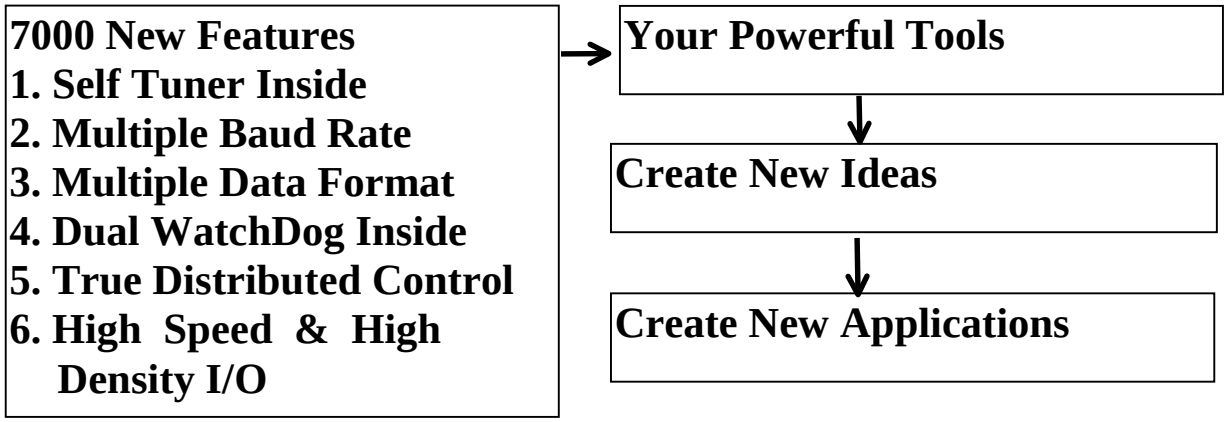
7017, 7018, 7013,7013D

User Manual



Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assume no liability for damages consequent to 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, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

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

Table of Contents

1. Introduction 4
   1.1 More Information 4
   1.2 Pin Assignment 5
   1.3 Specifications 7
   1.4 Block Diagram 10
   1.5 Application Wiring 11
   1.6 Quick Start 13
      1.6.1 Voltage Input Measurement 14
      1.6.2 Current Input Measurement 15
      1.6.3 Thermocouple Measurement 15
      1.6.4 RTD Measurement 16
   1.7 Default Setting 17
   1.8 Calibration 17
      1.8.1 7017 Calibration 19
      1.8.2 7018 Calibration 21
      1.8.3 7013 Calibration 22
   1.9 Tables 23

2. Command Set 29
   2.1 %AANNTTCCFF 34
   2.2 #** 35
   2.3 #AA 36
   2.4 #AAN 37
   2.5 $AA0 38
   2.6 $AA1 39
   2.7 $AA2 40
   2.8 $AA3 41
   2.9 $AA4 42
   2.10 $AA5VV 43
   2.11 $AA6 44
   2.12 $AA8V 45
   2.13 $AA9SCCCC 46
   2.14 $AA9S(data) 47
   2.15 $AAA 48
   2.16 $AAF 49

2.17 $AAM 50
2.18 ~** 51
2.19 ~AA0 52
2.20 ~AA1 53
2.21 ~AA2 54
2.22 ~AA3ETT 55
2.23 ~AAO(name) 56

3. Operations Principle & Application Notes 57
3.1 INIT*_pin Operation Principle 57
3.2 Dual WatchDog Operation Principle 58
3.3 Analog Data Format 59
3.4 Temperature Measurement 60
3.5 CJC Offset Calibration 61
3.6 Command Response Time 62
3.7 Response Time of A/D 63

1. Introduction

7000 is a family of network data acquisition and control modules. They provide A/D, D/A, DI/O, Timer/Counter and other functions. These modules can be remote controlled by a set of commands. The common features of A/D are given as following:

- 24 bits sigma-delta A/D converter to provide 16 bits precision.
- Input range is programmable.
- Software Calibration

The 7017 is a 8 channels analog input module. The 7018 is a 8 channels thermocouple input module. The 7013 is a single channel RTD input module. The 7013D is a 7013 with 4½ digit LED display.

The special features of 7017 are giving as following:

- Accept voltage or current input

The special features of 7018 are giving as following:

- Accept voltage or current input or J/K/T/E/R/S/B/N/C thermocouple

The special features of 7013 are giving as following:

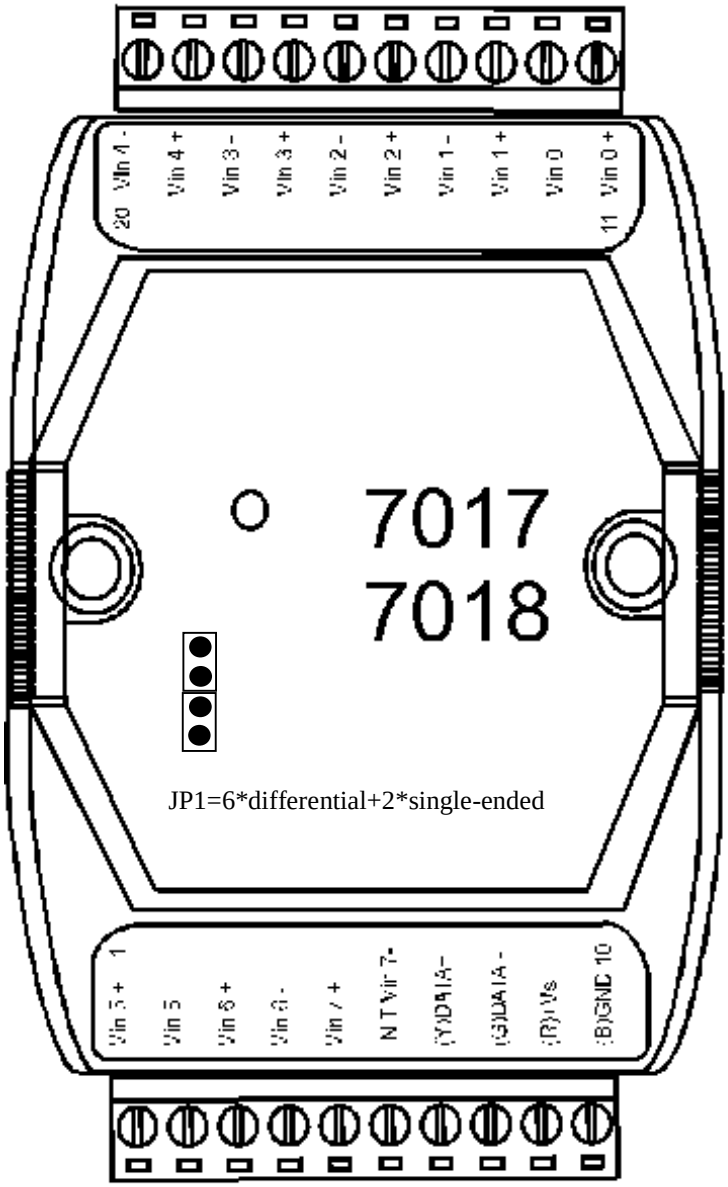
- Support 2-wire, 3-wire, 4-wire RTD input connection
- Accept platinum, nickel RTD
- Built-in excitation current source

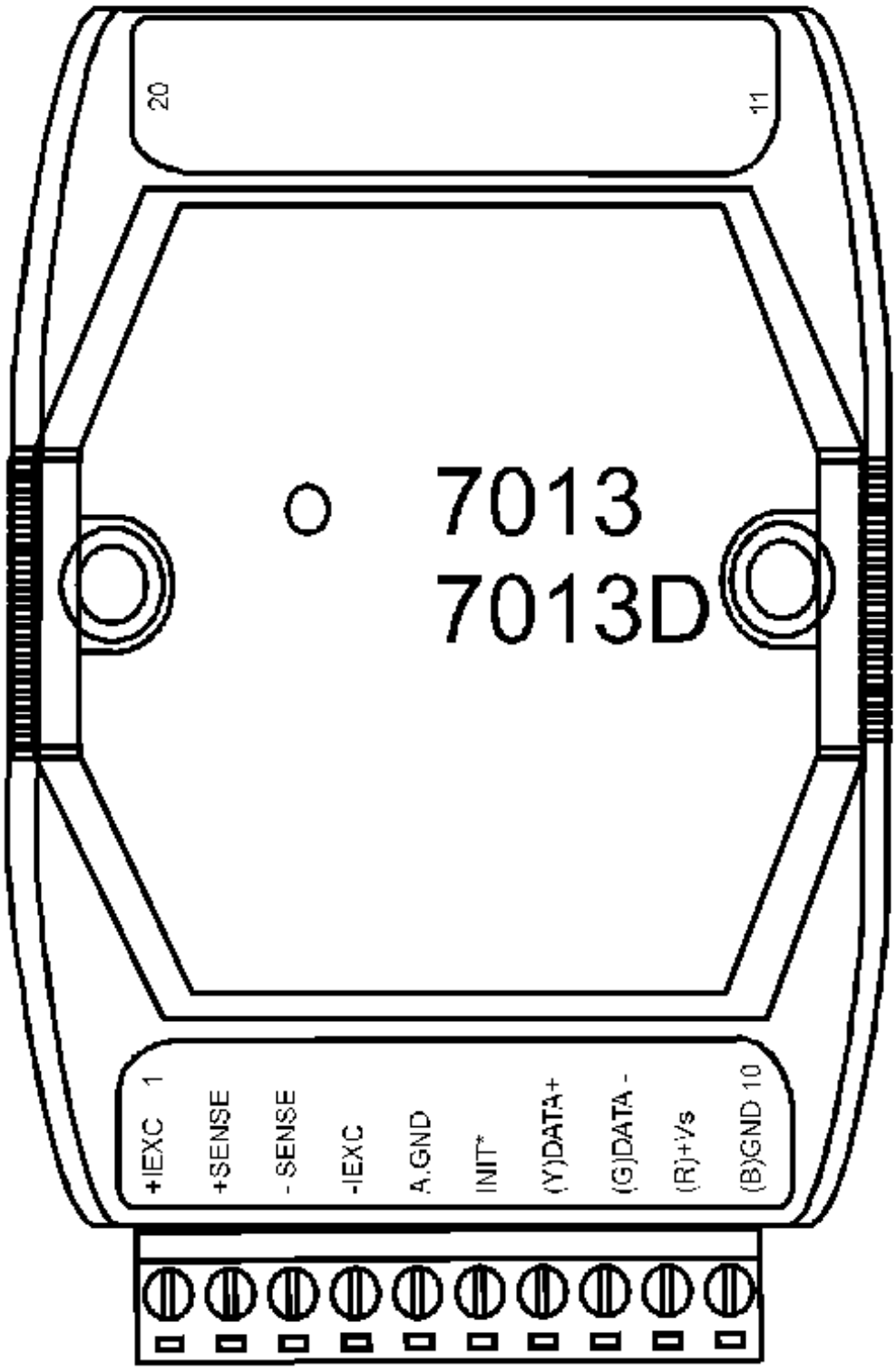
1.1 More Information

Refer to . 7000 Bus Converter User Manual. chapter 1 for more information as following:

1.1 7000 Overview
1.2 7000 Related Documentation
1.3 7000 Common Features
1.4 7000 System Network Configuration
1.5 7000 Dimension

1.2 Pin Assignment




20
11
7013
7013D
+IEXC 1
+SENSE
- SENSE
-IEXC
A.GND
INIT*
(Y)DATA+
(G)DATA -
(R)+Vs
(B)GND 10

1.3 Specifications

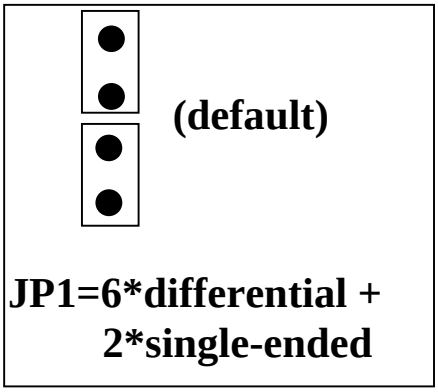
7017: 8 Channel Analog Input Module

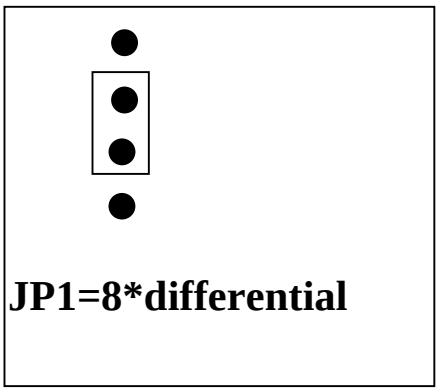
Analog Input

- Channels: 6 differential + 2 single-ended or 8 differential (selected by JP1)
- Input type: mV, V , mA
- Input range: ±150mV, ±500mV, ±1V, ±5V, ± 10V and ±20mA(with external resistor)
- Sample rate: 10 sample/sec(total)
- Bandwidth: 13.1 Hz
- Accuracy: ±0.1% or better
- Zero drift: ±0.03uV/°C
- Span drift: ±25ppm/°C
- CMR @ 50/60 Hz : 92 dB min
- Over voltage protection: ±35V

Power:

- Power consumption: 2W





7018: 8-Channel Thermocouple Input Module

Analog Input

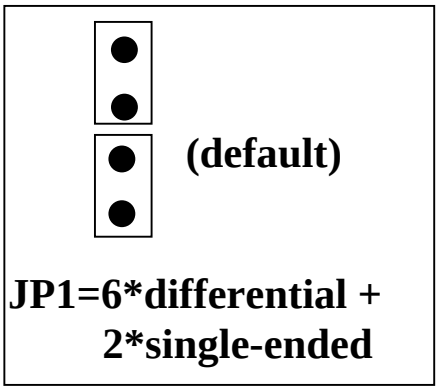
- Type: thermocouple, mV, V , or mA
- Channels: 6 differential + 2 single-ended or 8 differential(jumper select)
- Thermocouple type:

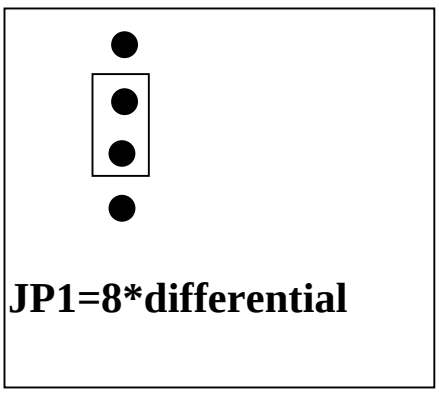
Type	Range	Type	Range
J	0°C~760°C	S	500°C~~1750°C
K	0°C~1000°C	B	500°C~~1800°
T	-100°C~+400°C	N	-270°C~1300°C
E	0°C~1000°C	C	0°C~2320°C
R	500°C~1750°C		

- Voltage range: ±15mV, ±50mV, ±100mV, ±500mV, ±1V, ±2.5V
- Current range: ±20mA (with external resistor)
- Sampling rate: 10 samples/sec(total)
- Bandwidth: 13.1 Hz
- Accuracy: ±0.05% or better
- Zero drift: ±0.033ppm/°C
- CMR @ 50/60 Hz: 150 dB
- NMR @50/60 Hz: 100 dB
- Span drift: 25ppm/°C
- Over voltage protection: ±35V

Power

- Power consumption: 2W





7013: Single Channel RTD Input Module
7013D: 7013 with display

Analog Input

- Channel: 1
- Input Type: Pt , Ni
- RTD temperature range

Pt100	-100°C	to	+100°C	α=0.00385
Pt100	0°C	to	+100°C	α=0.00385
Pt100	0°C	to	+200°C	α=0.00385
Pt100	0°C	to	+600°C	α=0.00385
Pt100	-100°C	to	+100°C	α=0.003916
Pt100	0°C	to	+100°C	α=0.003916
Pt100	0°C	to	+200°C	α=0.003916
Pt100	0°C	to	+600°C	α=0.003916
Ni120	-80°C	to	+100°C	
Ni120	0°C	to	+100°C	

- Sampling rate : 10 samples/sec
- Bandwidth : 4 Hz
- Wire connection: 2/3/4 wire
- Accuracy: ±0.05% or better
- Zero drift :±0.3uV/°C
- CMR @ 50/60 Hz: 92 dB min
- NMR @50/60 Hz: 100 dB
- Span drift :±25ppm/°C

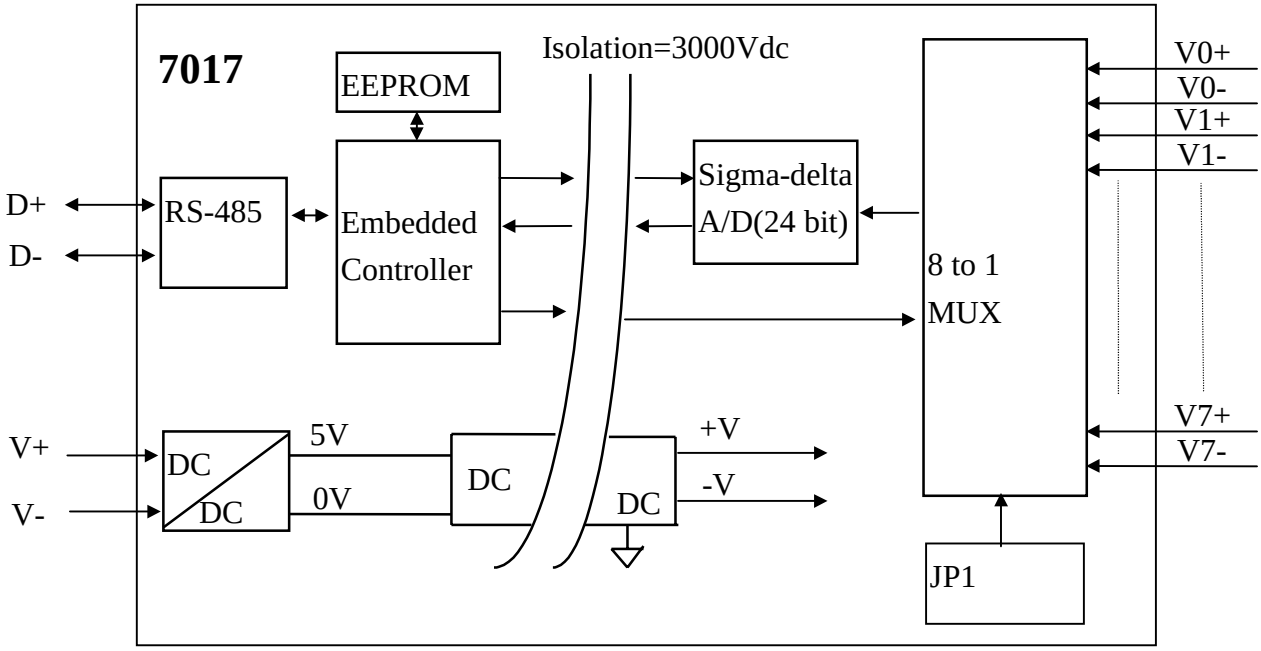
Display:

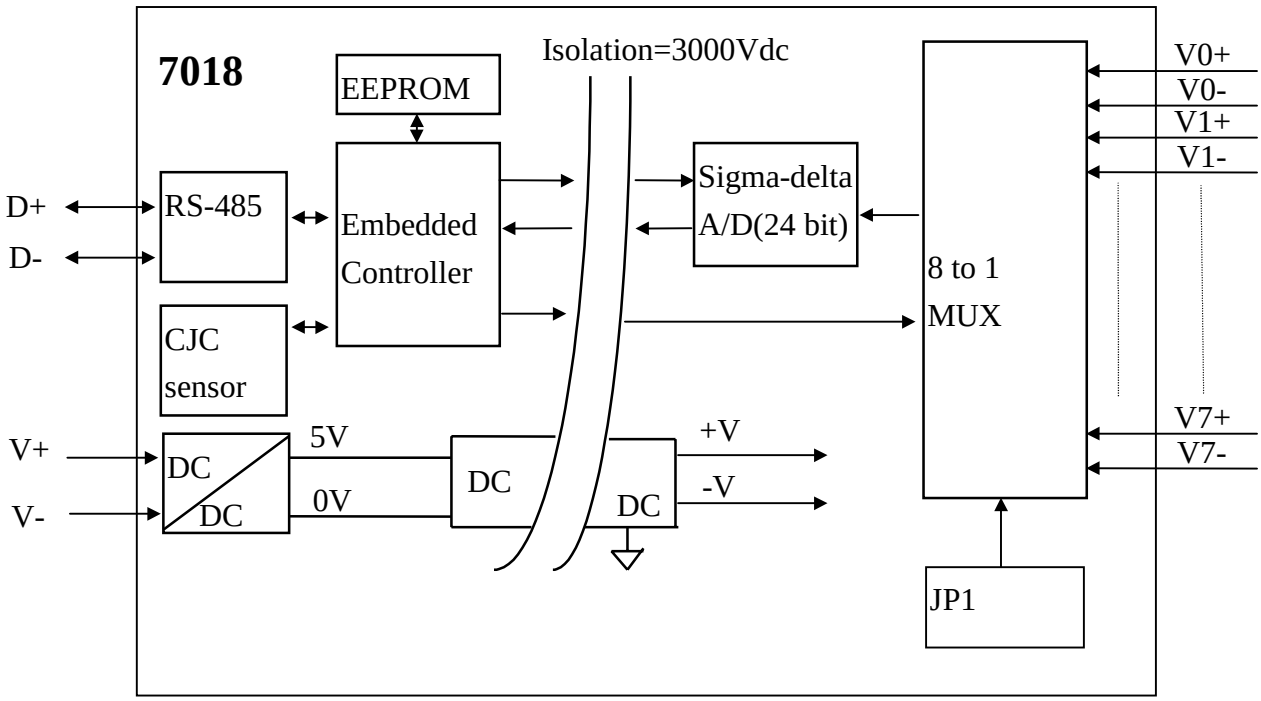
- LED: 4½ digit(7013D)

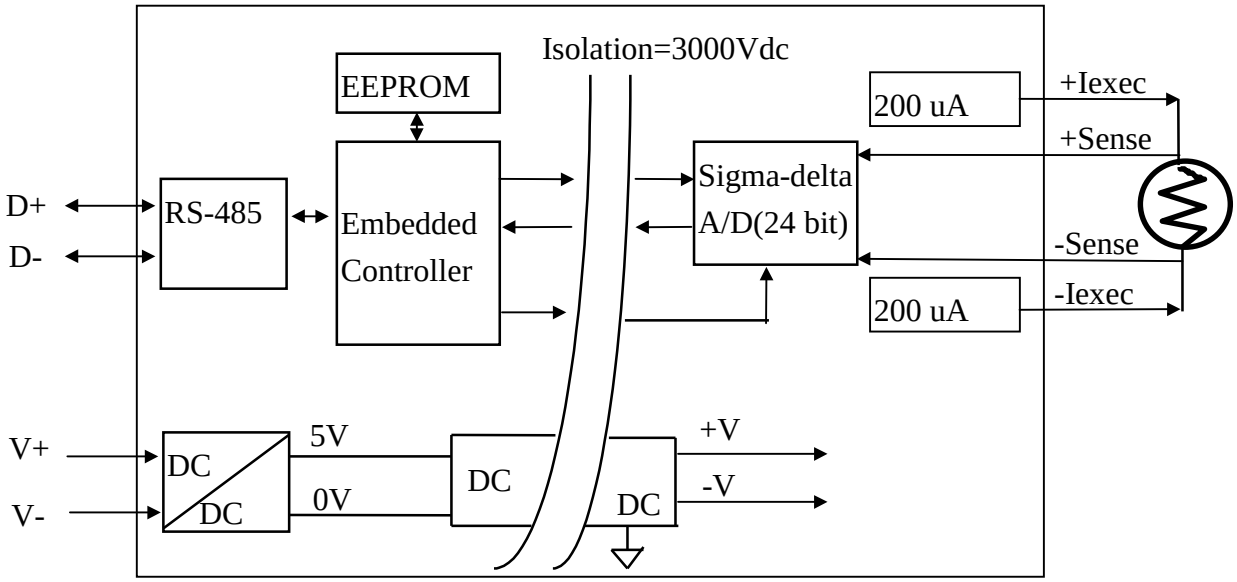
Power consumption:

- 2W for 7013
- 2.2W for 7013D

1.4 Block Diagram

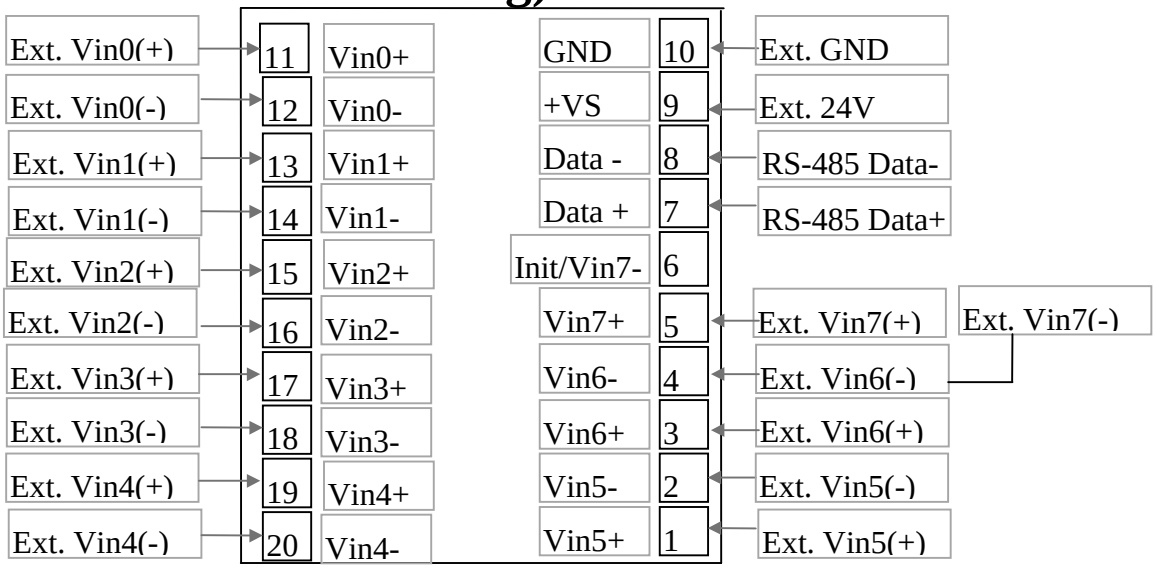






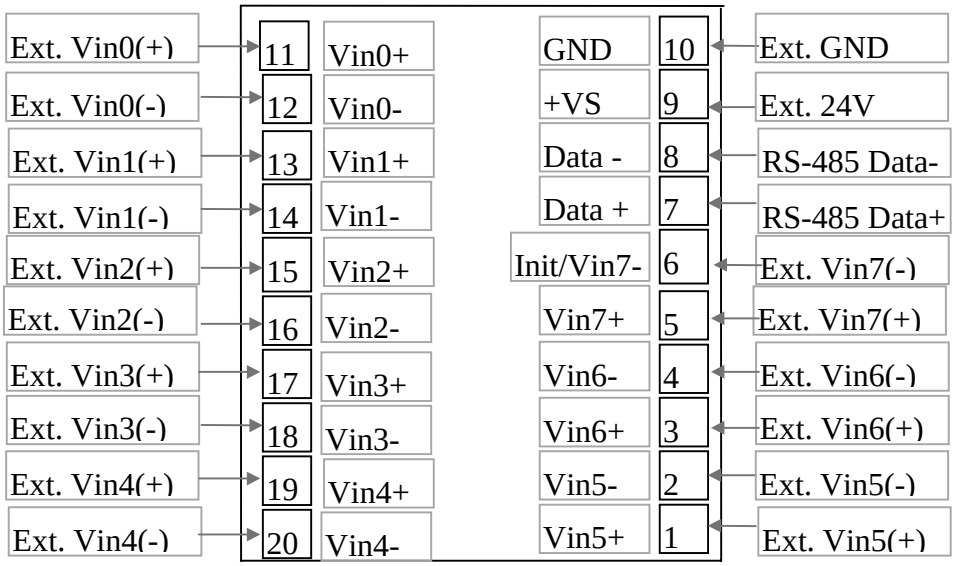
1.5 Application Wiring

JP1 select 6*differential + 2*single-ended (refer to Sec. 1.3 for JP1 setting)



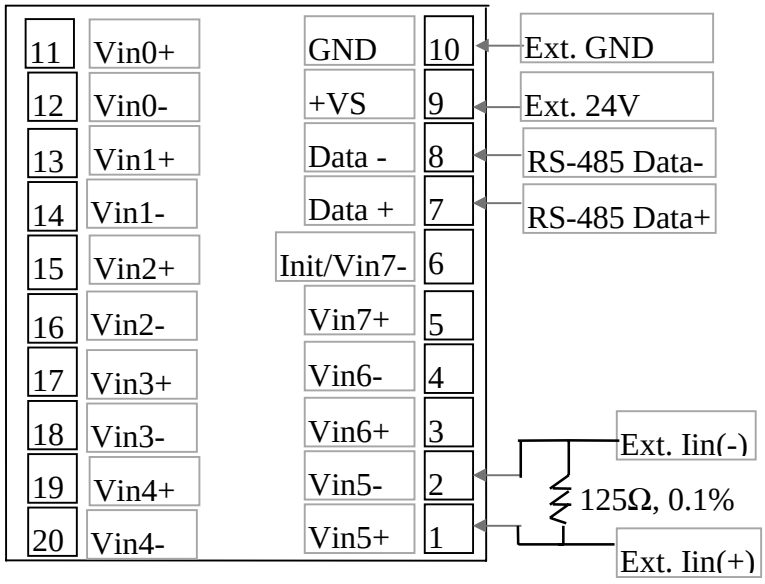
7017 & 7018

JP1 is used to select 8*differential (refer to Sec. 1.3 for JP1 setting)

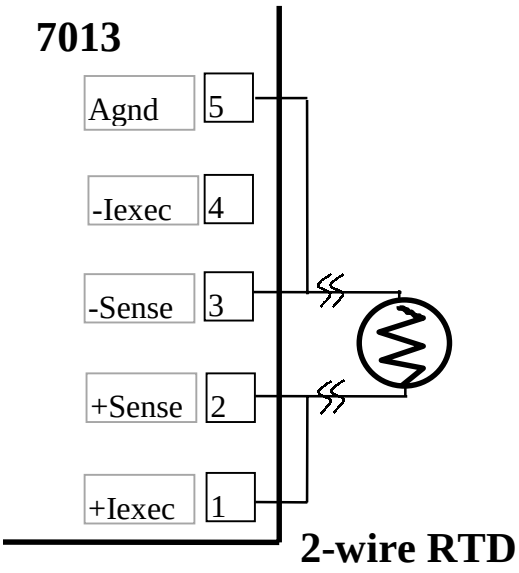


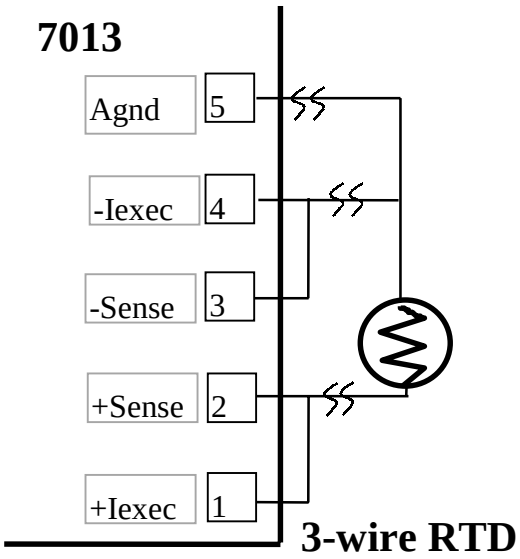
7017 & 7018

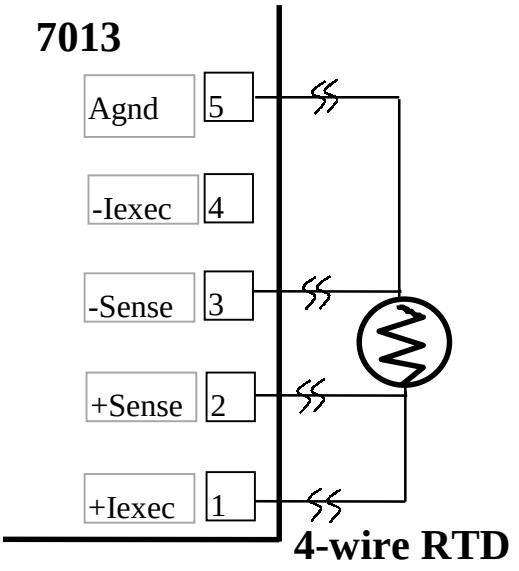
Current Measurement

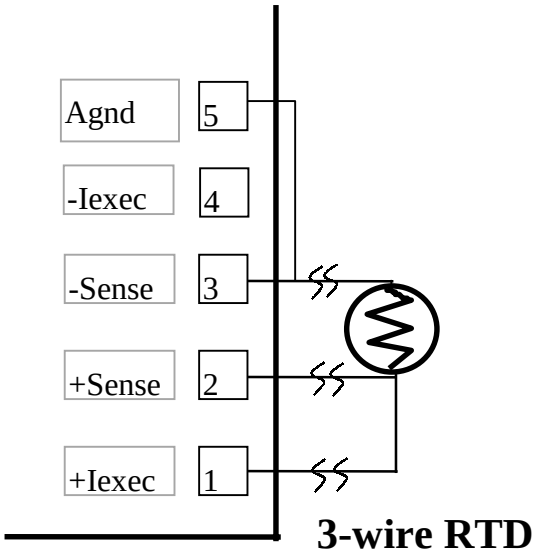


7017 & 7018









1.6 Quick Start

Refer to ．7000 Bus Converter User Manual． chapter-5 for the following functions:

- **module status unknown**(Sec. 5.1), **change address**(Sec. 5.2)
- **change baud rate**(Sec. 5.3), **checksum enable/disable**(Sec. 5.4)
- Diagnostic program **TEST.EXE**(Refer to ．NAP7000S User Manual． for details

1.6.1 Voltage Input Measurement

1. Refer to Sec. 1.5 for wire connection. Power on and run **test.exe**
2. press **2**
3. press **$012[Enter]** → Receive=!01080600
4. press **2**
5. press **#015[Enter]** → Receive=>+??.???
6. press **2**
7. press **%0101080602[Enter]**→ Receive=!01
8. press **2**
9. press **#015[Enter]** → Receive=>????
10. press **2**
11. press **%0101090600[Enter]**→ Receive=!01
12. press **2**
13. press **#015[Enter]** → Receive=+?.????

- step 3: read the status of 7017 is +/- 10V, engineering unit format
- step 5: show the channel_5 external voltage in engineering unit format
- step 7: change from engineering unit format to hex format
- step 9: show the channel_5 external voltage in hex format
- step 11: change status to +/- 5V, engineering unit
- step 13: show the channel_5 external voltage (+/- 5V range)

1.6.2 Current Input Measurement

1. Refer to Sec. 1.5 for wire connection, power on and run **test.exe**
2. press **2**
3. press **$012[Enter]** → Receive=!01080600
4. press **2**
5. press **%01010D0600[Enter]**→ Receive=!01
6. press **2**
7. press **#015[Enter]** → Receive=>+??.???

- step 3 : read the status of 7017
- step 5 : change status to +/- 20mA, engineering unit format
- step 7 : show the channel-5 external current in engineering unit format

1.6.3 Thermocouple Measurement

1. Refer to Sec. 1.5 for wire connection, power on and run **test.exe**
2. press **2**
3. press **$012[Enter]** → Receive=!01050600
4. press **2**
5. press **%01010E0600[Enter]**→ Receive=!01
6. press **2**
7. press **#015[Enter]** → Receive=>+???.??

- step 3 : read the status of 7018
- step 5 : change to J-type thermocouple, engineering unit format
- step 7 : show the channel-5 temperature value in engineering unit format

1.6.4 RTD Measurement

1. Refer to Sec. 1.5 for wire connection, power on and run **test.exe**
2. press **2**
3. press **$012[Enter]** → Receive=!01200600
4. press **2**
5. press **#01[Enter]** → Receive=>+???.??
6. press **2**
7. press **%0101200602[Enter]**→ Receive=!01
8. press **2**
9. press **#01[Enter]** → Receive=>????

- step 3 : read the status of 7013 is Platinum RTD, ±100°C, α=0.00385, engineering unit format
- step 5 : show the temperature in engineering unit format
- step 7 : change from engineering unit format to hex format
- step 9 : show the temperature in hex format

1.7 Default Setting

The default setting is given as following:

- address=01, baud rate=9600, checksum disable
- type=08=±10V input range (for 7017)
- type=05=±2.5V input range (for 7018)
- type=20=platinum, ±100°C(for 7013)
- data=1 start+8 data+1 stop(no parity)

JP1 of 7017 & 7018 is used to select 6*differential + 2*ingle-ended or 8*differential. Refer to Sec. 1.3 for JP1 setting.

The 7017 & 7018 can be configured to **8*differential.** Refer to Sec. 1.3 for JP1 setting.

1.8 Calibration

Zero/Span Table for 7017 Calibration.

Input Range Code	Input Range	Zero Voltage	Span Voltage
08	±10V	0V	10V
09	±5V	0V	5V
0A	±1V	0V	1V
0B	±500mV	0V	500mV
0C	±150mV	0V	150mV
0D	±20mA	0V or 0mA with 125Ω 0.1%	2.5V or 20mA with 125Ω 0.1%

Zero/Span Table for 7018 Calibration.

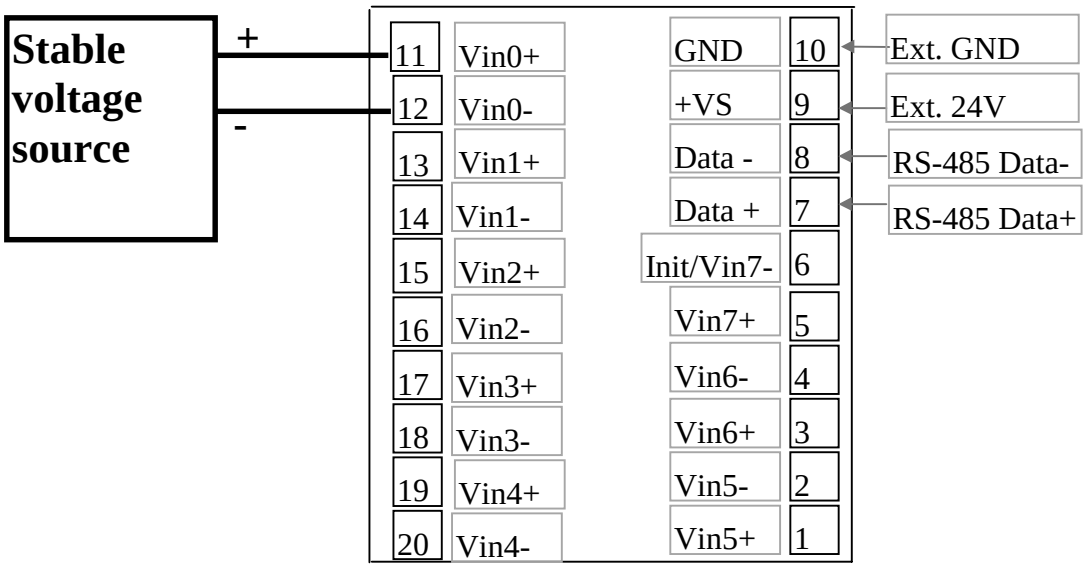
Input Range Code	Input Range	Zero Voltage	Span Voltage
00	±15mV	0V	15mV
01	±50mV	0V	50mV
02	±100mV	0V	100mV
03	±500mV	0V	500mV
04	±1V	0V	1V
05	±2.5V	0V	2.5V
06	±20mA	0V or 0mA with 125Ω 0.1%	2.5V or 20mA with 125Ω 0.1%
0E	J-type	0mV	42.922mV
0F	K-type	0mV	54.875mV
10	T-type	0mV	20.9mV
11	E-type	0mV	76.358mV
12	R-type	0mV	21.108mV
13	S-type	0mV	18.698mV
14	B-type	0mV	13.814mV
15	N-type	0mV	47.502mV
16	C-type	0mV	37.107mV

Zero/Span Table for 7013, 7013D Calibration.

Input Range Code	Input Range	Zero Resistor	Span Registor
All		**55.00 ohm, 0.01%**	**375.00 ohm, 0.01%**

NOTE: One type calibrating is enough.

1.8.1 7017 Calibration



Step 1: Wire connection, install a stable voltage source to channel_0.
Step 2: Power-on, warm-up about 30 minutes
Step 3: Perform type-08 calibration
Step 4: Perform type-09 calibration
..
Step 8: Perform type-0D calibration

The calibration steps of type-08 are given as following:

1. Run TEST.EXE
2. press **2**
3. press **%0101080600[Enter]→** Receive=!01
4. Apply ZERO-Voltage to Channel_0
5. press **2**
6. press **$011[Enter]** → Receive=!01
7. Apply SPAN-Voltage to Channel_0
8. press **2**
9. press **$010[Enter]** → Receive=!01
10. repeat step-4 to step-9 three times.

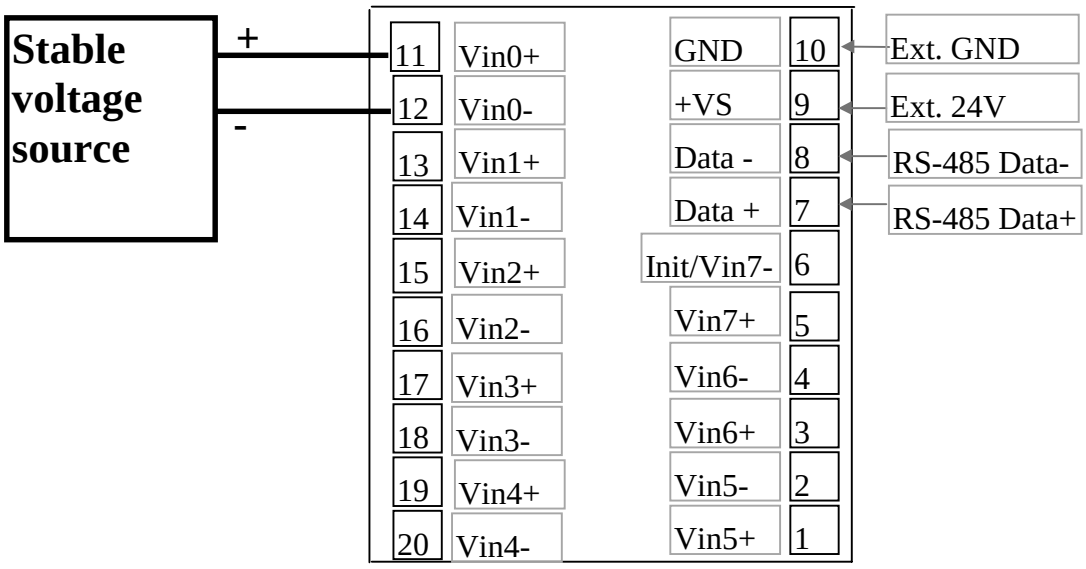
4→ 5→ 6
Zero Calibration

7→ 8→ 9
Span Calibration

NOTE: calibration steps are all the same for type-08 to type-

0D. Only the SPAN-Voltage is different.

1.8.2 7018 Calibration



Step 1: Wire connection, install a stable voltage source to channel_0.
Step 2: Power-on, warm-up about 30 minutes
Step 3: Perform type-00 calibration
Step 4: Perform type-01 calibration
..
Step 18: Perform type-16 calibration

The calibration steps of type-00 are given as following:

1. Run TEST.EXE
2. press **2**
3. press **%0101000600[Enter]→** Receive=!01
4. Apply ZERO-Voltage to Channel_0
5. press **2**
6. press **$011[Enter]** → Receive=!01
7. Apply SPAN-Voltage to Channel_0
8. press **2**
9. press **$010[Enter]** → Receive=!01
10. repeat step-4 to step-9 three times.

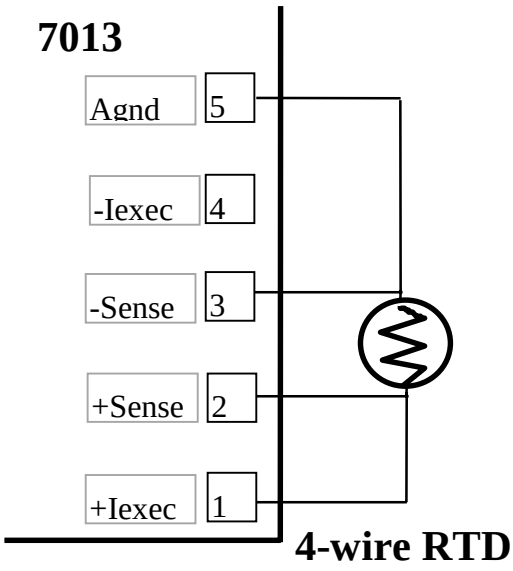
4→ 5→ 6
Zero Calibration

7→ 8→ 9
Span Calibration

NOTE: calibration steps are all the same for type-00 to type-

16. Only the SPAN-Voltage is different.

1.8.3 7013 Calibration



Step 1: Wire connection, install a stable voltage source to channel_0.
Step 2: Power-on, warm-up about 30 minutes
Step 3: Perform type-20 calibration

The calibration steps of type-20 are given as following:

1. Run TEST.EXE
2. press **2**
3. press **%0101200600[Enter]**→ Receive=!01
4. Install ZERO-Registor
5. press **2**
6. press **$011[Enter]** → Receive=!01
7. Install SPAN-Registor
8. press **2**
9. press **$010[Enter]** → Receive=!01
10. repeat step-4 to step-9 three times.

4→ 5→ 6
Zero Calibration

7→ 8→ 9
Span Calibration

NOTE: Only type-20 calibration is enough.

1.9 Tables

Configuration Code Table : CC (for 7017, 7018, 7013)

CC	Baud Rate
03	1200 BPS
04	2400 BPS
05	4800 BPS
06	9600 BPS
07	19200 BPS
08	38400 BPS
09	57600 BPS
0A	115200 BPS

Configuration Code : FF, 2-char (for 7017, 7018, 7013)

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	checksum 0=disable 1=enable	0				00: engineering unit 01: % of FSR 10: 2.s complement of hexadecimal 11: Ohms (for I-7013)	

Configuration Code Table : TT (for 7017)

TT	Input Range
08	+/- 10V
09	+/- 5V
0A	+/- 1V
0B	+/- 500mV
0C	+/- 150mV
0D	+/- 20mA

Configuration Code Table : TT (for 7018)

TT	Input Range
00	±15mV
01	±50mV
02	±100mV
03	±500mV
04	±1V
05	±2.5V
06	±20mA
0E	J-type(0°C ~ 760°C)
0F	K-type(0°C ~ 1000°C)
10	T-type(-100°C ~ 400°C)
11	E-type(0°C ~ 1000°C)
12	R-type(500°C ~ 1750°C)
13	S-type(500°C ~ 1750°C)
14	B-type(500°C ~ 1800°C)
15	N-type(-270°C ~ 1300°C)
16	C-type(0°C ~ 2320°C)

Configuration Code Table : TT (for 7013)

TT	Input Range
20	P.,±100°C,α =.00385
21	P.,0-100°C, α =.00385
22	P.,0-200°C, α =.00385
23	P.,0-600°C, α =.00385
24	P.,±100°C, α =.003916
25	P.,0-100°C, α =.003916
26	P.,0-200°C, α =.003916
27	P.,0-600°C, α =.003916
28	N.,-80°C to 100°C
29	N.,0°C to 100°C

Data Format Table (data): (for 7017)

TT	Range	Format	+FSR	Zero	-FSR
08	±10V	Engineering Unit	+10.000	±00.000	-10.000
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000
09	±5V	Engineering Unit	+5.0000	±0.0000	-5.0000
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000
0A	±1V	Engineering Unit	+1.0000	±0.0000	-1.0000
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000
0B	±500mV	Engineering Unit	+500.00	±000.00	-500.00
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000
0C	±150mV	Engineering Unit	+150.00	±000.00	-150.00
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000
0D	±20mA	Engineering Unit	+20.000	±00.000	-20.000
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000

Data Format Table (data): (for 7018)

TT	Range	Format	+FSR	Zero	-FSR
00	±15mV	Engineering Unit	+15.000	±00.000	-15.000
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000
01	±50mV	Engineering Unit	+50.000	±00.000	-50.000
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000
02	±100mV	Engineering Unit	+100.00	±000.00	-100.00
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000

Data Format Table (data): (for 7018, continue)

03	±500mV	Engineering Unit	+500.00	±000.00	-500.00
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000
04	±1V	Engineering Unit	+1.0000	±0.0000	-1.0000
		% of FSR	7FFF	0000	8000
		2.s complement	+100.00	±000.00	-100.00
05	±2.5V	Engineering Unit	+2.5000	±0.0000	-2.5000
		% of FSR	+100.00	±000.00	-100.00
		2.s complement	7FFF	0000	8000
06	± 20 mA	Engineering Unit	+20.000	±00.000	-20.000
		2.s complement	7FFF	0000	8000
		% of FSR	+100.00	±000.00	-100.00
0E	J-type thermocouple	Engineering Unit	+760.00	+000.00	
		% of FSR	7FFF	0000	
		2.s complement	+100.00	±000.00	
0F	K-type thermocouple	Engineering Unit	+1000.0	+0000.0	
		% of FSR	7FFF	0000	
		2.s complement	+100.00	±000.00	
10	T-type thermocouple	Engineering Unit	+400.00	±000.00	-270.00
		% of FSR	7FFF	0000	8000
		2.s complement	+100.00	±000.00	-100.00
11	E-type thermocouple	Engineering Unit	+1000.0	+0000.0	
		% of FSR	7FFF	0000	
		2.s complement	+100.00	±000.00	
12	R-type thermocouple	Engineering Unit	+1750.0	+0000.0	
		% of FSR	7FFF	0000	
		2.s complement	+100.00	±000.00	
13	S-type thermocouple	Engineering Unit	+1750.0	+0500.0	
		% of FSR	7FFF	2492	
		2.s complement	+100.00	+028.57	
14	B-type thermocouple	Engineering Unit	+1800.0	+0500.0	
		% of FSR	7FFF	2381	
		2.s complement	+100.00	±027.77	

Data Format Table (data): (for 7018, continue)

15	N-type thermocouple	Engineering Unit	+1300.0	+0000.0	-0270.0
		% of FSR	7FFF	0000	8000
		2.s complement	+100.00	±000.00	-100.00
16	C-type thermocouple	Engineering Unit	+2320.0	+0000.0	
		% of FSR	7FFF	0000	
		2.s complement	+100.00	±000.00	

Data Format Table : TT (for 7013)

TT	Input Range	Format	+FSR	-FSR
20	Platinum ±100°C α=.00385	Engineering Unit	+100.00	-100.00
		% of FSR	+100.00	+000.00
		2.s complement	7FFF	8000
		Ohm	+138.50	+060.60
21	Platinum 0-100°C α=.00385	Engineering Unit	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2.s complement	7FFF	0000
		Ohm	+138.50	+100.0
22	Platinum 0-200°C α=.00385	Engineering Unit	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2.s complement	7FFF	0000
		Ohm	+175.84	+100.0
23	Platinum 0-600°C α=.00385	Engineering Unit	+600.00	+000.00
		% of FSR	+100.00	+000.00
		2.s complement	7FFF	0000
		Ohm	+313.59	+100.0
24	Platinum ±100°C α=.00392	Engineering Unit	+100.00	-000.00
		% of FSR	+100.00	-100.00
		2.s complement	7FFF	8000
		Ohm	+139.16	-60.60
25	Platinum 0-100°C α=.00392	Engineering Unit	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2.s complement	7FFF	0000
		Ohm	+139.16	+100.0

Data Format Table : TT (for 7013, continue)

26	Platinum 0-200°C α=.00392	Engineering Unit	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2.s complement	7FFF	0000
		Ohm	+177.13	+100.0
27	Platinum 0-600°C α=.00392	Engineering Unit	+600.00	+000.00
		% of FSR	+100.00	+000.00
		2.s complement	7FFF	0000
		Ohm	+317.28	+100.0
28	120 ohm Nickel -80-100°C	Engineering Unit	+100.00	-080.00
		% of FSR	+100.00	+080.00
		2.s complement	7FFF	999A
		Ohm	+200.64	+066.60
29	120 ohm Nickel 0-100°C	Engineering Unit	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2.s complement	7FFF	0000
		Ohm	+200.64	+120.00

2. Command Set

Command Set Table

Command	Response	Description	Reference
%AANNTTCCFF	!AA	Set module configuration	Sec. 2.1
#**	No Response	Synchronized Sampling	Sec. 2.2
#AA	>(data)	Read analog input	Sec. 2.3
#AAN	>(data)	Read analog input from channel_N	Sec. 2.4
$AA0	!AA	Perform span calibration	Sec. 2.5
$AA1	!AA	Perform zero calibration	Sec. 2.6
$AA2	!AATTCCFF	Read configuration	Sec. 2.7
$AA3	!AA(data)	Read CJC value	Sec. 2.8
$AA4	!AA(data)	Read Synchronized Data	Sec. 2.9
$AA5VV	!AA	Enable/disable channel multiplexing	Sec. 2.10
$AA6	!AAVV	Read channel multiplexing status	Sec. 2.11
$AA8V	!AA	Select Led Configuration	Sec. 2.12
$AA9SCCCC	!AA	Set CJC Offset Value	Sec. 2.13
$AA9S(data)	!AA	Send Led Display	Sec. 2.14
$AAA	**>(data)*8**	**Read all 8 channel data**	Sec. 2.15
$AAF	!AA(data)	Read the firmware version number	Sec. 2.16
$AAM	!AA(data)	Read the module name	Sec. 2.17
~**	No Response	Host OK	Sec. 2.18
~AA0	!AASS	Read Module Status	Sec. 2.19
~AA1	!AA	Reset Module Status	Sec. 2.20
~AA2	!AATT	Read Host Watchdog Timer Value	Sec. 2.21
~AA3ETT	!AA	Enable Host Watchdog Timer	Sec. 2.22
~AAO(name)	!AA	Set module name	Sec. 2.23

7017 Command Set Table

Command	Response	Description	Reference
%AANNTTCCFF	!AA	Set module configuration	Sec. 2.1
#AAN	>(data)	Read analog input from channel_N	Sec. 2.4
$AA0	!AA	Perform span calibration	Sec. 2.5
$AA1	!AA	Perform zero calibration	Sec. 2.6
$AA2	!AATTCCFF	Read configuration	Sec. 2.7
$AA5VV	!AA	Enable/disable channel multiplexing	Sec. 2.10
$AA6	!AAVV	Read channel multiplexing status	Sec. 2.11
$AAA	**>(data)*8**	**Read all 8 channel data**	Sec. 2.15

$AAF	!AA(data)	Read the firmware version number	Sec. 2.16
$AAM	!AA(data)	Read the module name	Sec. 2.17
~**	No Response	Host OK	Sec. 2.18
~AA0	!AASS	Read Module Status	Sec. 2.19
~AA1	!AA	Reset Module Status	Sec. 2.20
~AA2	!AATT	Read Host Watchdog Timer Value	Sec. 2.21
~AA3ETT	!AA	Enable Host Watchdog Timer	Sec. 2.22
~AAO(name)	!AA	Set module name	Sec. 2.23

7018 Command Set Table

Command	Response	Description	Reference
%AANNTTCCFF	!AA	Set module configuration	Sec. 2.1
#AAN	>(data)	Read analog input from channel_N	Sec. 2.4
$AA0	!AA	Perform span calibration	Sec. 2.5
$AA1	!AA	Perform zero calibration	Sec. 2.6
$AA2	!AATTCCFF	Read configuration	Sec. 2.7
$AA3	!AA(data)	Read CJC value	Sec. 2.8
$AA5VV	!AA	Enable/disable channel multiplexing	Sec. 2.10
$AA6	!AAVV	Read channel multiplexing status	Sec. 2.11
$AA9SCCCC	!AA	Set CJC Offset Value	Sec. 2.13

$AAF	!AA(data)	Read the firmware version number	Sec. 2.16
$AAM	!AA(data)	Read the module name	Sec. 2.17
~**	No Response	Host OK	Sec. 2.18
~AA0	!AASS	Read Module Status	Sec. 2.19
~AA1	!AA	Reset Module Status	Sec. 2.20
~AA2	!AATT	Read Host Watchdog Timer Value	Sec. 2.21
~AA3ETT	!AA	Enable Host Watchdog Timer	Sec. 2.22
~AAO(name)	!AA	Set module name	Sec. 2.23

7013 Command Set Table

Command	Response	Description	Reference
%AANNTTCCFF	!AA	Set module configuration	Sec. 2.1
#**	No Response	Synchronized Sampling	Sec. 2.2
#AA	>(data)	Read analog input	Sec. 2.3
$AA0	!AA	Perform span calibration	Sec. 2.5
$AA1	!AA	Perform zero calibration	Sec. 2.6
$AA2	!AATTCCFF	Read configuration	Sec. 2.7
$AA4	!AA(data)	Read Synchronized Data	Sec. 2.9

$AAF	!AA(data)	Read the firmware version number	Sec. 2.16
$AAM	!AA(data)	Read the module name	Sec. 2.17
~**	No Response	Host OK	Sec. 2.18
~AA0	!AASS	Read Module Status	Sec. 2.19
~AA1	!AA	Reset Module Status	Sec. 2.20
~AA2	!AATT	Read Host Watchdog Timer Value	Sec. 2.21
~AA3ETT	!AA	Enable Host Watchdog Timer	Sec. 2.22
~AAO(name)	!AA	Set module name	Sec. 2.23

7013D Command Set Table

Command	Response	Description	Reference
%AANNTTCCFF	!AA	Set module configuration	Sec. 2.1
#**	No Response	Synchronized Sampling	Sec. 2.2
#AA	>(data)	Read analog input	Sec. 2.3
$AA0	!AA	Perform span calibration	Sec. 2.5
$AA1	!AA	Perform zero calibration	Sec. 2.6
$AA2	!AATTCCFF	Read configuration	Sec. 2.7
$AA4	!AA(data)	Read Synchronized Data	Sec. 2.9
$AA8V	!AA	Select Led Configuration	Sec. 2.10
$AA9S(data)	!AA	Send Led Display	Sec. 2.11

$AAF	!AA(data)	Read the firmware version number	Sec. 2.16
$AAM	!AA(data)	Read the module name	Sec. 2.17
~**	No Response	Host OK	Sec. 2.18
~AA0	!AASS	Read Module Status	Sec. 2.19
~AA1	!AA	Reset Module Status	Sec. 2.20
~AA2	!AATT	Read Host Watchdog Timer Value	Sec. 2.21
~AA3ETT	!AA	Enable Host Watchdog Timer	Sec. 2.22
~AAO(name)	!AA	Set module name	Sec. 2.23

2.1 %AANNTTCCFF

For all

- **Description**: Set the configuration of module.

- **Syntax**: %AANNTTCCFF[chk](cr)
  % is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  NN=new AA
  TT=Input range code, refer to Sec. 1.9
  CC=baud rate code, refer to Sec. 1.9
  FF=status code, refer to Sec. 1.9
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:
  command: %0102090600(cr)
  response : !02(cr)

  address 01 is configured to a new address 02, ±5V input

  command: %0202080600(cr)
  response : !02(cr)

  change to ±10V input

Refer to . 7000 Bus Converter User Manual. chapter-5 for the following functions:

- **module status unknown**(Sec. 5.1), **change address**(Sec. 5.2)
- **change baud rate**(Sec. 5.3), **checksum enable/disable**(Sec. 5.4)

2.2 #**

For 7013/7013D

- **Description**: All 7000 input module, including digital and analog, will sample all their input data immediately and store these data in the internal register of module. Later the host computer can read these data one by one by the command **$AA4, read synchronized data.**
- **Syntax**: #**[chk](cr)
  # is a delimiter character
  * is a command character
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D
- **Response**: no response
- **Example**:

command: #**(cr)
response: no response
command: $014(cr)
response: !1©©©©©©©©(cr)
command: $024(cr)
response: !1©©©©©©©©(cr)
command: $034(cr)
response: !1©©©©©©©©(cr)

Order all modules perform synchronized sampling

Read these synchronized sampling data one-by one. In this example, read module-01, 02, 03. © is a character depended on module.s wiring and command

NOTE : What.s . synchronize sampling. ?

The host computer can send only one command string once a time. If there are two modules, the host computer must send and receive the module-1 command then the module-2 command. **So there is a time delay between these two commands.** The . synchronize sampling. command is designed for all input modules. When receiving **#**[0x0D]**, synchronized sampling command, **all the input modules in the RS-485 network will perform the input function at the same time and store these values into the module.s memory.** Then the host computer can send out the . $AA4, read synchronize data. command to read these data separately.

2.3 #AA

For-7013/7013D

- **Description**: Read the analog input value.

- **Syntax** : #AA[chk](cr)
  # is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response** : valid command → >(data)[chk](cr)
  invalid command → No Response
  no response → syntax error or communication error or address error
  > is a delimiter character indicating a valid command
  (data) = refer to Sec. 1.9
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: #01(cr)
  response : >+100.00(cr)

  Temperature=100.0°C

  command: #02(cr)
  response : >-100.00(cr)

  Temperature=-100.0°C

2.4 #AAN

For 7017/7108

- **Description**: Read the analog value from channel N.

- **Syntax**: #AAN[chk](cr)
  # is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  N=channel number, from 0 to 7
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → >(data)[chk](cr)
  invalid command → No Response
  no response → syntax error or communication error or address error
  > is a delimiter character indicating a valid command
  (data) = refer to Sec. 1.9
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: #010(cr)
  response : >+1.2345(cr)

  channel_0=1.2345V.

  command: #012(cr)
  response : >+444.44(cr)

  channel_2=444.44mV

2.5 $AA0

For all

- **Description**: Perform the SPAN calibration. Refer to Sec. 1.8.1 & Sec. 1.8.2 & Sec. 1.8.3 for more information.

- **Syntax**: $AA0[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example:**
  command: $010(cr)
  response : !01(cr)

  address 01 perform SPAN calibration

  command: $020(cr)
  response : !02(cr)

  address 02 perform SPAN calibration

2.6 $AA1

For all

- **Description**: Perform the ZERO calibration. Refer to Sec. 1.8.1 & Sec. 1.8.2 & Sec. 1.8.3 for more information.
-

- **Syntax**: $AA1[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $011(cr)
  response : !01(cr)

  address 01 perform ZERO calibration

  command: $021(cr)
  response : !02(cr)

  address 02 perform ZERO calibration

2.7 $AA2

For all

- **Description**: Read the configuration of module.
- **Syntax**: $AA2[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D
- **Response**: valid command → !AATTCCFF[chk](cr),
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  TT, CC, FF: refer to Sec. 1.9
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D
- **Example**:
  command: $012(cr)
  response : !01080600(cr)

  address 01, ±10V, 9600 BPS, checksum disable, engineering unit

  command: $022(cr)
  response : !02050700(cr)

  address 02, ±2.5V, 19200 BPS, checksum disable, engineering unit

NOTE: If the user use %AANNTTCCFF command to change module configuration, the new configuration code will be stored into EEPROM immediately. The configuration code includes module address, module type, baud rate code, checksum enable/disable code, calibration code, power-on value and safe value. **The EEPROM data of 7000 can be read infinite times and can be written about 100,000 times max.** Therefore the user should not change configuration code often for testing.
The $AA2 command is used to read EEPROM data only, therefore the user can send this command to 7000 module infinitely.

2.8 $AA3

For 7018

- **Description**: Read current CJC value. Refer to Sec. 3.5 for more information.

- **Syntax**: $AA3[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !S(data)[chk](cr),
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  S=+ or -
  (data)=CJC value
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $013(cr)
  response : !+0030.0(cr)

  CJC=30°C

  command: $023(cr)
  response : !+0032.1(cr)

  CJC=32.1°C

2.9 $AA4

For 7013/7013D

- **Description**: Read the synchronized data.
- **Syntax**: $AA4[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D
- **Response**: valid command → !S(data)[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  S=1=first reading, S=0=not first reading
  (data) = refer to Sec. 1.9
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D
- **Example** :

command: $01M(cr)
response : !017013(cr)
command: $02M(cr)
response : !027013D(cr)
command: #**
response : No Responde
command: $014(cr)
response : !1+123.45(cr)
command: $014(cr)
response : !00+123.45(cr)
command: $024(cr)
response : !1-123.45(cr)
command: $024(cr)
response : !0-123.45(cr)

(1) address-01 is 7013.
(2) address-02 is 7013D
(3) perform synchronized sampling

(5) synchronized data = +123.45, first time
(6) synchronized data = +123.45, not first time

(8) synchronized data = -123.45, first time
(9) synchronized data = -123.45, not first time

2.10 $AA5VV

For 7017/7018

- **Description**: Enable or disable channel multiplexing.

- **Syntax**: $AA5VV[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  VV=2-character HEX value, from 00 to FF, 8 bits refer to 8 channels, 1=enable, 0=disable
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $015F0(cr)
  response : !01(cr)

  enable channel_7 to channel_4
  disable channel_0 to channel_3

  command: $025AA(cr)
  response : !02(cr)

  enable channel_7/5/3/1
  disable channel_6/4/2/0

2.11 $AA6

For 7017/7018

- **Description**: Read channel multiplexing status.

- **Syntax**: $AA6[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AAVV[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  VV=2-character HEX value, from 00 to FF, 8 bits refer to 8 channels, 1=enable, 0=disable
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $016(cr)
  response : !01F0(cr)

  channel_7 to channel_4 are enable
  channel_0 to channel_3 are disable

  command: $026(cr)
  response : !02AA(cr)

  channel_7/5/3/1 are enable
  channel_6/4/2/0 are disable

2.12 $AA8V

For 7013D

- **Description**: Select LED Configuration.

- **Syntax**: $AA8V[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  V=1 → module control LED, V=2 → host control LED
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or
  communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $0181(cr)
  response : !01(cr)

  7013D will control LED

  command: $0282(cr)
  response : !02(cr)

  Host will control LED

2.13 $AA9SCCCC

For 7018

- **Description**: Set CJC offset value. Refer to Sec. 3.5 for more information.

- **Syntax**: $AA9SCCCC[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  S= + or -
  CCCC= 4-char HEX value, 1 count=0.01°C
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $019+000A(cr)
  response : !01(cr)

  CJC offset=10*0.01=0.1°C

  command: $029-0014(cr)
  response : !02(cr)

  CJC offset=-20*0.01=-0.2°C

2.14 $AA9S(data)

For 7013D

- **Description**: Send LED display.

- **Syntax**: $AA9S(data)[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  S=+ or -
  (data)=5 decimal digit + 1 decimal point, max=19999.
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or
  communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $019+19999.(cr)
  response : !01(cr)

  Show max = +19999.

  command: $029-19999.(cr)
  response : !02(cr)

  Show min = -19999.

  command: $039+12.345(cr)
  response : !03(cr)

  Show display = +12.345

2.15 $AAA

For 7017

- **Description**: Read all 8 channel data. Refer to Sec. 3.8 for more information.
- **Syntax**: $AAA[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D
- **Response**: valid command → !(data)*8[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  data=4-character HEX value, from 0000 to FFFF,
  2.s complement data format
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D
- **Example**:
  command: $01A(cr)
  response : !0000111122223333444455556666777(cr)

channel_0=0000
channel_1=1111
channel_2=2222
channel_3=3333
channel_4=4444
channel_5=5555
channel_6=6666
channel_7=7777

- All data are in HEX format
- 8000 → min
- 7FFF → max
- 0000 → 0
- assume type=08,

1. 8000 → -10V
2. 7FFF → +10V
3. 0000 → 0V
4. 1000 → +1.25V
5. F000 → -1.25V

2.16 $AAF

For all

- **Description**: Read the version number of firmware.

- **Syntax**: $AAF[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA(data)[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or
  communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  data=5-character for version number
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $01F(cr)
  response : !01A2.0(cr)

  module 01 version A2.0

  command: $02F(cr)
  response : !02A3.0(cr)

  module 02 version A3.0

2.17 $AAM

For all

- **Description**: Read the module name.

- **Syntax**: $AAM[chk](cr)
  $ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA(data)[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or
  communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  data=4-character for module name
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $01M(cr)
  response : !017017(cr)

  Module name of 01 is 7017

  command: $02M(cr)
  response : !027018(cr)

  Module name of 02 is 7018

  command: $03M(cr)
  response : !037013(cr)

  Module name of 03 is 7013

  command: $04M(cr)
  response : !047013D(cr)

  Module name of 04 is 7013D

2.18 ~**

For all

- **Description**: Host send this command to tell all modules . Host is OK. . Refer to Sec. 3.5 for more information.

- **Syntax**: ~**[chk](cr)
  ~ is a delimiter character
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: no response

- **Example**:
  command: ~**(cr)
  response : No Response

2.19 ~AA0

For all

- **Description**: Read the module status. The module status will be latch until ~AA1 command is sent**. If the module status=0x04, all output command will be ignored.**

- **Syntax**: ~AA0[chk](cr)
  ~ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AASS[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  SS=2-character HEX status value as following:
  Bit_0, Bit_1 = reserved
  Bit_2 = 0 → OK,
  1 → host watchdog failure
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

command: ~010(cr)
response : !0100(cr)

Status of module 01 is OK

command: ~020(cr)
response : !0204(cr)

Status of module 02 is . host watchdog failure. → HOST is down now

2.20 ~AA1

For all

- **Description**: Reset module status. The module status will be latched until ~AA1 command is sent. **If the module status=0x04, all output command will be ignored.** Therefore the user should read the module status first to make sure that the module status is 0. If the module status is not 0, only ~AA1 command can clear the module status.

- **Syntax**: ~AA1[chk](cr)
  ~ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D
- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

command: ~010(cr)	
response : !0104(cr)	module status=0x04 → host is down
command: #0105.000(cr)	
response : !(cr)	Output command is ignored
command: ~011(cr)	
response : !01(cr)	clear module status
command: ~010(cr)	
response : !0100(cr)	module status=0x00
command: #0105.000(cr)	
response : >(cr)	Output command is OK

2.21 ~AA2

For all

- **Description**: Read the status and timer value of host watchdog. The host watchdog timer is designed for host watchdog. When the host watchdog is enable, the host must send ~** command to all modules before the timer is up. When the ~** command is received, the host watchdog timer is reset and restart. Use ~AA3ETT to enable/disable/setting the host watchdog timer.

- **Syntax**: ~AA2[chk](cr)
  ~ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D
- **Response**: valid command → !AASTT[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  S=0: host watchdog is disable
  S=1: host watchdog is enable
  TT=2-character HEX value, from 00 to FF, unit=0.1 second
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: ~012(cr)
  response : !01000(cr)

  Host watchdog timer of module 01 is disable

  command: ~022(cr)
  response : !0210A(cr)

  host watchdog timer of module 02 is enable and =0.1*10 =1 second.

2.22 ~AA3ETT

For all

- **Description**: Enable/disable the timer value of host watchdog. The host watchdog timer is designed for software host watchdog. When the software host watchdog is enable, the host must send ~** command to all modules before the timer is up. When the ~** command is received, the host watchdog timer is reset and restart. Use ~AA2 to read the host watchdog status & value.

- **Syntax**: ~AA3ETT[chk](cr)
  ~ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  E=0 is disable and 1 is enable
  TT=2-character HEX value, from 00 to FF, unit=0.1 second
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:
  command: ~013000(cr)
  response : !01(cr)

  disable host watchdog timer of module 01

  command: ~02310A(cr)
  response : !02(cr)

  host watchdog timer of module 02 is enable and equal to 0.1*10 =1 second.

2.23 ~AAO(name)

For all

- **Description**: Set module name.

- **Syntax**: ~AAO(name)[chk](cr)
  ~ is a delimiter character
  AA=2-character HEX module address, from 00 to FF
  (name)=4-character/5-character module name
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Response**: valid command → !AA[chk](cr)
  invalid command → ?AA[chk](cr)
  no response → syntax error or communication error or address error
  ! is a delimiter character indicating a valid command
  ? is a delimiter character indicating a invalid command
  AA=2-character HEX module address
  [chk]=2-character checksum, if checksum disable → no [chk]
  (cr)=0x0D

- **Example**:

  command: $01M(cr)
  response : !017050(cr)
  command: ~01O8050(cr)
  response : !01(cr)

  Change module name from 7050 to 8050

  command: $01M(cr)
  response : !017080D(cr)
  command: ~01O8080D(cr)
  response : !01(cr)

  Change module name from 7080D to 8080D

Note: This command is designed for OEM/ODM user. The user can use it to change the module name for other purpose.

3. Operations Principle & Application Notes

3.1 INIT*_pin Operation Principle

All 7000 modules contain an EEPROM to store configuration information. Therefore the user is difficult to find out the status of the 7000 modules. The user can connect the INIT*_pin to GND_pin and power on the module. The 7000 modules will **go to the factory default setting without change the EEPROM data.** The factory default setting is given as following:

address = 00
baud rate = 9600
checksum = DISABLE
data format = 1 start + 8 data bits + 1 stop bit

If the user disconnect the INIT*_pin and GND_pin, the 7000 module will be auto configured according to the EEPROM data. The user is easy to find the EEPROM configuration data in the default setting. The steps are shown as following:

Step 1 : power off and connect INIT*_pin to GND_pin
Step 2 : power on
Step 3 : send command string **$002[0x0D]**
Step 4 : record the status of this 7000 module
Step 5 : power off and disconnect INIT*_pin and GND_pin
Step 6 : power on

Refer to .7000 Bus Converter User Manual. Sec. 5.1 for more information.

3.2 Dual WatchDog Operation Principle

Dual watchdog = host watchdog + module watchdog
The host watchdog is a software watchdog.
The module watchdog is a hardware watchdog.

The 7000 series are designed for industry applications, therefore they can work in the harsh envioronment. There are many couple noise or energy transient in such environment. The modules may be down if these noise is really too large. The module watchdog can reset the module if it is down for too large signal. Sometimes even the host-PC may be down for hardware or software reasons. The host watchdog can monitor the status of host-PC. If the host-PC is down, all the output of 7000 modules will go to their predefined safe states for safety protection.

If the RS-485 network is open**,** all the host command can not send to remote modules. This is very dangerous in real world application. The 7000 output module will force their output going to their predefined safe state for safety consideration if the host watchdog is enable. This dual watchdog feature will increase the system reliability very much.

The 7017, 7018, 7013 and 7013D are input modules. They will not take any action if the module watchdog or host watchdog is failure. The application program does not have to detect the dual watchdog status.

3.3 Analog Data Format

The 7000 can be configured to one of the following data formats:

- Engineering unit
- Percent of FSR
- Two.s complement hexadecimal
- Ohms

Assume the range is ±5V, the data format are given as following:

Engineering Unit	Percent of FSR	Two.s complement
-5V	-100.00	8000
0V	+000.00	0000
+5V	+100.00	7FFF

The above table is validate for 7017, 7013 & 7013D. If the 7018 is configured to type-00, 01,02, 03, 04,05, 06 and 07. The data format is the same as mention above. The data format of thermocouple is given as following:

Volt	Engineering Unit	Percent of FSR	Two.s complement
-max	Table(-max)+CJC	-100.00	8000
0V	Table(0)+CJC	+000.00	0000
+max	Table(+max)+CJC	+100.00	7FFF

It is recommended to select . engineering unit. if the 7018 is configured as thermocouple type.

3.4 Temperature Measurement

The 7018 can be configured as thermocouple input. The steps to measure thermocouple is given as following:

1. A/D conversion → measure thermocouple voltage
2. Table lookup → T1=Table(thermocouple voltage)
3. Get Temperature → Temperature=T1+CJC-value

The steps to measure CJC is given as following:

1. A/D conversion → measure CJC voltage
2. Table lookup → T1=Table(CJC voltage)
3. CJC compensation → CJC-value=T1+CJC-offset

- The CJC-offset is defined by $AA9 command.
- The CJC-value can be read back by $AA3 command.

Therefore the temperature error is composed of four errors as following:

1. Thermocouple error → small
2. A/D converter error → small
3. Table lookup error → small
4. CJC-error → **may be big**
5. Temperature error = (1)+(2)+(3)+(4)

Refer to Sec. 3.5 for CJC offset calibration if the CJC value is not proper.

3.5 CJC Offset Calibration

The operation steps of CJC offset calibration are given as following:

1. Place a silver temperature sensor just beside the 7018 CJC sensor. Power on and warm-up about 30 minutes in real place. This step is used to find the circumstance temperature. The silver sensor is used to calibrate the CJC sensor.
2. Use $AA9+0000 to set CJC offset=0
3. Use $AA3 to read out CJC value, T1
4. Read out silver sensor temperature, T2
5. CJC offset=T2-T1
6. Use $AA9+???? to set CJC offset
7. Use $AA3 to read out CJC value, T1
8. Repeat step 2 to step 7 until T1=T2

3.6 Command Response Time

The command response time of 7000 D/I/O, D/A and Timer/Counter modules is given as following:

1. Host send command
2. 7000 module receive this command nearly at the same time
3. **7000 wait a character time**
4. 7000 get the current value and send this value back to host.
5. Assume the communication condition is given as following:

- baud rate=115.2K
- command= **#01(cr)** → 4 character
- wait 1 character
- response= >**HHHH(cr)** → 6 character
- total character= 4+1+6=11 character
- 1 character= 10 bits → 115.2K/10=11.52K
- 11 character → 11.52K/11=1.0K max.
  → 1000 command/response per second max.

The length of the command/response is different for different command. There will be some extra computation & control time in real world application. Therefore the user can not reach this max. peformance. **There are two demo program given in NAP7000S, they can be used to test the real performance.** The real performance under 7053 & Pentium-120 is about 820 command/response per second. The user can install 256 modules max. in one RS-485 network. The time to send out 256 command/response is about 256/820=0.3 sec. The 7053 equips 16 channels of D/I, therefore the performance is about 256*16 = **4096 channels per 0.3 second.**

The 7000 A/D modulues have very heavy computation load. Therefore they can not reach this max. performance. **There is one demo program given in NAP7000S, it can be used to test the real performance.** The real performance under 7017 & Pentium-120 is about 63 command/response per second. The 7017 equips 8 channels of A/I, therefore the performance is about 63*8=**500 channels/per second.**

3.7 Response Time of A/D

The sampling rate of 7017, 7018, 7013 and 7013D is **10 samples per second**. The 7017 & 7018 have 8 channels of A/I. Therefore the sampling rate is 10/8=1.25 samples per seconds for every channel.

The max. response time is about 63 command/response per second for 7017(refer to Sec. 3.6). Therefore the user can read 63 data if only one 7017 installed in the network. But the data is updated 10 times per second. So many of these data are repeated. If the user install 63 modules, the performance is 500 channels per second(Sec. 3.6)

Module	One Network	Four Network
7012 **1 channels of A/I**	**256 samples per 0.3 second**	**1000 samples per second**
7017 **8 channels of A/I**	**1900 samples per sec**	**7000 samples per second**
7053 **16 channels of D/I**	**4096 samples per 0.3 sec**	**16000 samples per second**
7043 **16 channels of D/O**	**4096 samples per 0.3 sec**	**16000 samples per second**