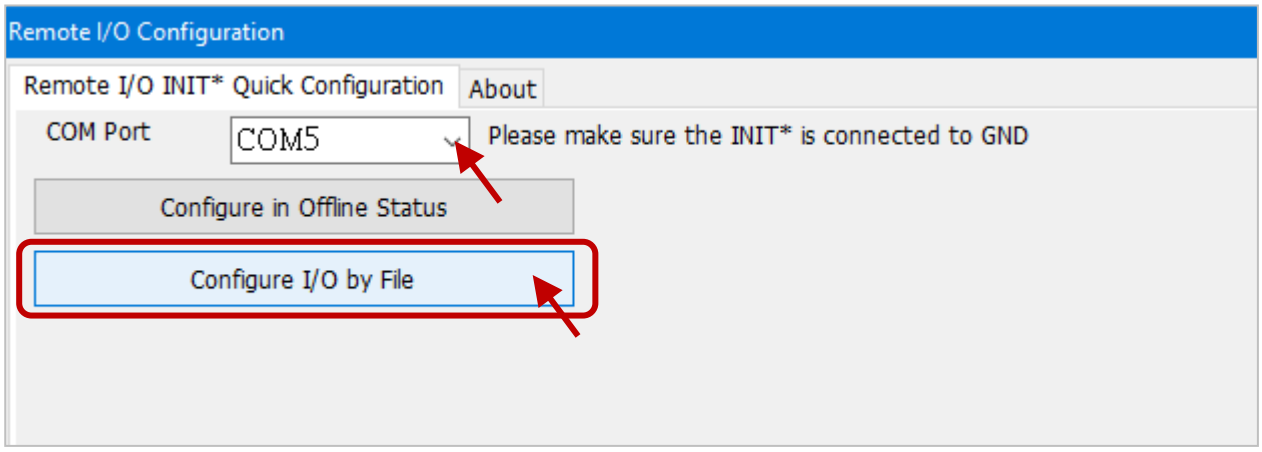


Note: Only configure one module at a time under Init mode.

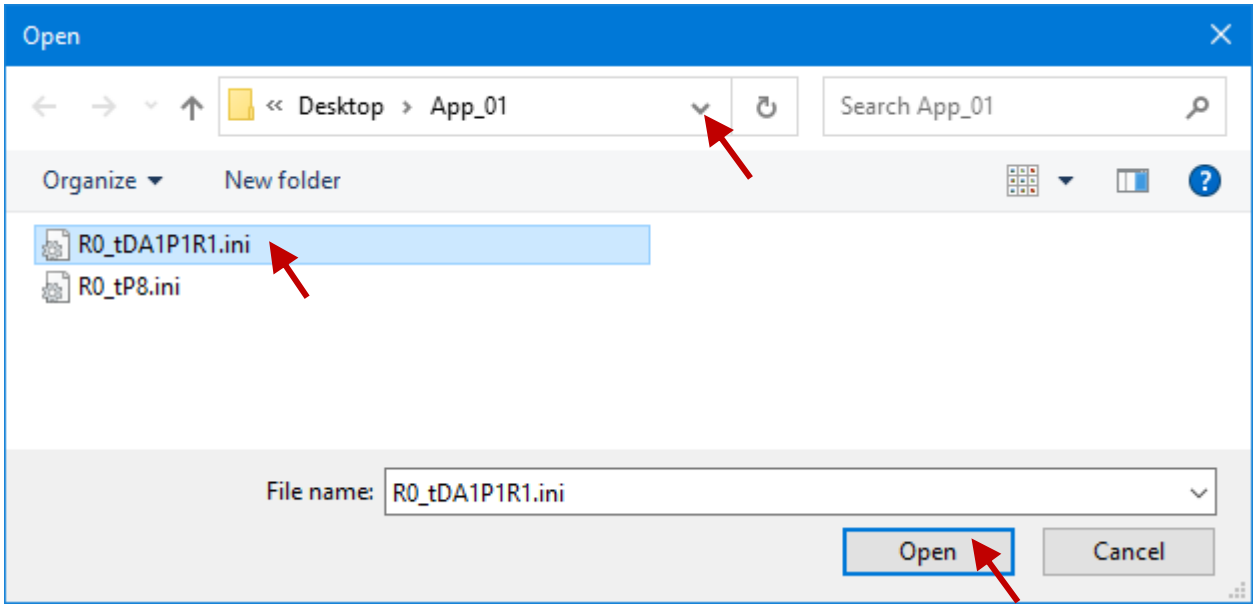
1. Click the **Remote I/O INIT* Quick Configuration** button in DCON Utility Pro on PC.



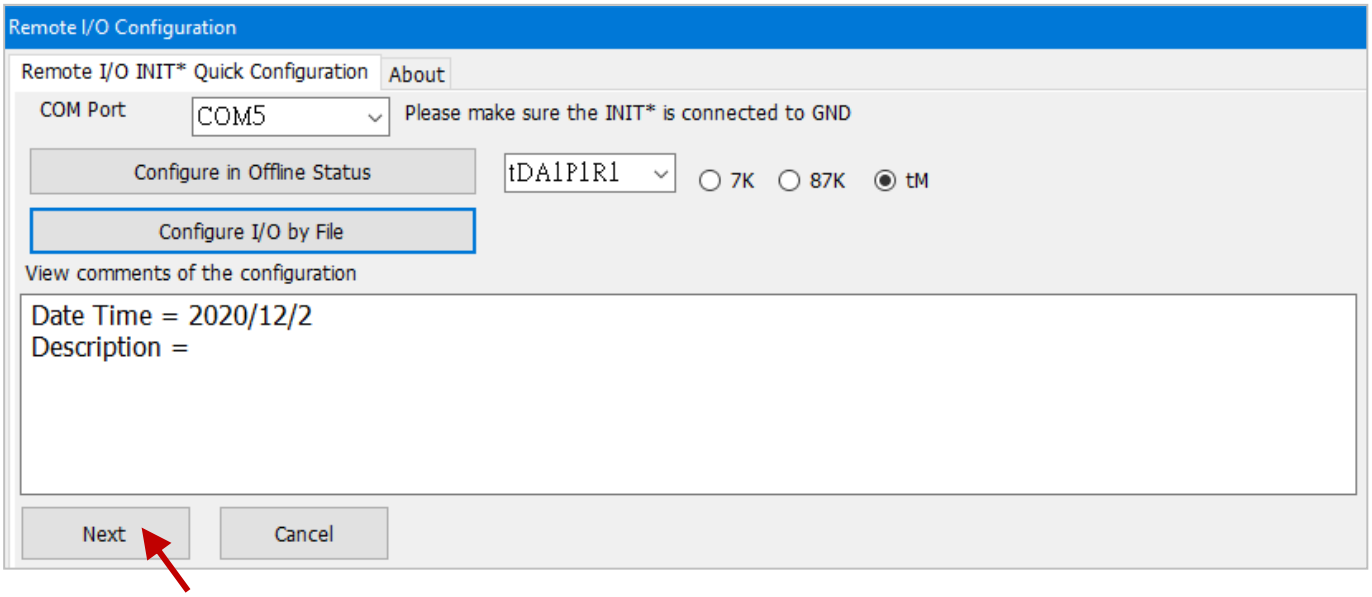
2. In the “Remote I/O Configuration” window, choose a COM Port and click the **Configure as INIT* Status** button.



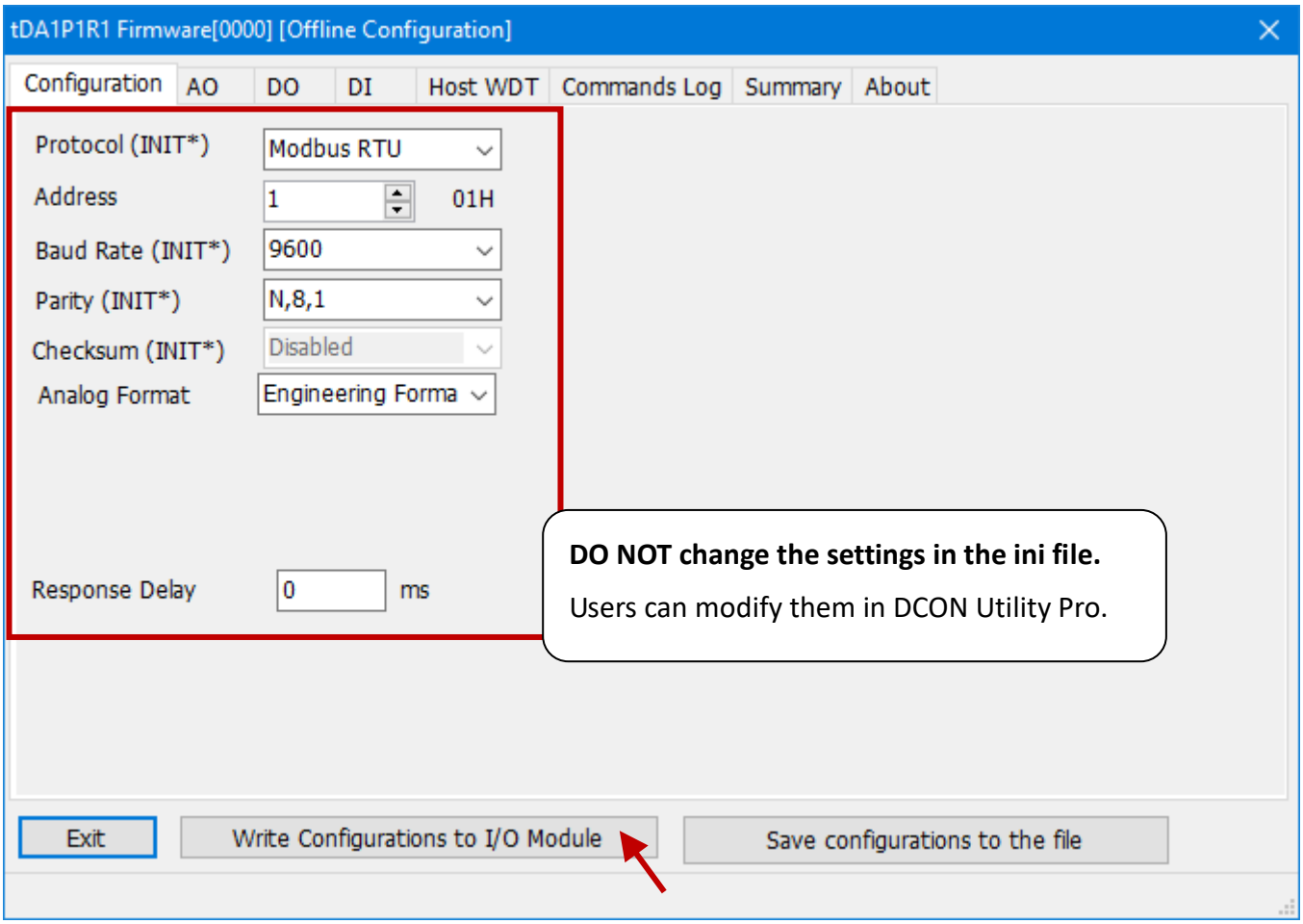
3. Select the configuration file (.ini) and click the **Open** button.



Click the **Next** button under the expanded text box.



4. Check or modify I/O settings and click the **Write Configurations to I/O Module** button to write configurations to the I/O module. After completing, adjust the DIP switch to the **Run** position and reboot the module to make new settings effective.



Appendix G Save or Load I/O Project

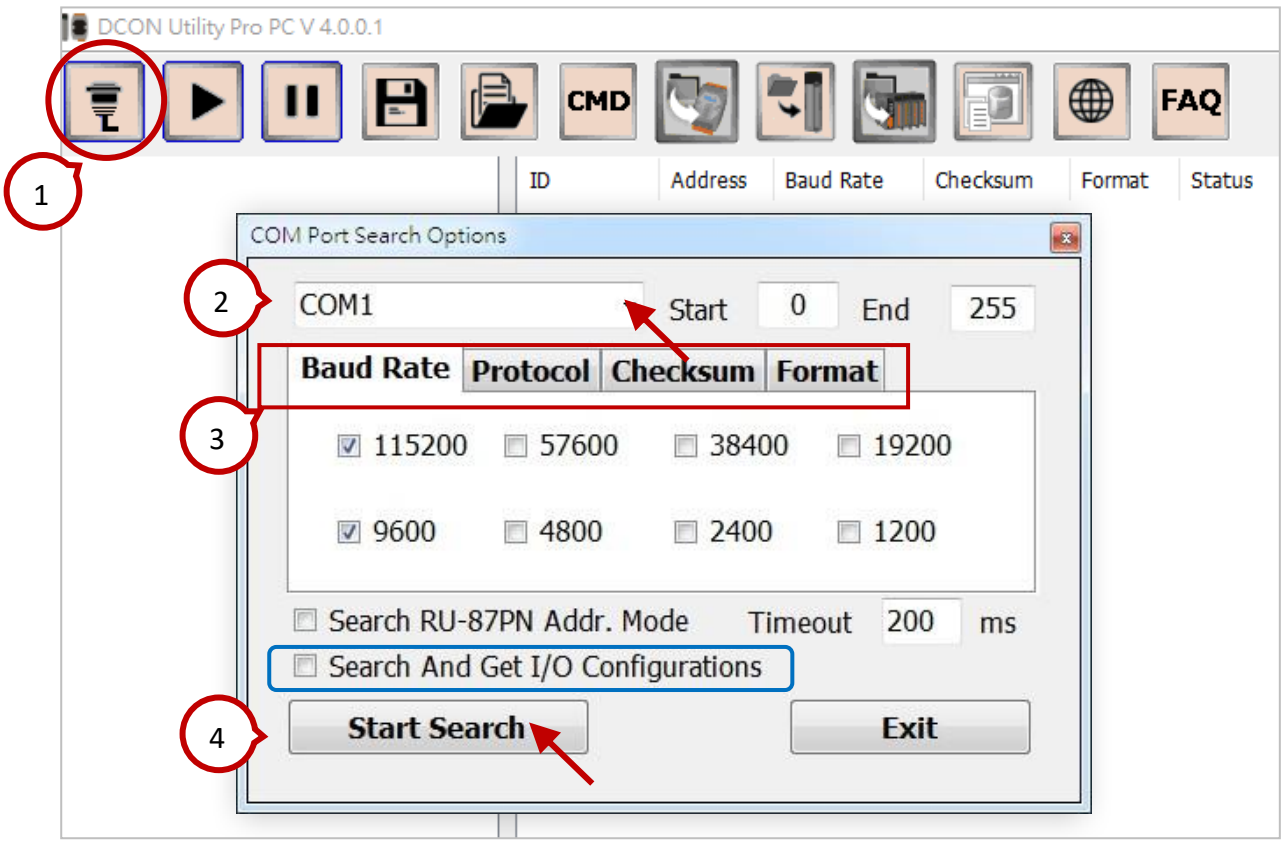
DCON Utility Pro v4.0 adds the **Save Project** and the **Load Project** functions. When many modules are required for a case, the user can search for the configured modules through different COM ports and save them as a project file, also encrypt the file by using any ZIP utility. Afterward, if the user needs to replace modules or build the same system, only load the project file to automatically compare the module settings, module names, and the COM port number to see if new modules meet the requirements of the original system for working normally.

G.1 Save Project

Follow the steps to save a project.

1. **Search I/O modules.**

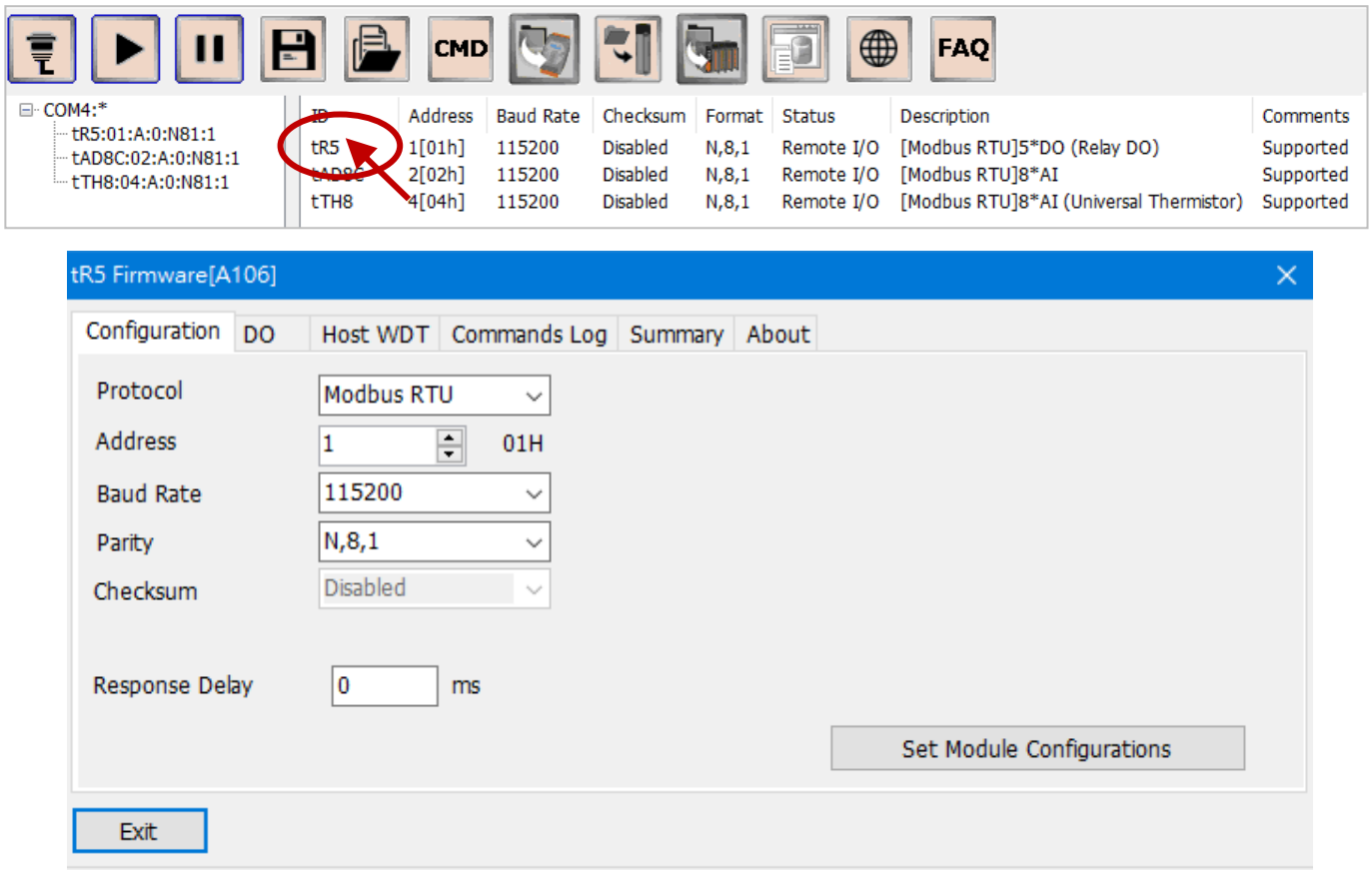
   Click the **COM Port** button and select a COM Port and conditions for searching, then click the **Start Search** button.



Note:

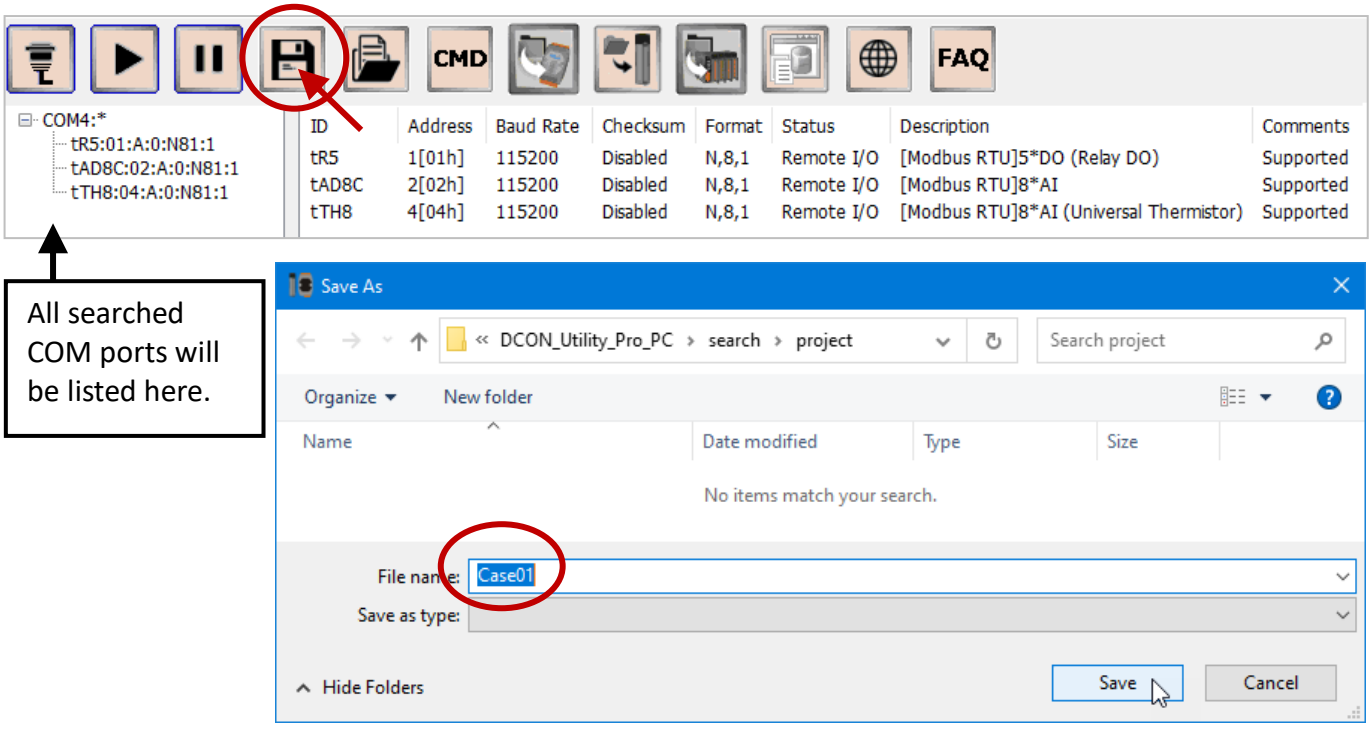
- Select the checkbox next to “Search and Get I/O Configurations” to search and load settings for modules. If NOT checked, also open the module configuration page to load settings.
- Using the same way to search again if there are some modules connected with other COM port.

If “**Search and Get I/O Configurations**” is not ticked, open the configuration page for specified modules to load settings before saving the project.

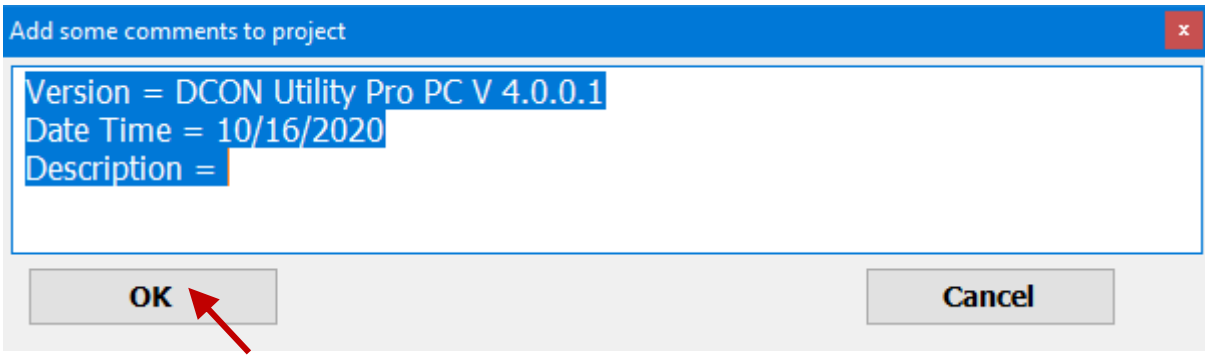


2. Save the I/O Project.

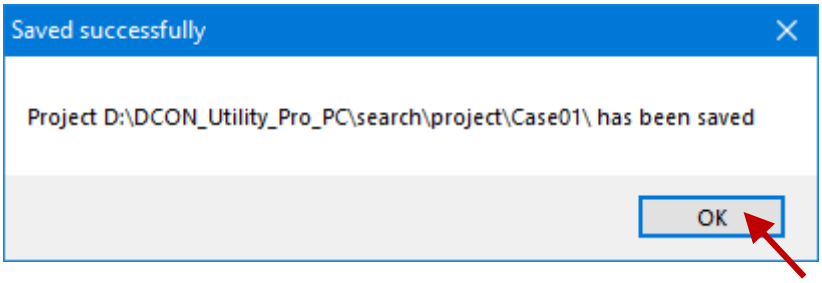
Click the **Save Project** button, enter a file name, and click the **Save** button to save all modules and configurations used in the system.



Next, the project comments window will be displayed. The user can add notes for the project and click OK.

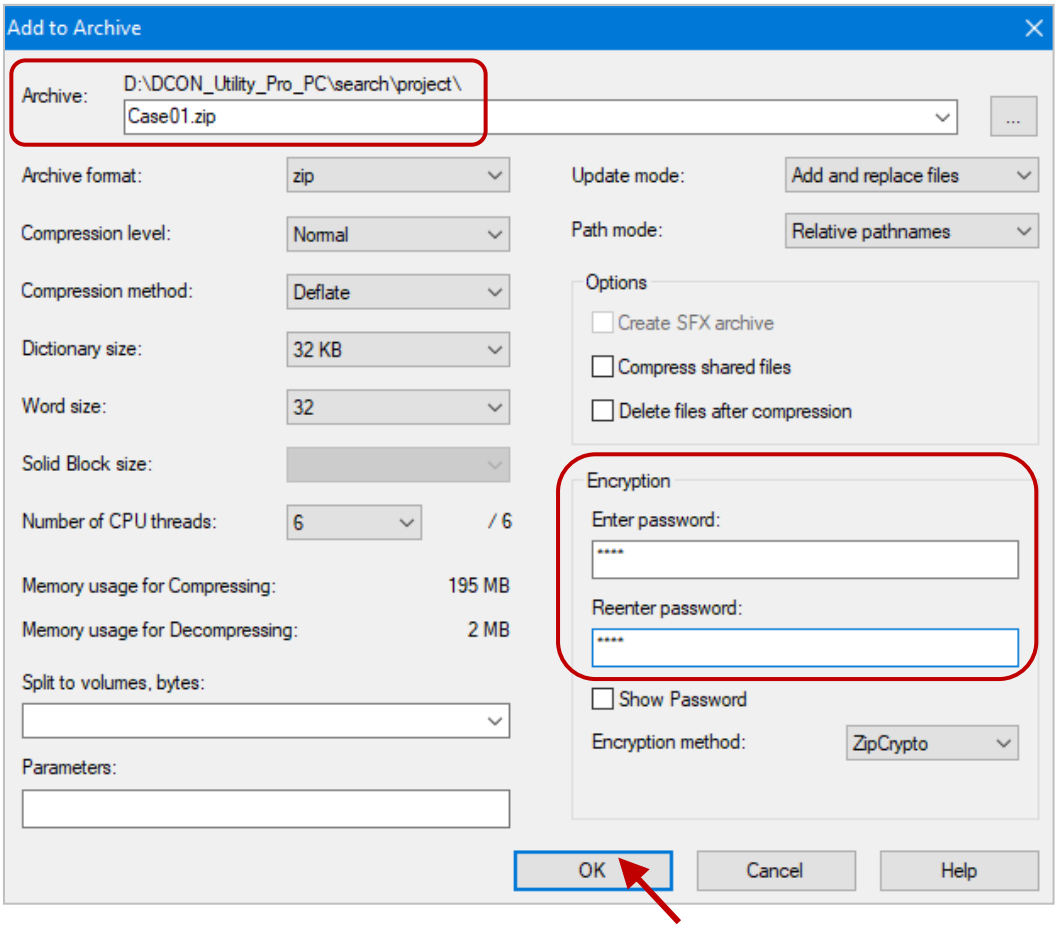


The dialog box will be displayed when saved successfully.



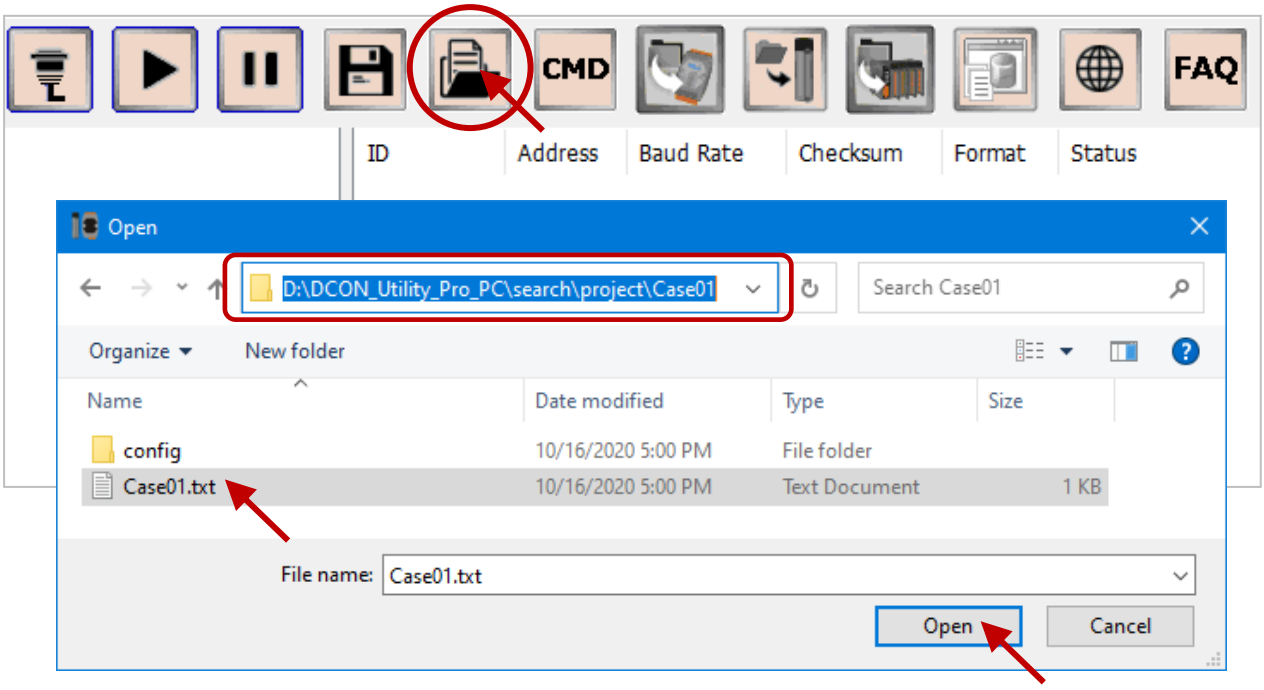
Notice:

The I/O project will be stored in the "..\DCON_Utility_Pro_PC\search\project" folder. It is recommended to back up the project on PC to avoid the user remove all older files when using a new version of DCON Utility Pro. Also, the project can be encrypted by using any ZIP utility.



G.2 Load Project

Click the **Load Project** button and open the project file (.txt) in the project folder (e.g., Case01).



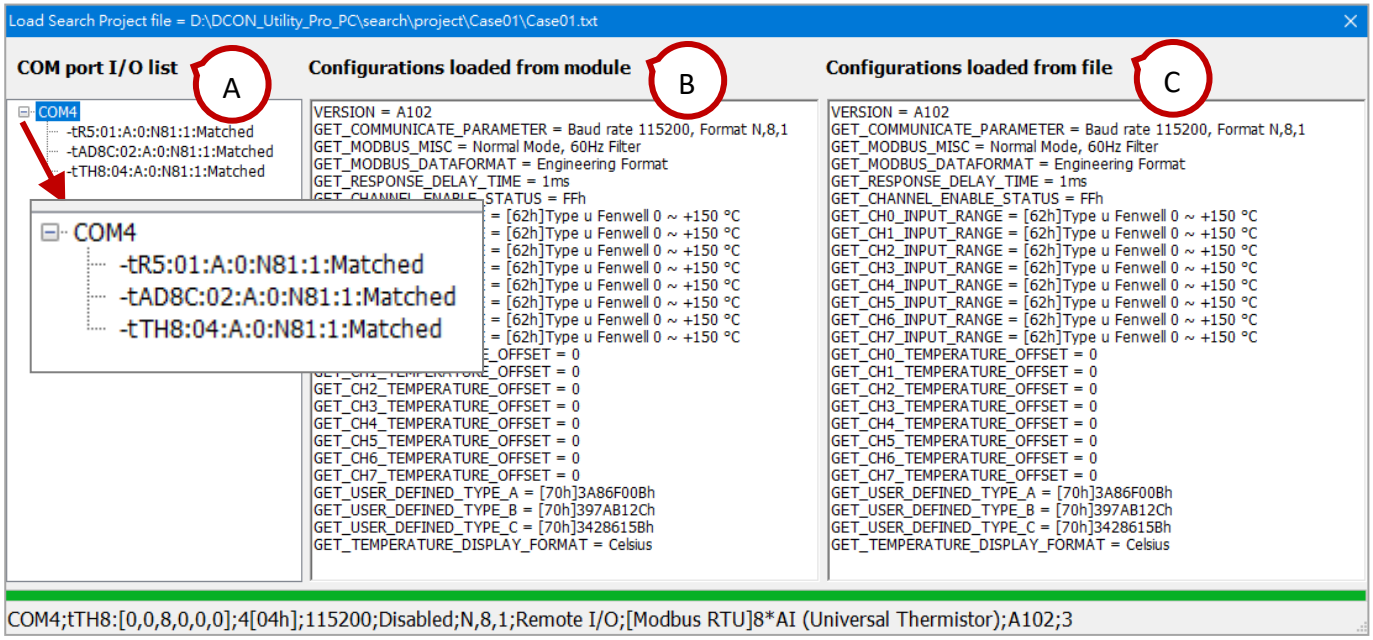
Note: After clicking the Open button, the project comments window will be displayed. Refer to the previous page.

If the PC is connected with modules and the settings are correct, you can see the following window after loading the I/O project.

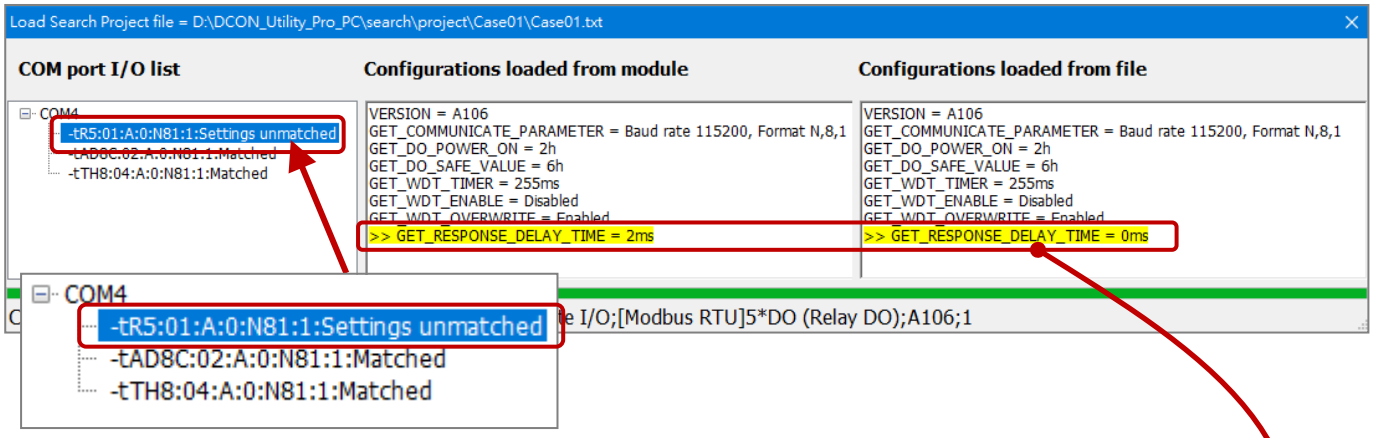
A. **COM Port I/O list**: Displays the COM port number and module names loaded from the project file, also comparison status for modules. The user can click each item to check the settings.

B. **Configurations Loaded from a module**: Displays the current module settings.

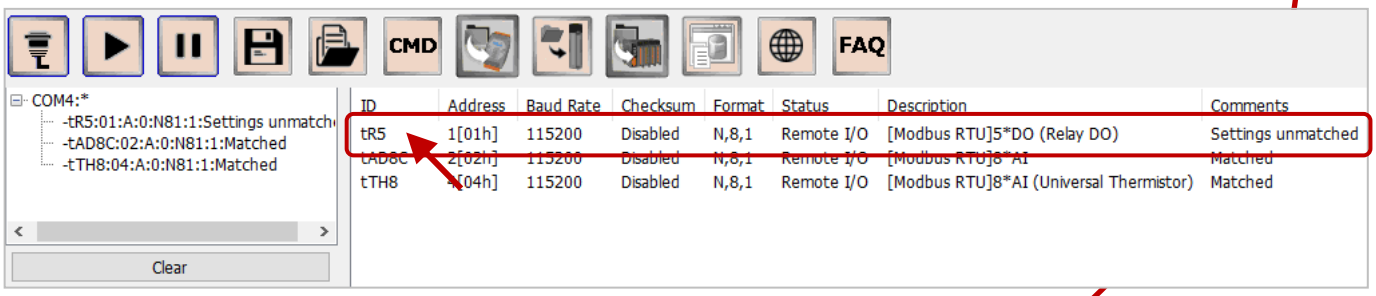
C. **Configurations Loaded from file**: Displays the module settings loaded from the project file.

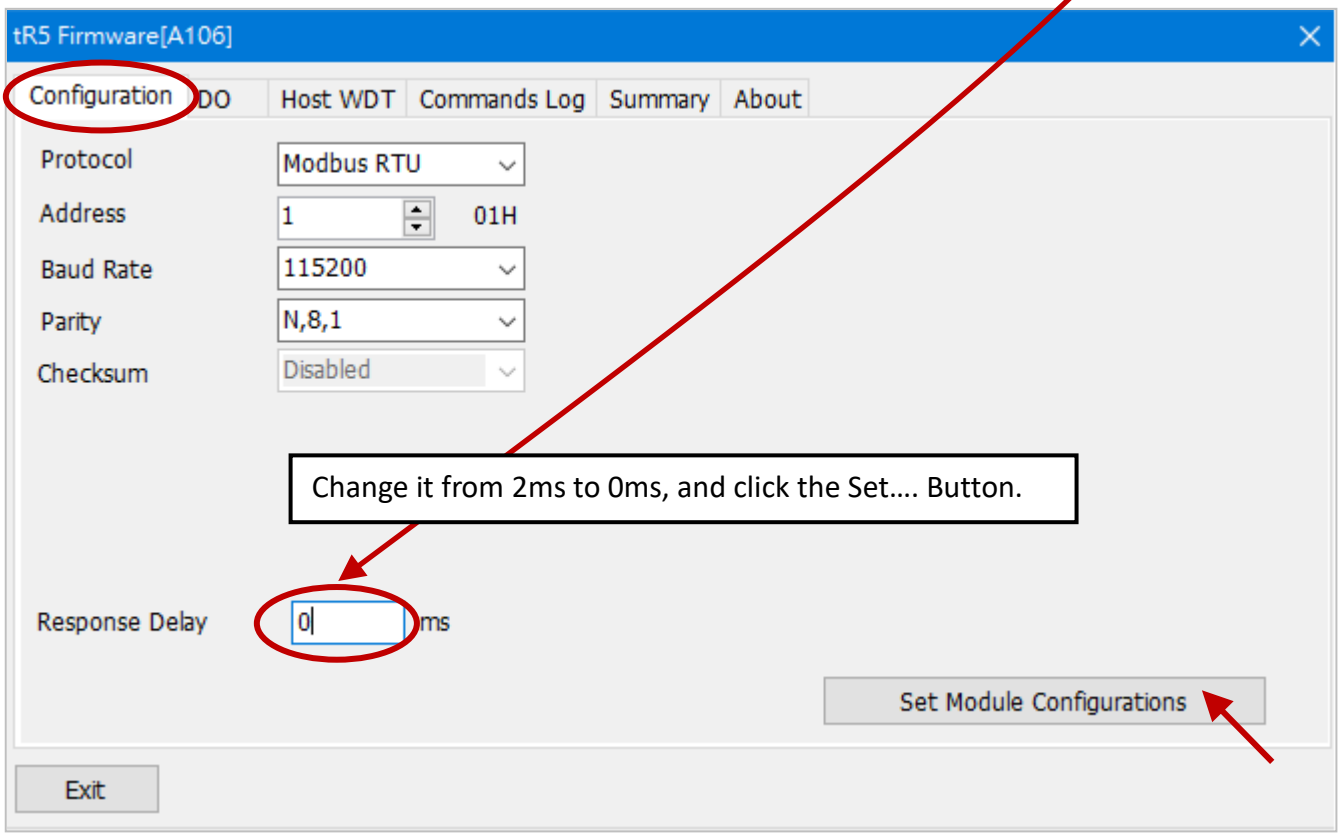


If the PC is connected with modules but the settings are incorrect, the “Settings unmatched” text will be displayed next to the module name and the error will be marked after loading the I/O project. The user can close the window and modify the settings.

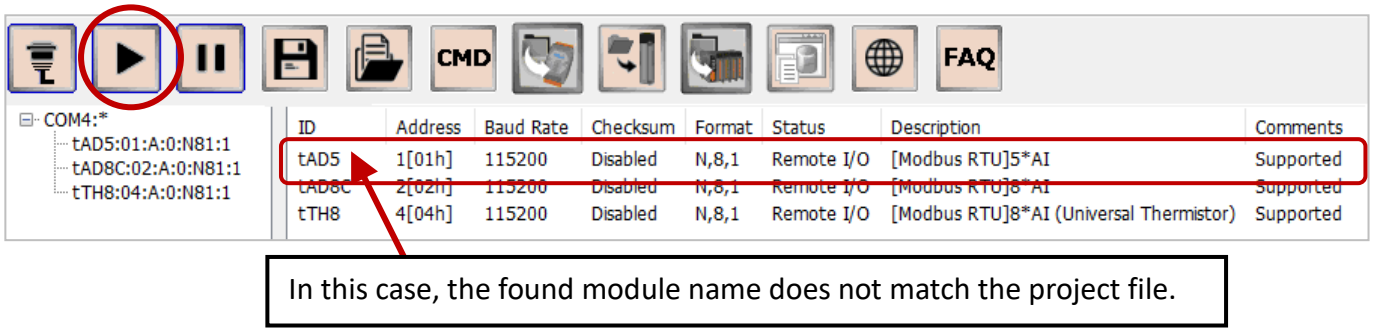


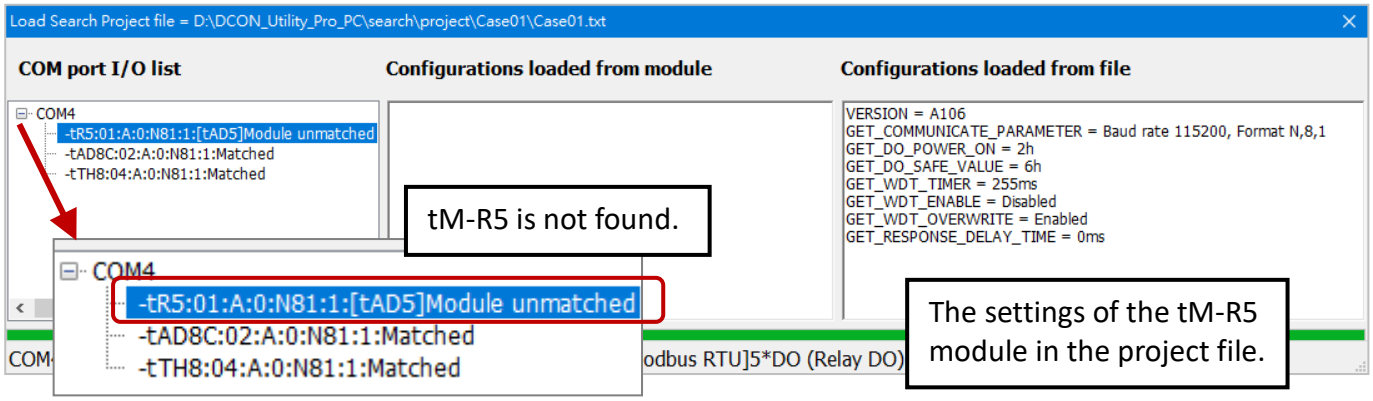
Change the settings to be the same with the project file, in this case, set the Response Delay to 0 ms.



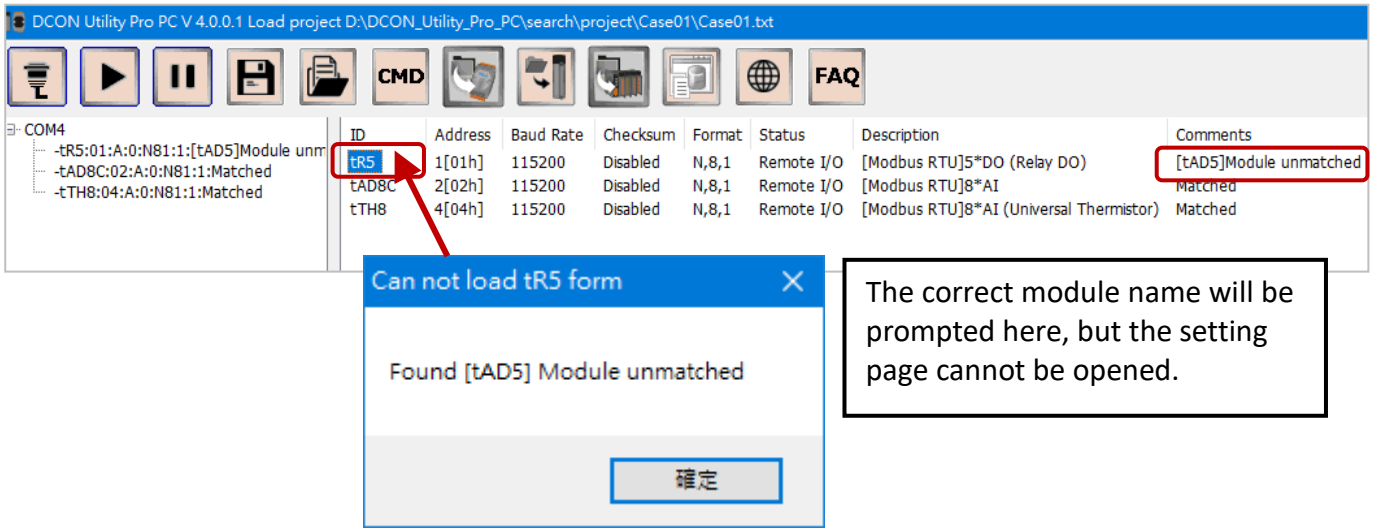


If the PC is connected with modules but the communication settings or module names are incorrect, the "Module not found" or "Module unmatched" text will be displayed next to the module name and only the loaded module settings can be reviewed.

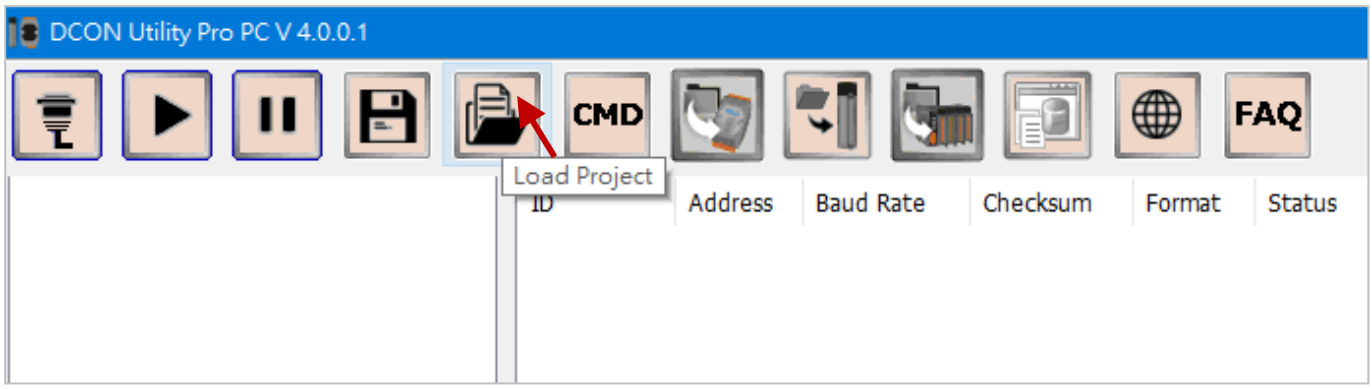




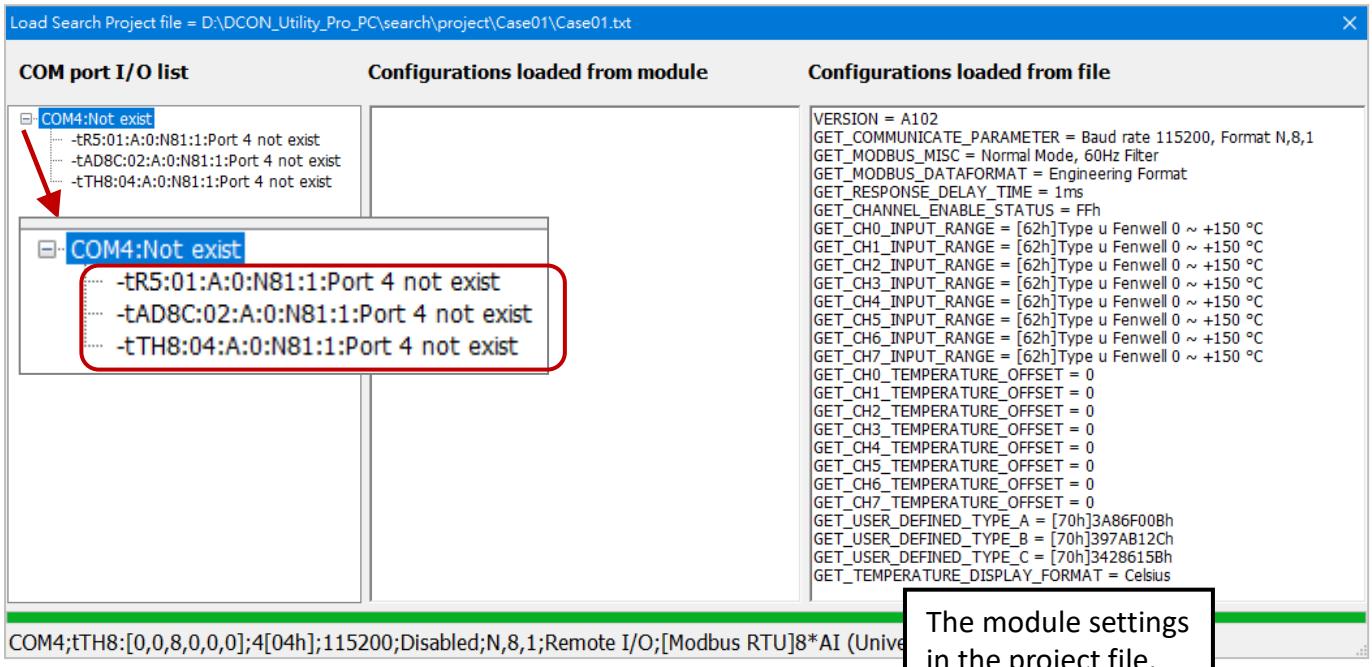
After closing the "**Load**…" window, **the user can see the correct module name but cannot open the setting page.** Please replace the correct module and set the communication parameters properly, then execute the **Load Project** to do the comparison.



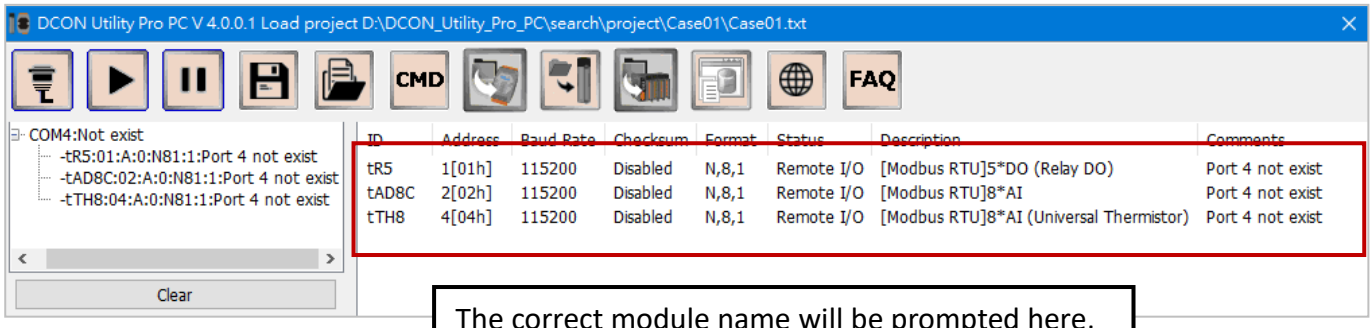
If the PC is not connected with the I/O module, the “Not exist” text will be displayed next to the module name and only the loaded module settings can be reviewed.



Also, click each module name to view the settings.



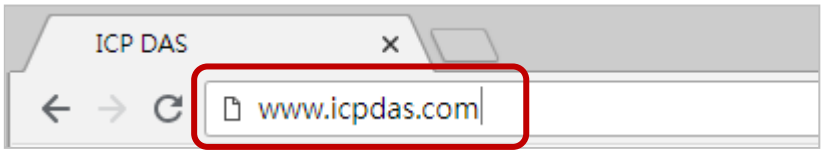
After closing the “**Load**…” window, **the user can see the correct module names.** Please connect modules and set the communication parameters properly, then execute the **Load Project** to do the comparison.

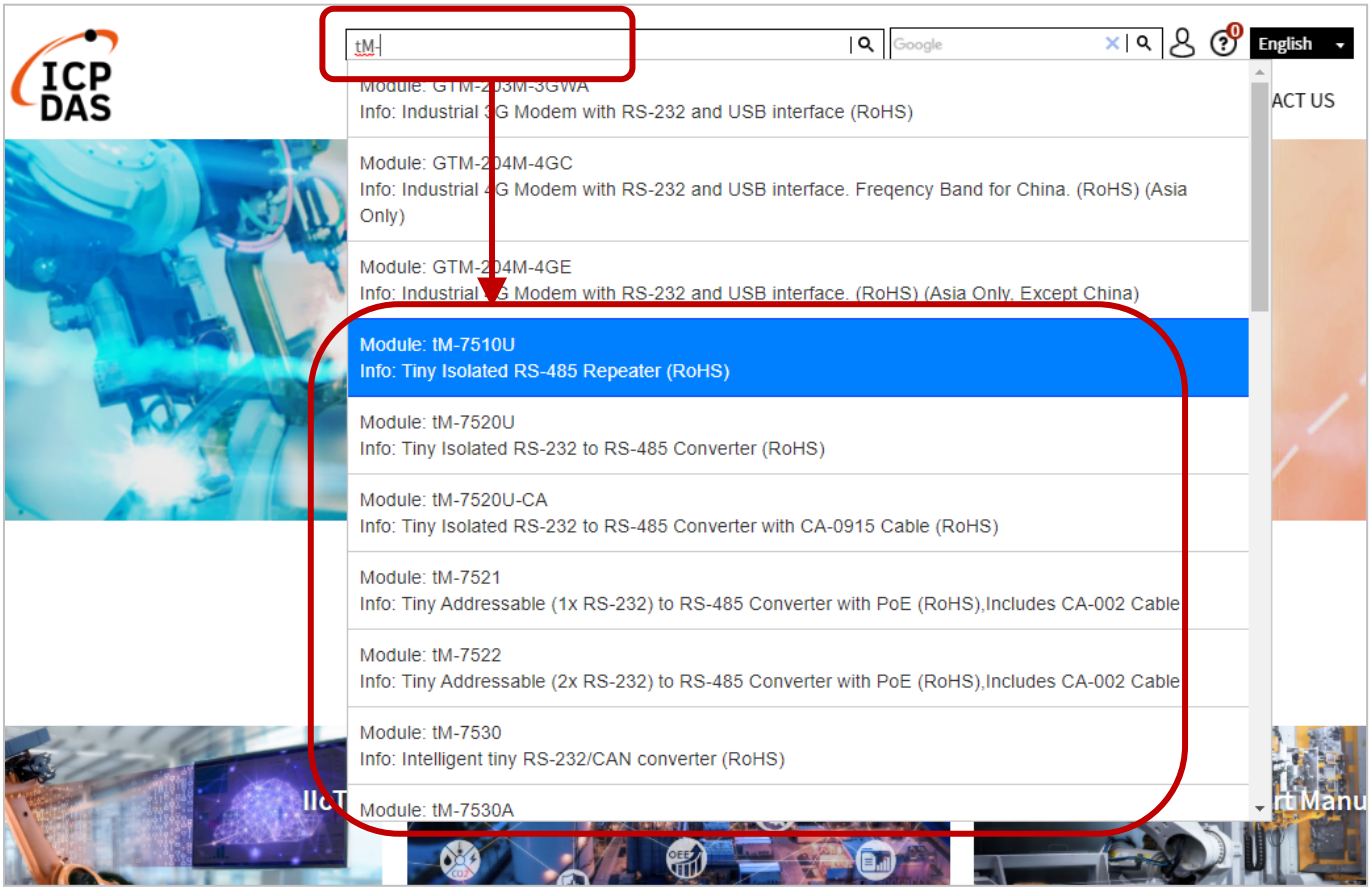


Appendix H Other Information

H.1 How do I Search the Product Webpage?

Enter the keyword in the Search box on the ICP DAS website (www.icpdas.com) to find out the product information.

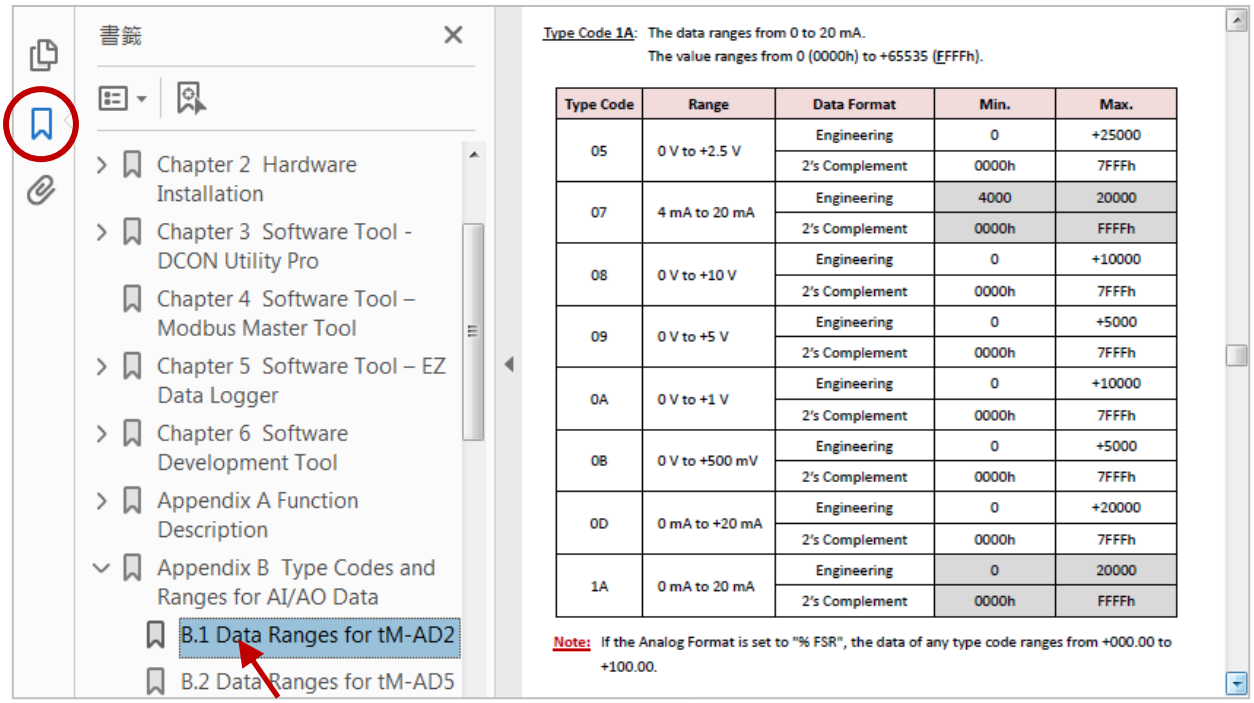




H.2 How do I Quickly Navigate a Specific Chapter?

1) Read the PDF file on PC

Click the Bookmark icon on the left-side panel, and then click any heading with the chapter number.



2) Read the PDF file in a browser

Depending on the browser, find out the Bookmark icon at the top of the PDF file or just click on the page to show the toolbar.